Towards Multi-Scale Signal Processing on Graphs

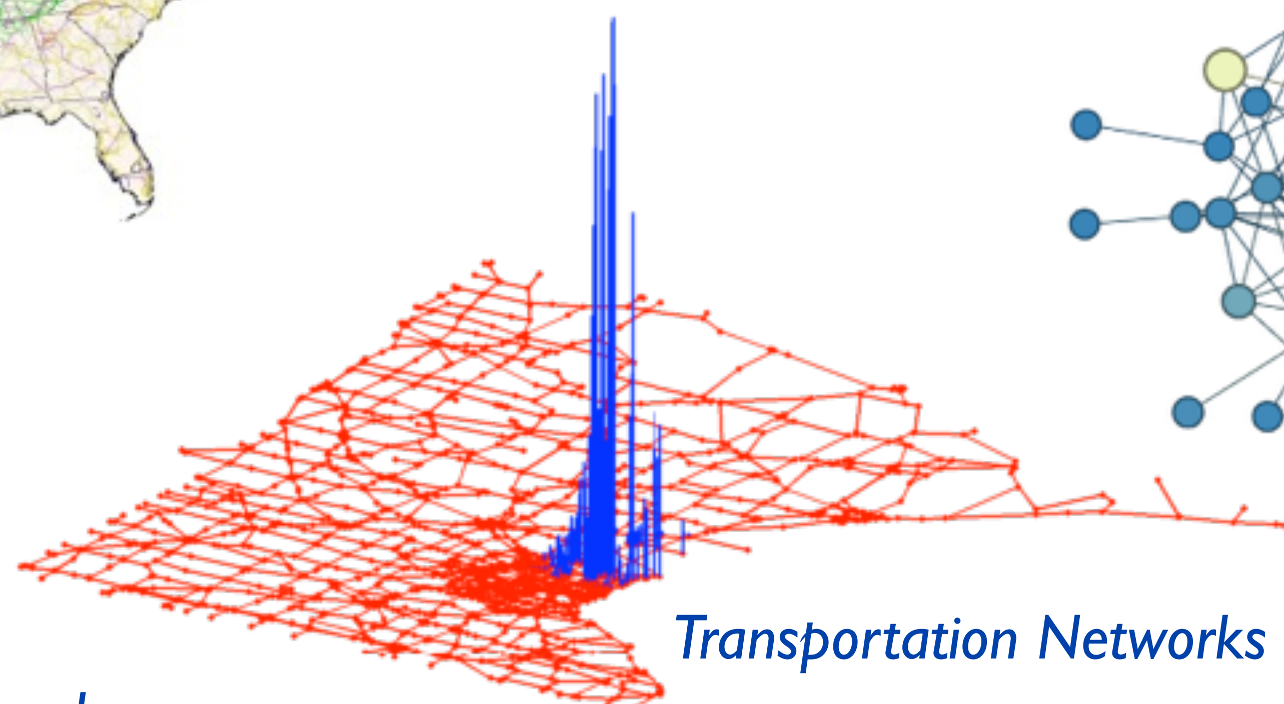
Pierre Vandergheynst
Signal Processing Lab (IEL-LTS2)

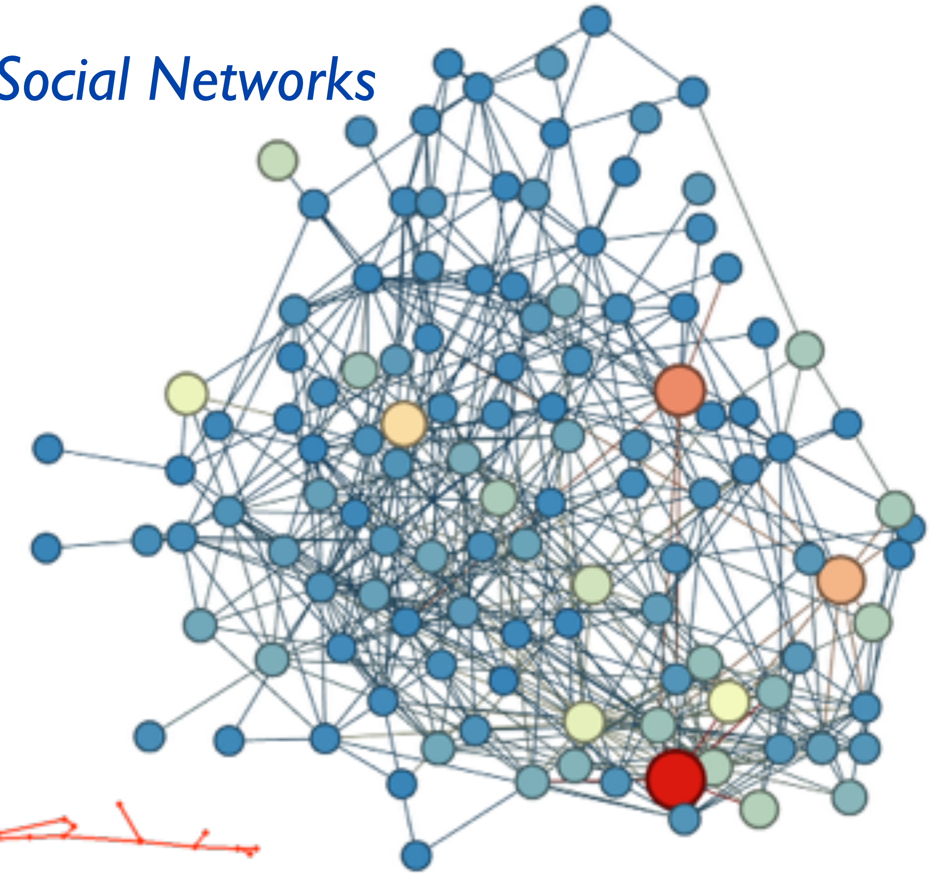
Royal Society Science on the Sphere Meeting
July 2014

Signal Processing on Graphs

Energy Networks

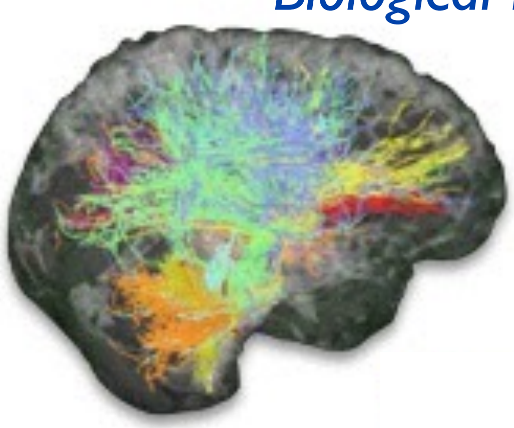
Social Networks

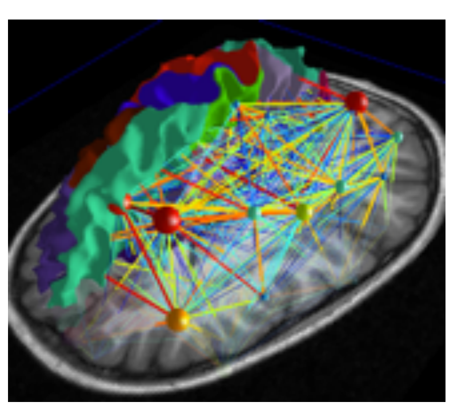


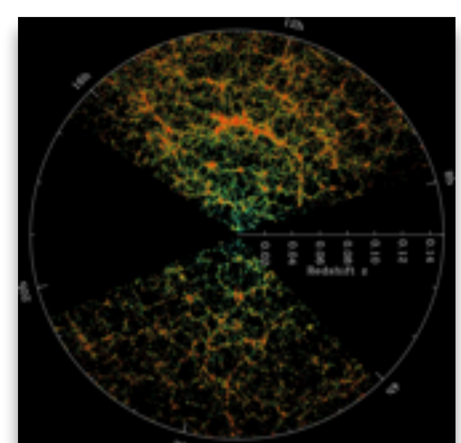


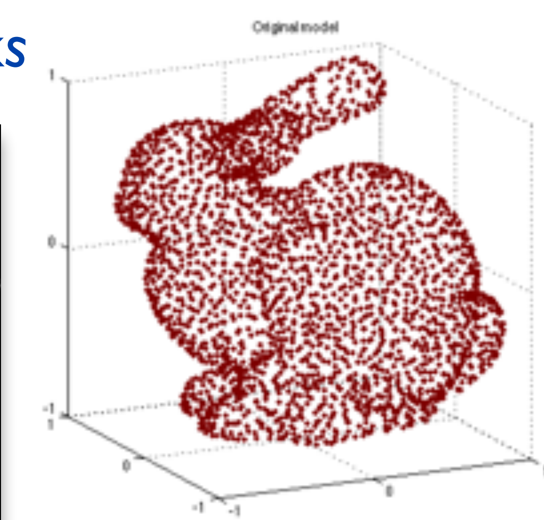
Transportation Networks

Biological Networks









Irregular Data Domains

Fundamentals of DSP

It seems hard to formulate a linear shift-invariant systems theory (LTI) for graphs. But we can try to get close.

The (combinatorial) Laplacian will be our main building block

$$\mathcal{L} = \mathbf{D} - \mathbf{W} \qquad \{(\lambda_\ell, \mathbf{u}_\ell)\}_{\ell=0,1,\ldots,N-1}$$

That particular ortho basis will play the role of the Fourier basis

$$\hat{f}(\lambda_\ell) := \langle \mathbf{f}, \mathbf{u}_\ell \rangle = \sum_{i=1}^{N} f(i) u_\ell^*(i)$$

$$\mu := \max_{\ell,i} |\langle \mathbf{u}_\ell, \delta_i \rangle| \in \left[\frac{1}{\sqrt{N}}, 1\right[$$

Graph Coherence

Simple Motivating Example

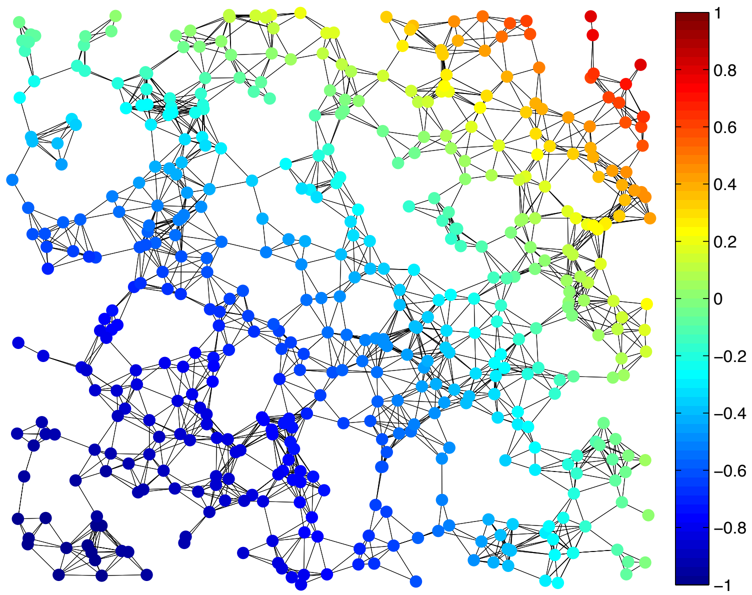
$$\operatorname{argmin}_f \left\{ ||f - y||_2^2 + \gamma f^T \mathcal{L} f \right\}$$

Simple Motivating Example

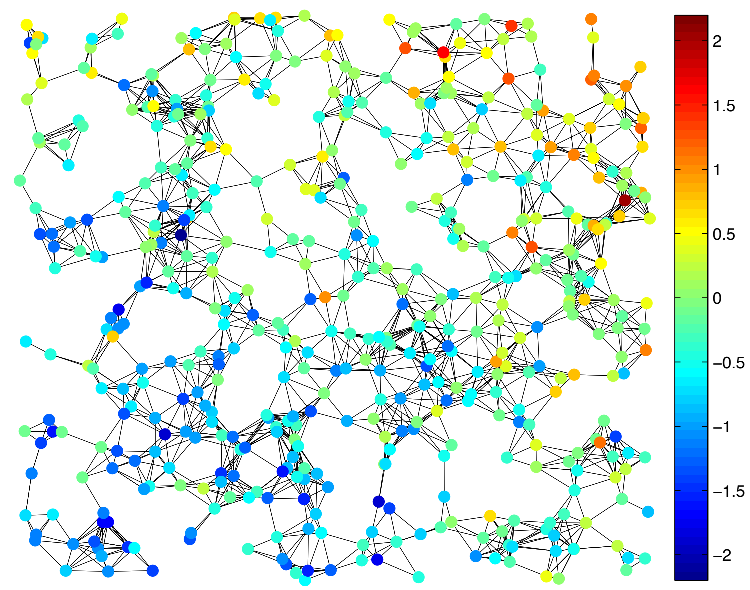
- Tikhonov regularization for denoising: $\operatorname{argmin}_f \left\{ ||f - y||_2^2 + \gamma f^T \mathcal{L} f \right\}$

Simple Motivating Example

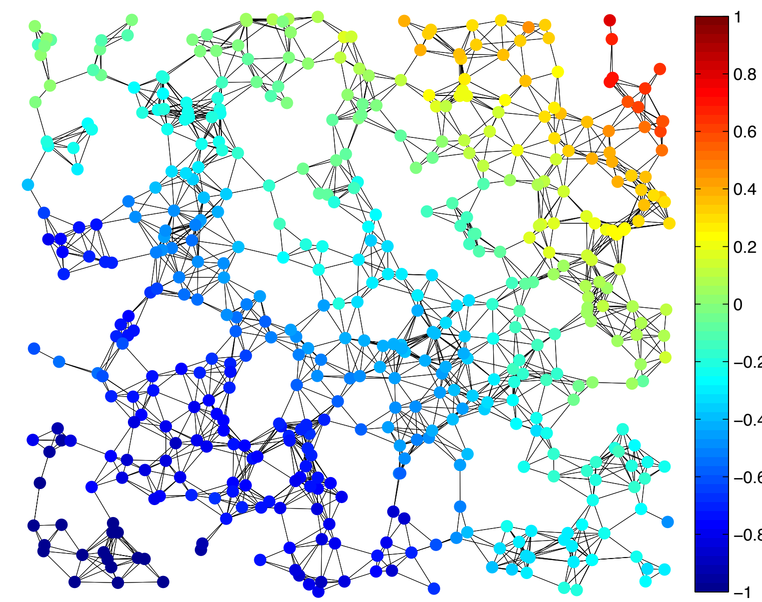
- Tikhonov regularization for denoising: $\mathrm{argmin}_f \left\{ ||f - y||_2^2 + \gamma f^T \mathcal{L} f \right\}$



Original



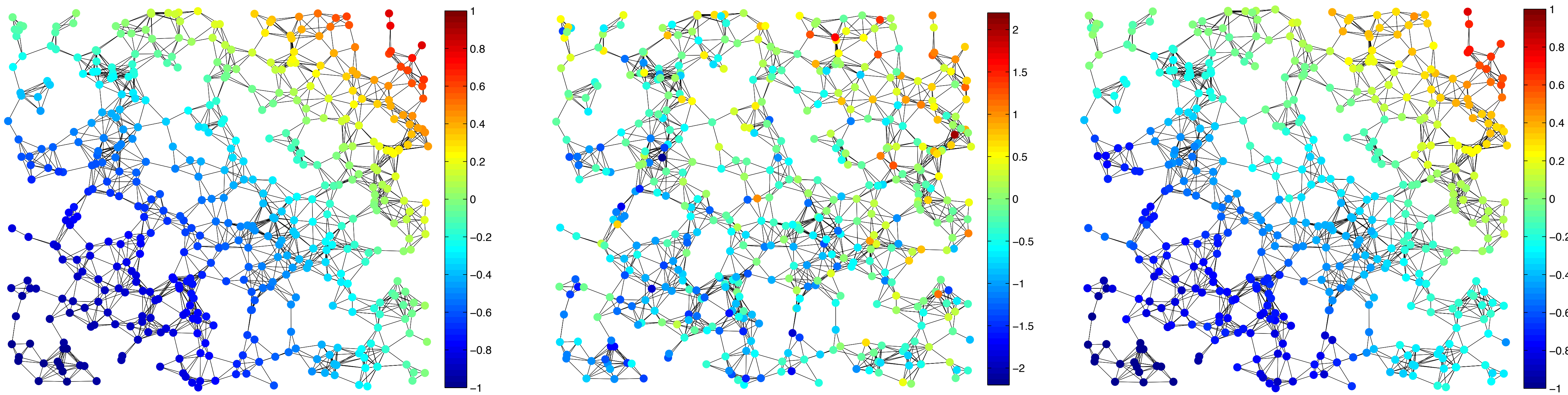
Noisy



Denoised

Simple Motivating Example

- Tikhonov regularization for denoising: $\text{argmin}_f \{ ||f - y||_2^2 + \gamma f^T \mathcal{L} f \}$



Original Noisy Denoised

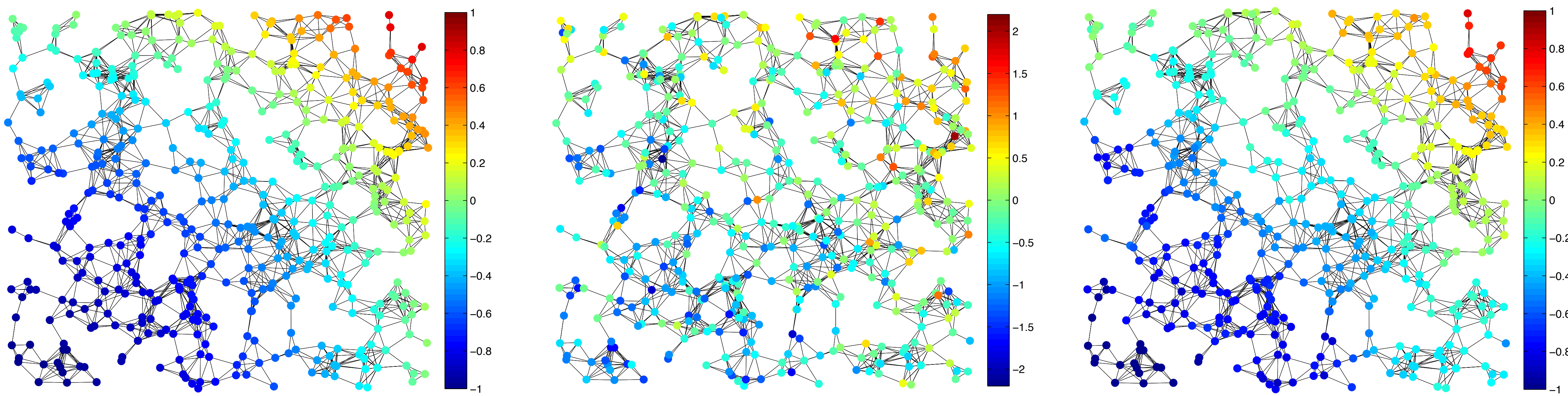
$$\underset{f}{\text{argmin}} \frac{\tau}{2} \|f - y\|_2^2 + f^{\mathrm{T}} \mathcal{L}^r f \quad \Rightarrow \quad \mathcal{L}^r f_* + \frac{\tau}{2}(f_* - y) = 0$$

$$\Rightarrow \quad \widehat{f_*}(\ell) = \frac{\tau}{\tau + 2\lambda_\ell^r} \hat{y}(\ell)$$

"Low pass" filtering !

Simple Motivating Example

- Tikhonov regularization for denoising: $\mathrm{argmin}_f \left\{ ||f-y||_2^2 + \gamma f^T \mathcal{L} f \right\}$



Original   Noisy   Denoised

$$\underset{f}{\mathrm{argmin}} \frac{\tau}{2} \|f-y\|_2^2 + f^{\mathrm{T}} \mathcal{L}^r f \quad \Rightarrow \quad \mathcal{L}^r f_* + \frac{\tau}{2}(f_* - y) = 0$$

$$\Rightarrow \quad \widehat{f_*}(\ell) = \frac{\tau}{\tau + 2\lambda_\ell^r} \hat{y}(\ell)$$ "Low pass" filtering !

Filtering: $\hat{f}_{out}(\lambda_\ell) = \hat{f}_{in}(\lambda_\ell)\hat{h}(\lambda_\ell) \qquad f_{out}(i) = \sum_{\ell=0}^{N-1} \hat{f}_{in}(\lambda_\ell)\hat{h}(\lambda_\ell)u_\ell(i)$

Kernels, Convolutions and Translations

$$(f * g)(n) := \sum_{\ell=0}^{N-1} \hat{f}(\ell)\hat{g}(\ell)u_\ell(n)$$

Inherits a lot of properties of the usual convolution

associativity, distributivity, diagonalized by GFT

$$g_0(n) := \sum_{\ell=0}^{N-1} u_\ell(n) \quad \Longrightarrow \quad f * g_0 = f$$

$$\mathcal{L}(f * g) = (\mathcal{L}f) * g = f * (\mathcal{L}g)$$

Use convolution to induce translations

$$(T_i f)(n) := \sqrt{N}(f * \delta_i)(n) = \sqrt{N} \sum_{\ell=0}^{N-1} \hat{f}(\ell)u_\ell^*(i)u_\ell(n)$$

Polynomial Localization

Given a spectral kernel g, construct the family of features:

$$\phi_n(m) = \big(T_n g\big)(m) \qquad \phi_n(m) = \sqrt{N} \sum_{\ell=0}^{N-1} \hat{g}(\ell) u_\ell^*(m) u_\ell(n)$$

Are these features localized ?

Polynomial Localization

Given a spectral kernel g, construct the family of features:

$$\phi_n(m) = \big(T_n g\big)(m) \qquad \phi_n(m) = \sqrt{N} \sum_{\ell=0}^{N-1} \hat{g}(\ell) u_\ell^*(m) u_\ell(n)$$

Are these features localized ?

Suppose the GFT of the kernel is smooth enough (K+1 different.):

$$B = \sup_x |\hat{g}^{(K+1)}(x)|$$

Polynomial Localization

Given a spectral kernel g, construct the family of features:

$$\phi_n(m) = \big(T_n g\big)(m) \qquad \phi_n(m) = \sqrt{N} \sum_{\ell=0}^{N-1} \hat{g}(\ell) u_\ell^*(m) u_\ell(n)$$

Are these features localized ?

Suppose the GFT of the kernel is smooth enough (K+1 different.):

$$B = \sup_x |\hat{g}^{(K+1)}(x)|$$

Construct an order K polynomial approximation:

$$\sup_\ell |\hat{g}(x) - P_K(x)| \leq \frac{B}{2^K (K+1)!}$$

Polynomial Localization

Given a spectral kernel g, construct the family of features:

$$\phi_n(m) = \big(T_n g\big)(m) \qquad \phi_n(m) = \sqrt{N} \sum_{\ell=0}^{N-1} \hat{g}(\ell) u_\ell^*(m) u_\ell(n)$$

Are these features localized ?

Suppose the GFT of the kernel is smooth enough (K+1 different.):

$$B = \sup_x |\hat{g}^{(K+1)}(x)|$$

Construct an order K polynomial approximation:

$$\sup_\ell |\hat{g}(x) - P_K(x)| \leq \frac{B}{2^K (K+1)!}$$

$$\phi'_n(m) = \langle \delta_m, P_K(\mathcal{L}) \delta_n \rangle$$ Exactly localized in a K-ball around n

Polynomial Localization - Extended

f is $(K{+}1)$-times differentiable:

$$\inf_{q_K}\{\|f - q_K\|_\infty\} \leq \frac{\left[\frac{b-a}{2}\right]^{K+1}}{(K+1)!\; 2^K}\|f^{(K+1)}\|_\infty$$

Let $K_{in} := d(i,n) - 1$

$$|(T_i g)(n)| \leq \sqrt{N} \inf_{\widehat{p_{K_{in}}}} \left\{ \sup_{\lambda\in[0,\lambda_{\max}]} |\hat{g}(\lambda) - \widehat{p_{K_{in}}}(\lambda)| \right\} = \sqrt{N} \inf_{\widehat{p_{K_{in}}}} \{\|\hat{g} - \widehat{p_{K_{in}}}\|_\infty\}$$

Regular Kernels are Localized

If the kernel is $d(i,n)$-times differentiable:

$$|(T_i g)(n)| \leq \left[\frac{2\sqrt{N}}{d_{in}!}\left(\frac{\lambda_{\max}}{4}\right)^{d_{in}} \sup_{\lambda\in[0,\lambda_{\max}]} |\hat{g}^{(d_{in})}(\lambda)|\right]$$

Polynomial Localization - Extended

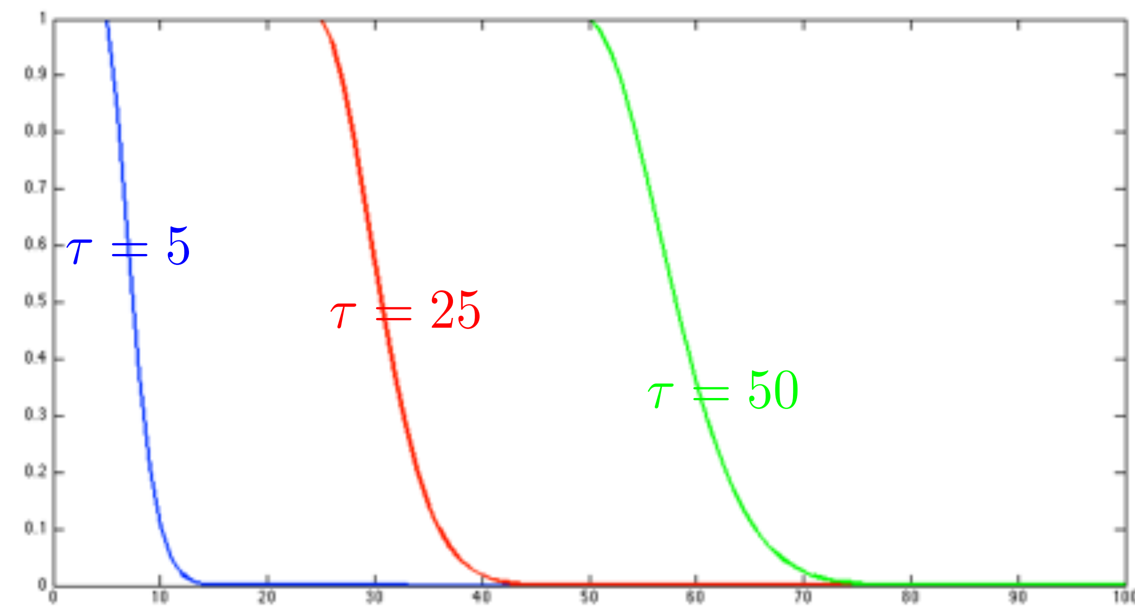
Example: for the heat kernel $\hat{g}(\lambda) = e^{-\tau\lambda}$

$$\frac{|(T_i g)(n)|}{\|T_i g\|_2} \leq \frac{2\sqrt{N}}{d_{in}!}\left(\frac{\tau\lambda_{\max}}{4}\right)^{d_{in}} \leq \sqrt{\frac{2N}{d_{in}\pi}}\, e^{-\frac{1}{12d_{in}+1}}\left(\frac{\tau\lambda_{\max}e}{4d_{in}}\right)^{d_{in}}$$

We can estimate an explicit measure of spread in terms of the degrees:

$$\Delta_i^2(f) = \frac{1}{\|f\|_2^2}\sum_{n=1}^{N} d_{in}^2 [f(n)]^2$$

$$\Delta_i^2(T_i g) \leq \frac{\tau N \lambda_{\max} e D_i}{(2\pi)^{\frac{3}{2}}} e^{\frac{\tau\lambda_{\max}e^2(D_{\max}-1)}{4}}$$



$$\tau \to 0 \Rightarrow T_i g \to \delta_i,\ \Delta_i^2(T_i g) \to 0$$

$$\tau \to +\infty \Rightarrow T_i g \to \frac{1}{\sqrt{N}},\ \Delta_i^2(T_i g) \to \frac{1}{N}\sum_{n=1}^{N} d(i,n)^2$$

Localization in action

Hammond et al., Wavelets on graphs via spectral graph theory, ACHA, 2011

Localization in action

Hammond et al., Wavelets on graphs via spectral graph theory, ACHA, 2011

- Generalized translation

Localization in action

Hammond et al., Wavelets on graphs via spectral graph theory, ACHA, 2011

- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$

Localization in action

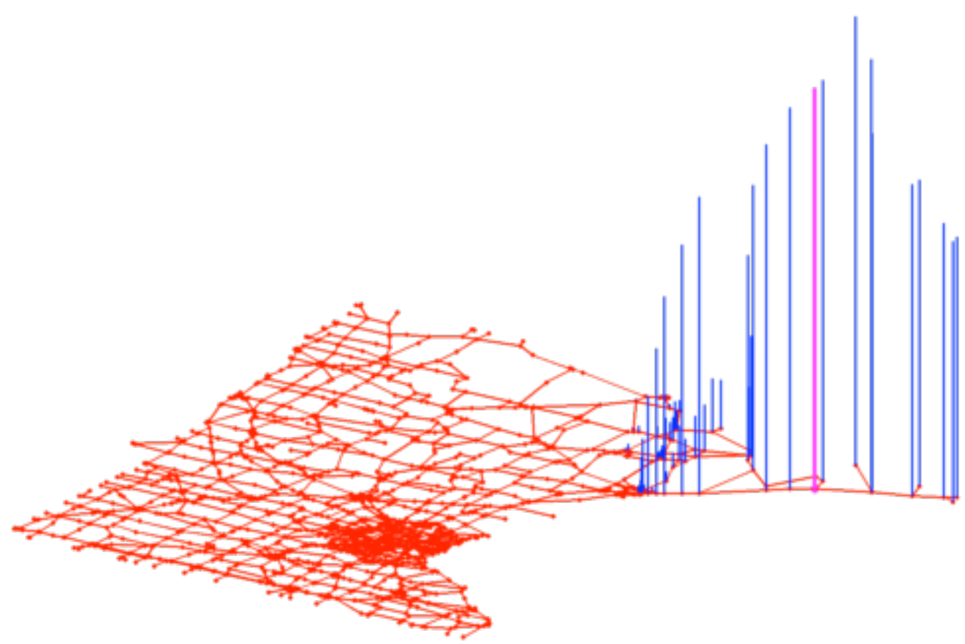
Hammond et al., Wavelets on graphs via spectral graph theory, ACHA, 2011

- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$
  - Graph setting: $(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$

Localization in action

Hammond et al., Wavelets on graphs via spectral graph theory, ACHA, 2011

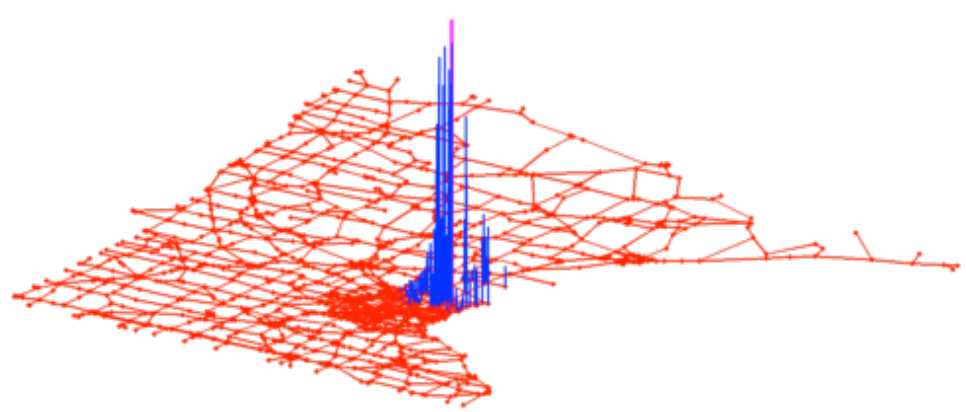
- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$
  - Graph setting: $(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$



Localization in action

Hammond et al., Wavelets on graphs via spectral graph theory, ACHA, 2011

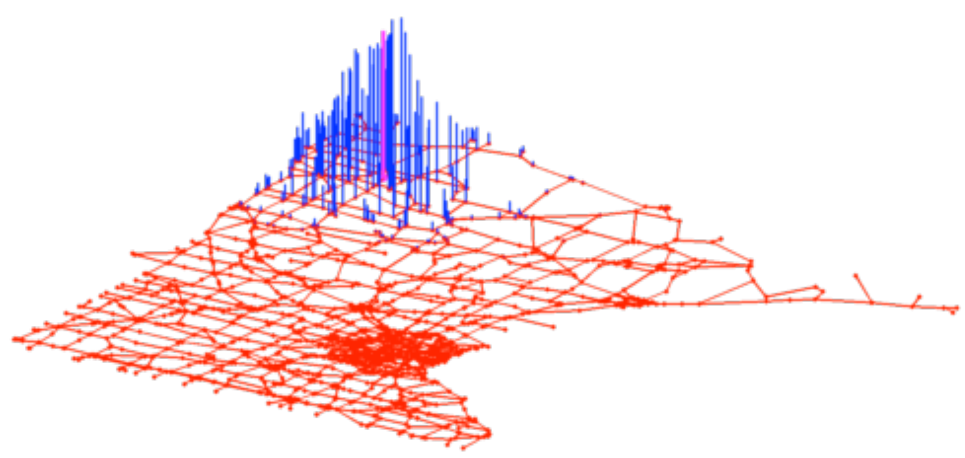
- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$
  - Graph setting: $(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$



Localization in action

Hammond et al., Wavelets on graphs via spectral graph theory, ACHA, 2011

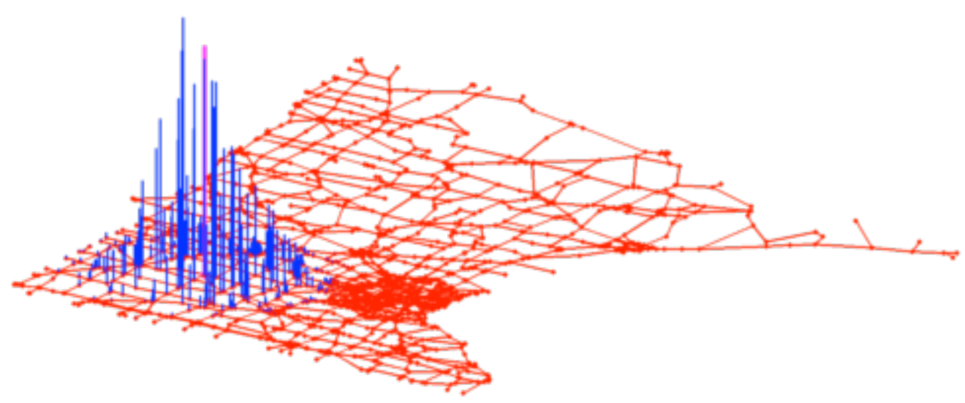
- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$
  - Graph setting: $(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$



Localization in action

Hammond et al., Wavelets on graphs via spectral graph theory, ACHA, 2011

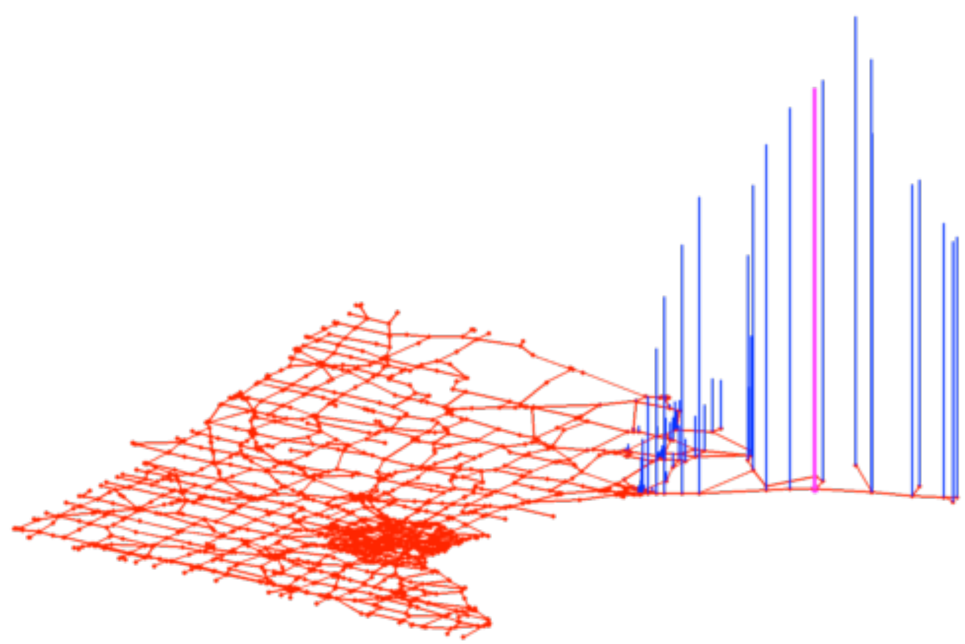
- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$
  - Graph setting: $(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$



Localization in action

Hammond et al., Wavelets on graphs via spectral graph theory, ACHA, 2011

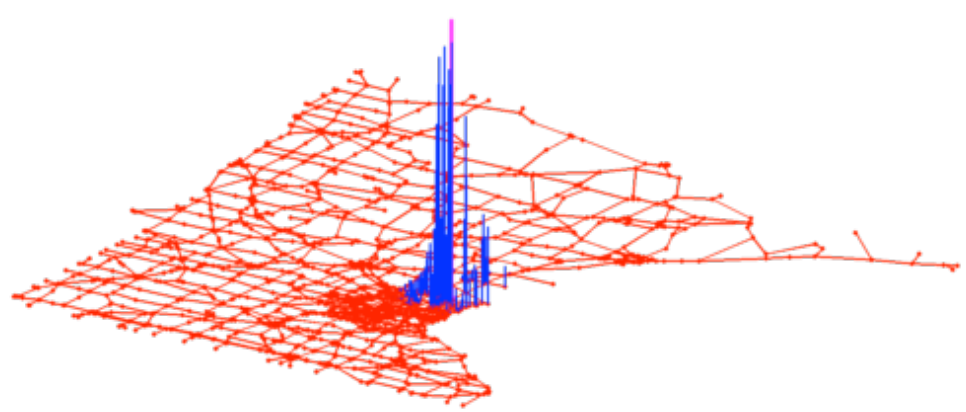
- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$
  - Graph setting: $(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$



Localization in action

Hammond et al., Wavelets on graphs via spectral graph theory, ACHA, 2011

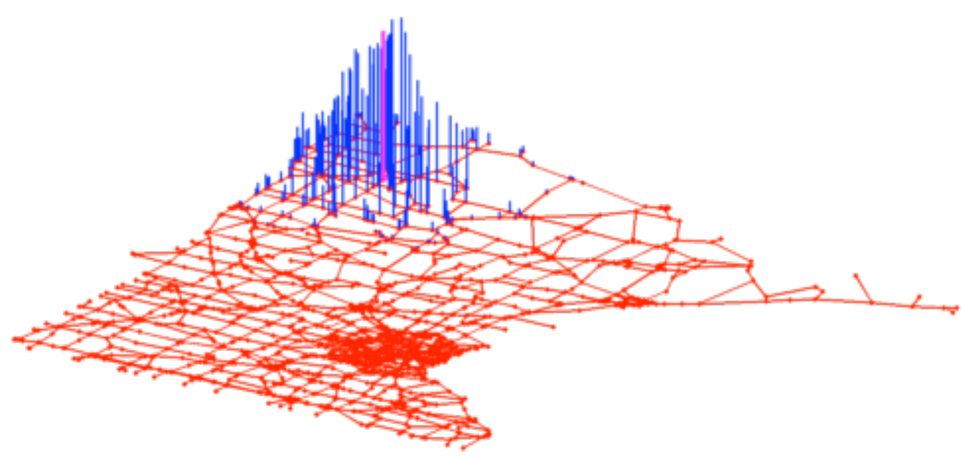
- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$
  - Graph setting: $(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$



Localization in action

Hammond et al., Wavelets on graphs via spectral graph theory, ACHA, 2011

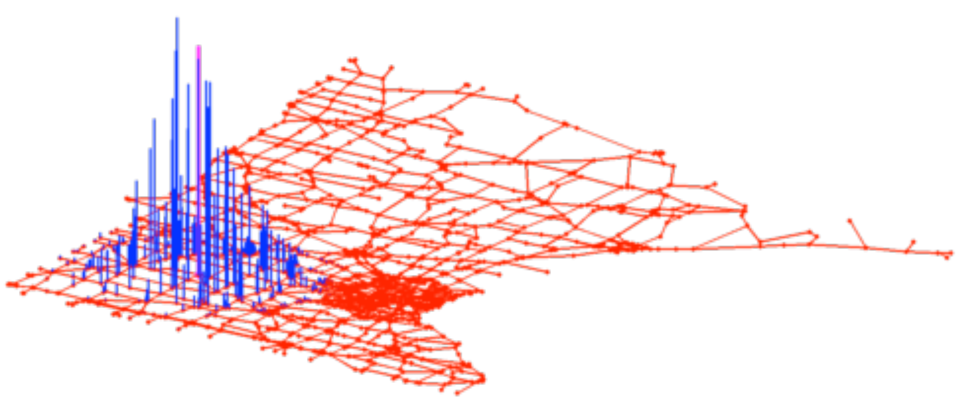
- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$
  - Graph setting: $(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$



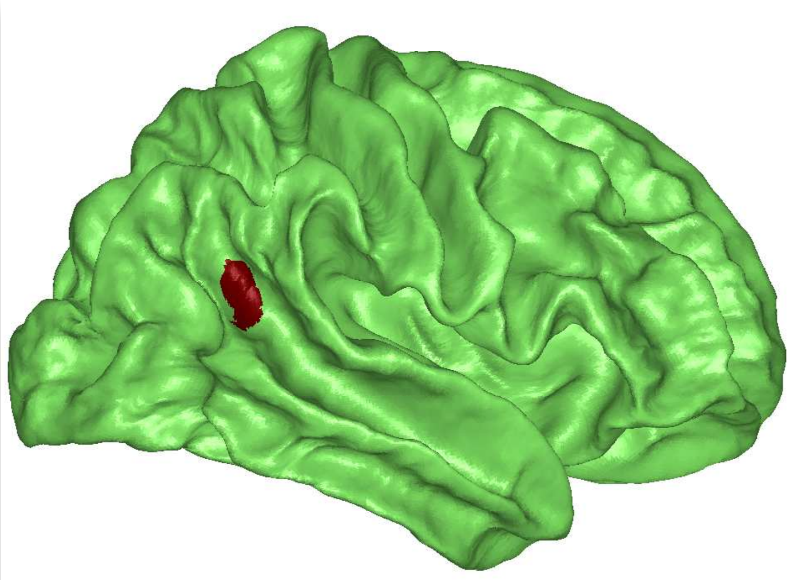
Localization in action

Hammond et al., Wavelets on graphs via spectral graph theory, ACHA, 2011

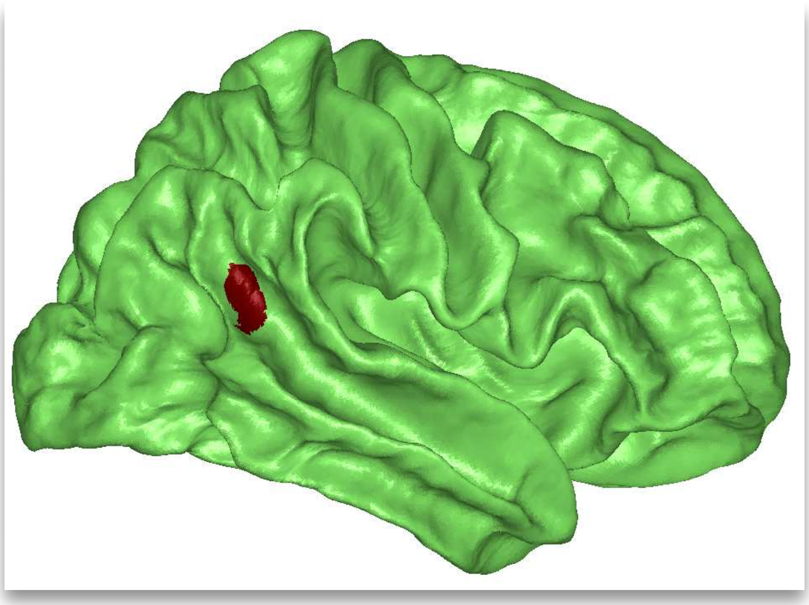
- Generalized translation
  - Classical setting: $(T_s g)(t) = g(t-s) = \int_{\mathbb{R}} \hat{g}(\xi) e^{-2\pi i \xi s} e^{2\pi i \xi t} d\xi$
  - Graph setting: $(T_n g)(i) := \sum_{\ell=0}^{N-1} \hat{g}(\lambda_\ell) u_\ell^*(n) u_\ell(i)$

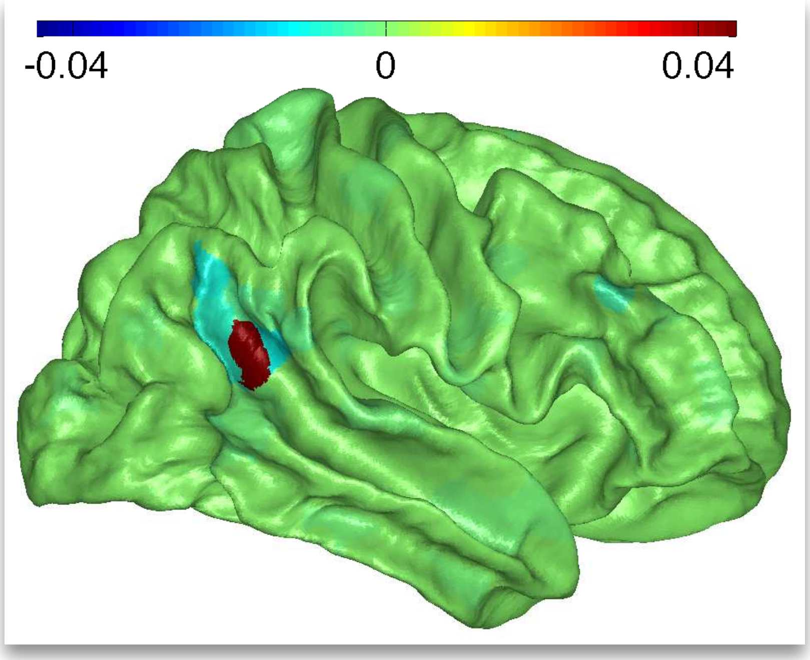


Localization in action

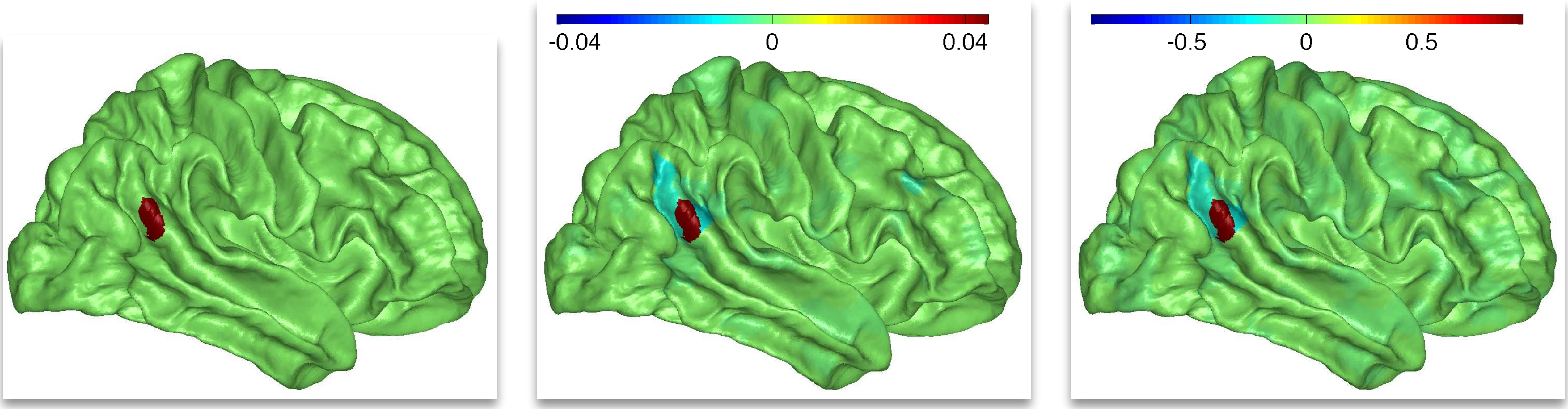


Localization in action

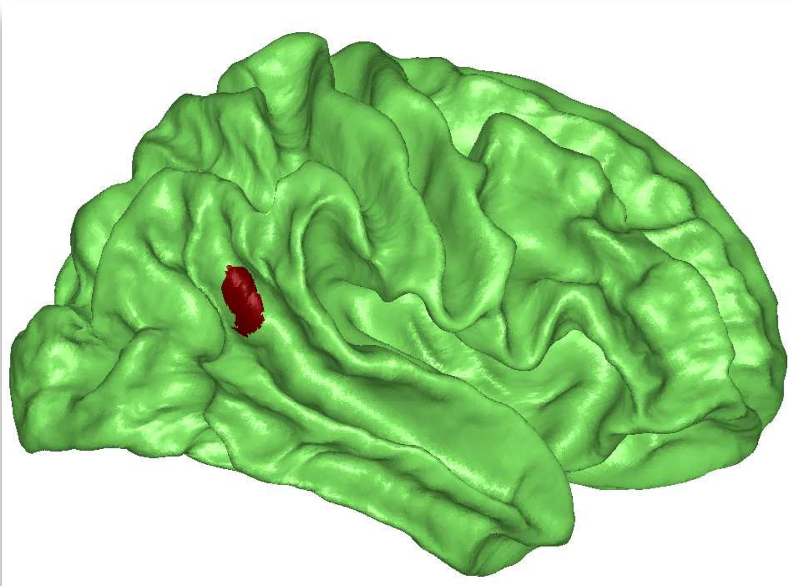


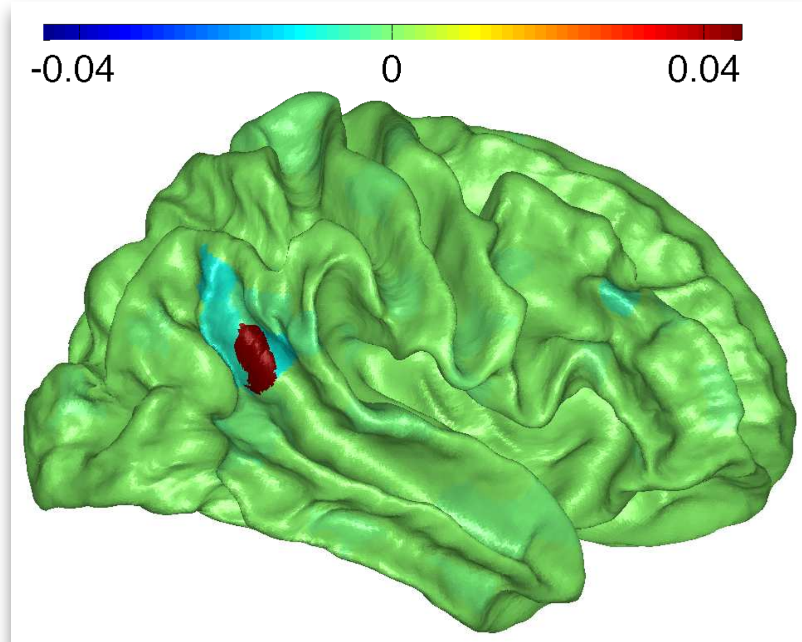


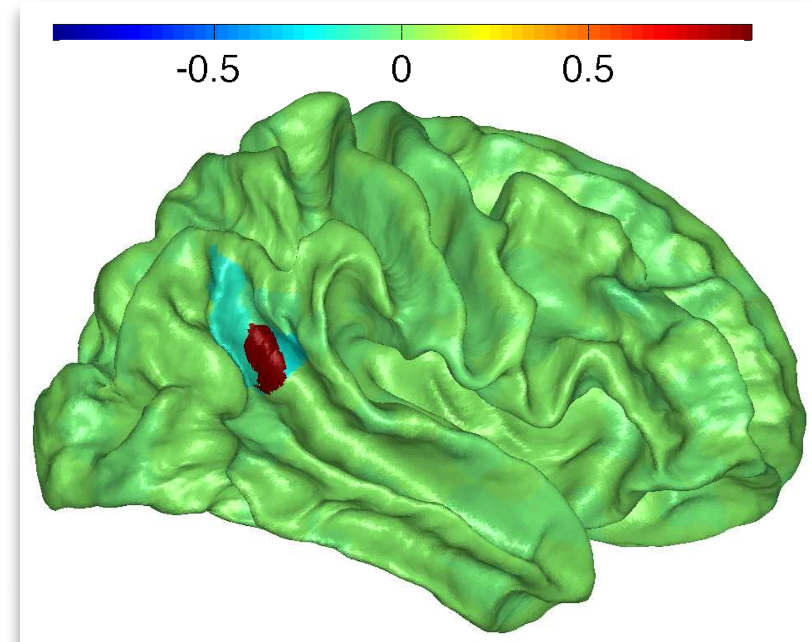
Localization in action

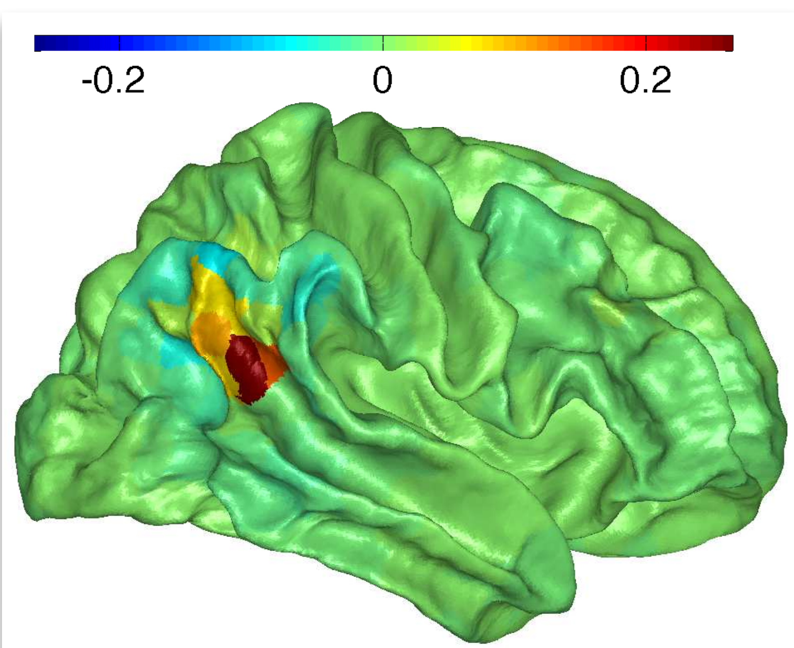


Localization in action

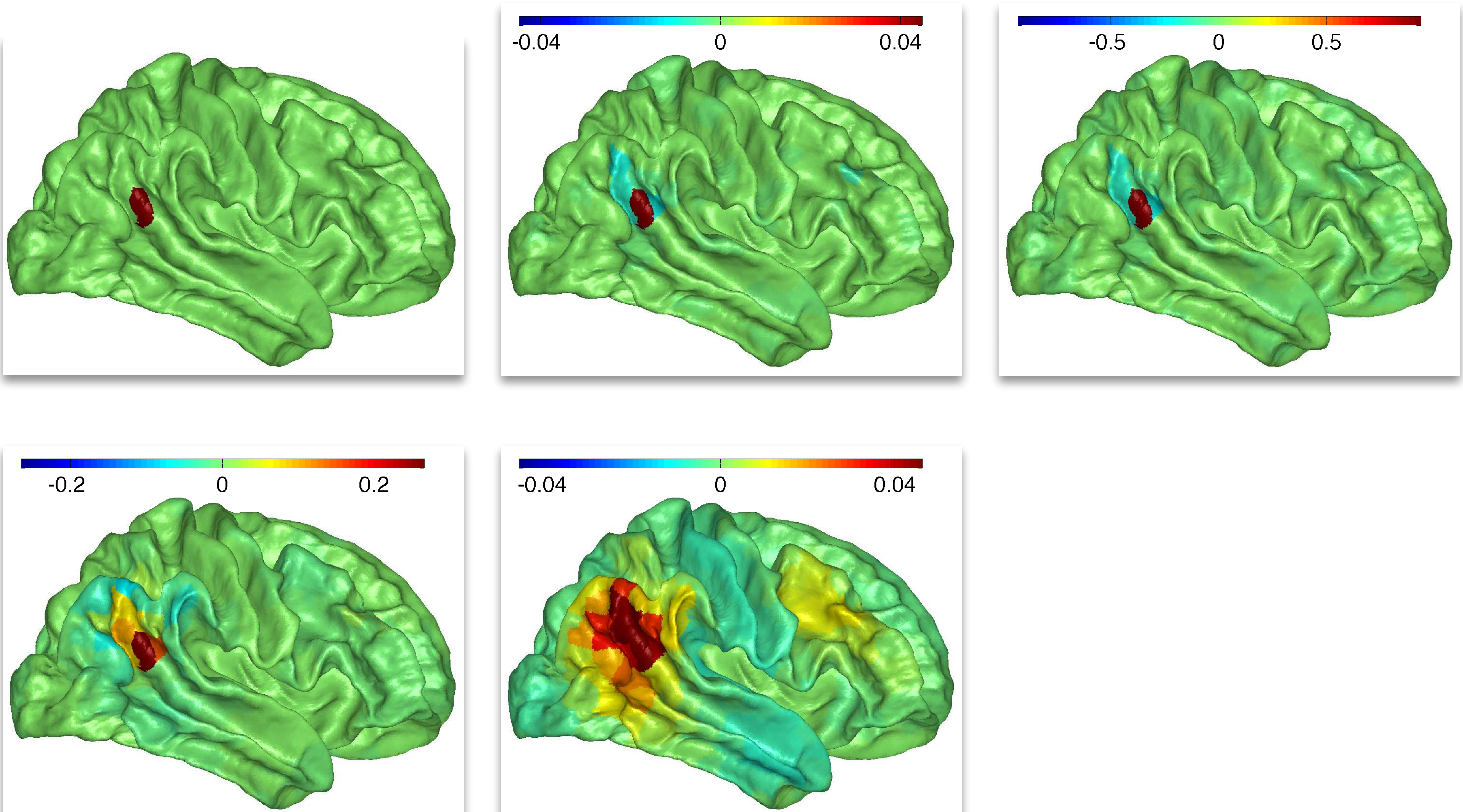




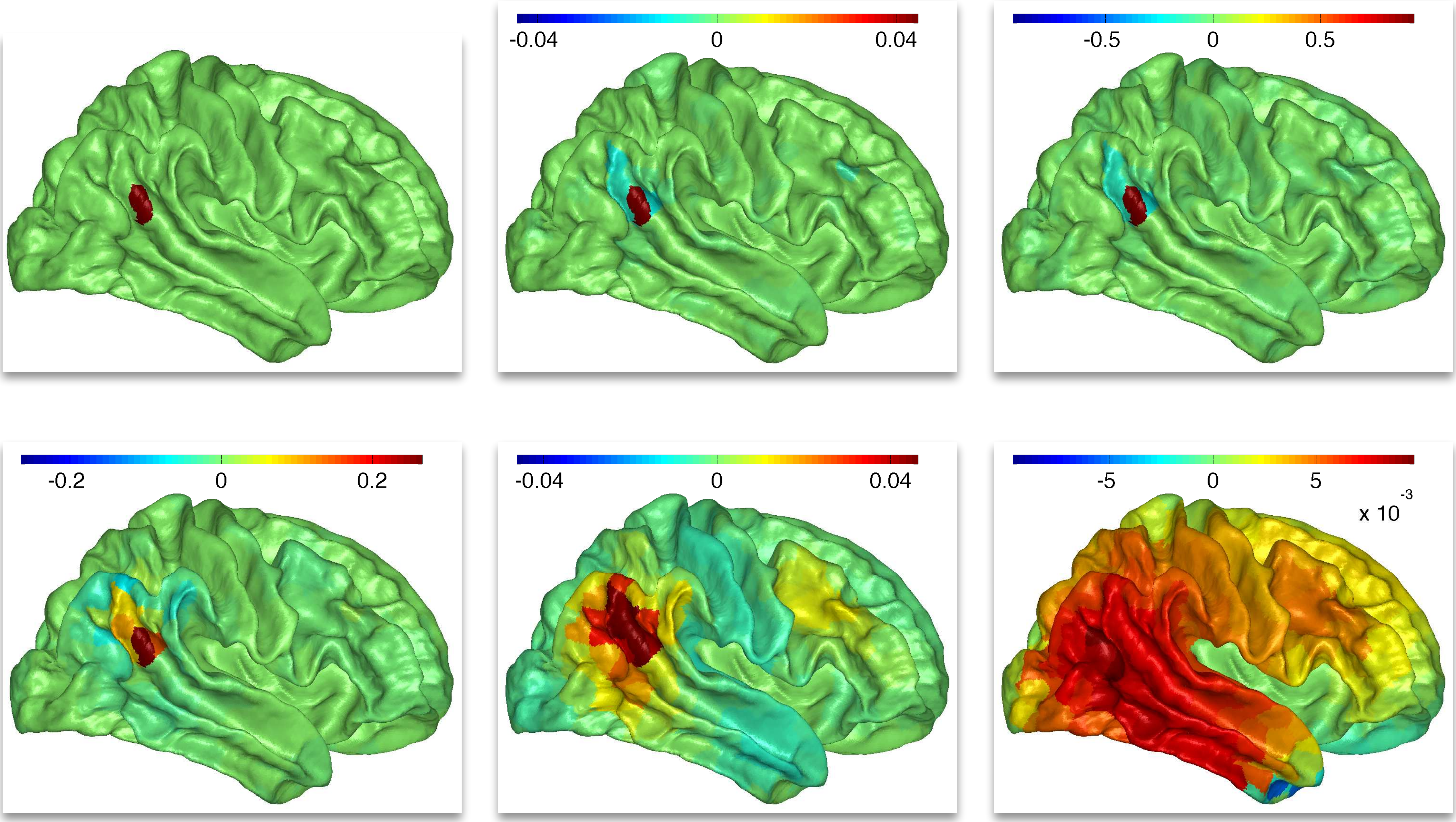




Localization in action

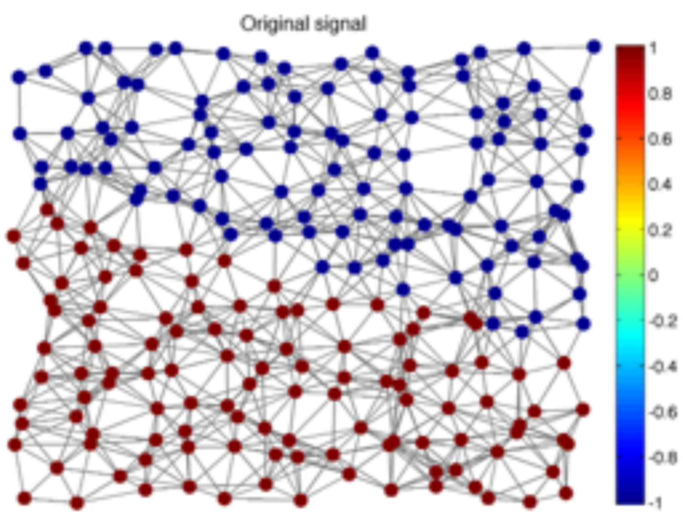


Localization in action

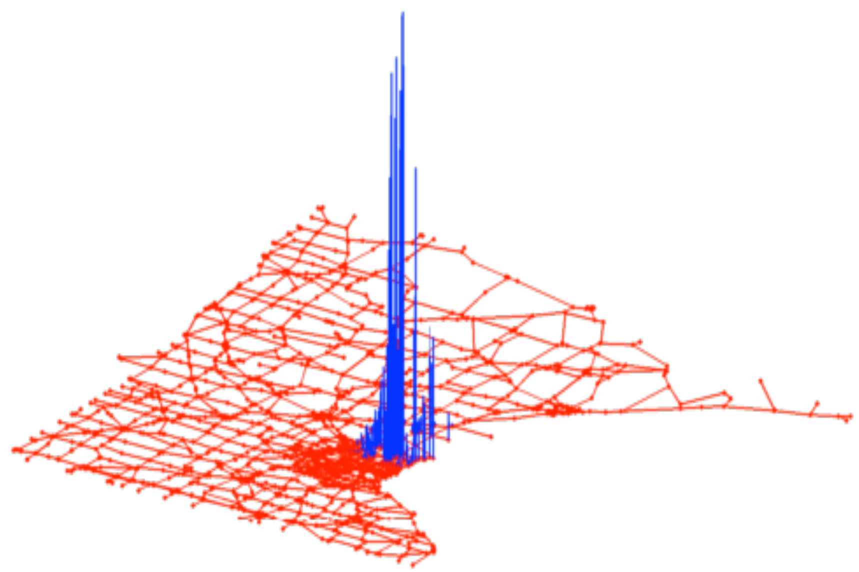


Simple De-Noising with Wavelets

$$\mathrm{argmin}_f \left\{ ||f - y||_2^2 + \gamma f^T \mathcal{L} f \right\}$$



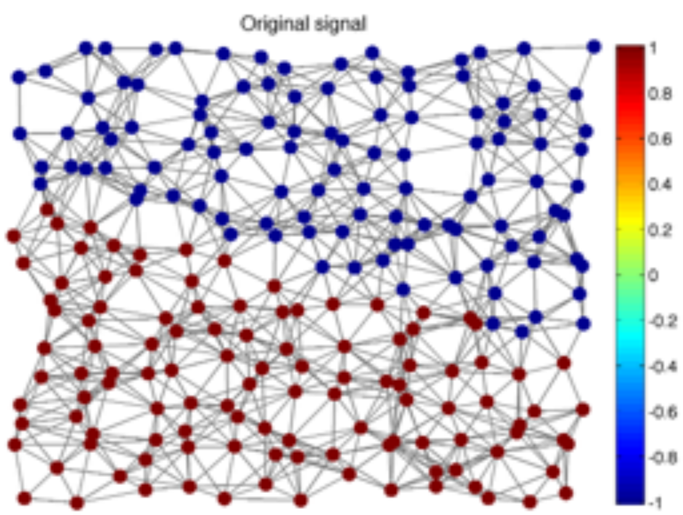
Original



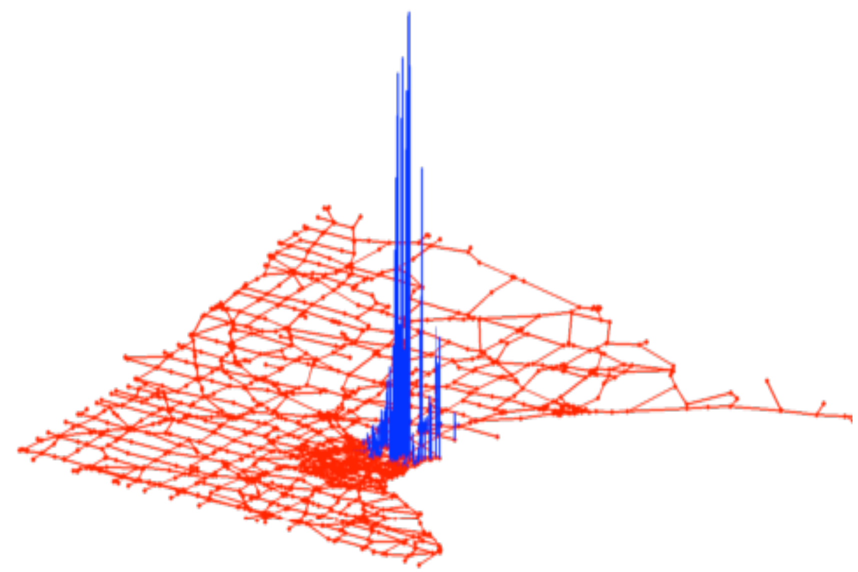
Noisy

Simple De-Noising with Wavelets

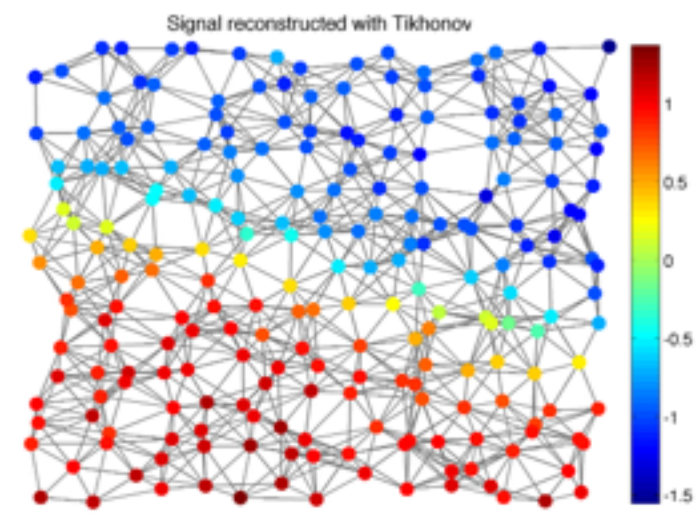
$$\text{argmin}_f \left\{ ||f - y||_2^2 + \gamma f^T \mathcal{L} f \right\}$$



Original



Noisy

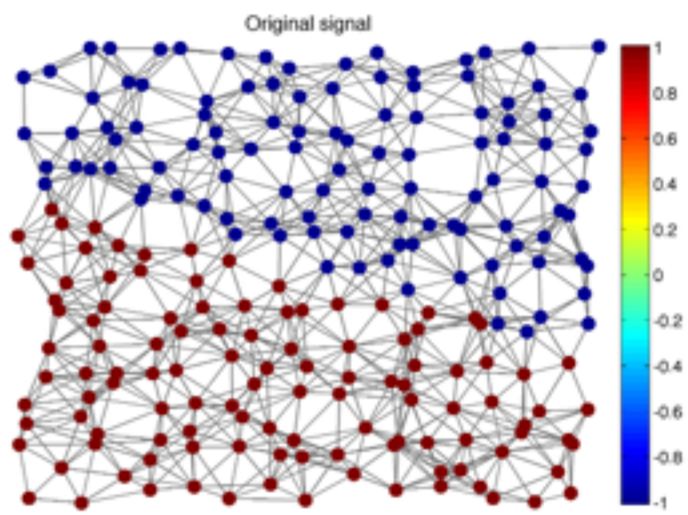


Denoised

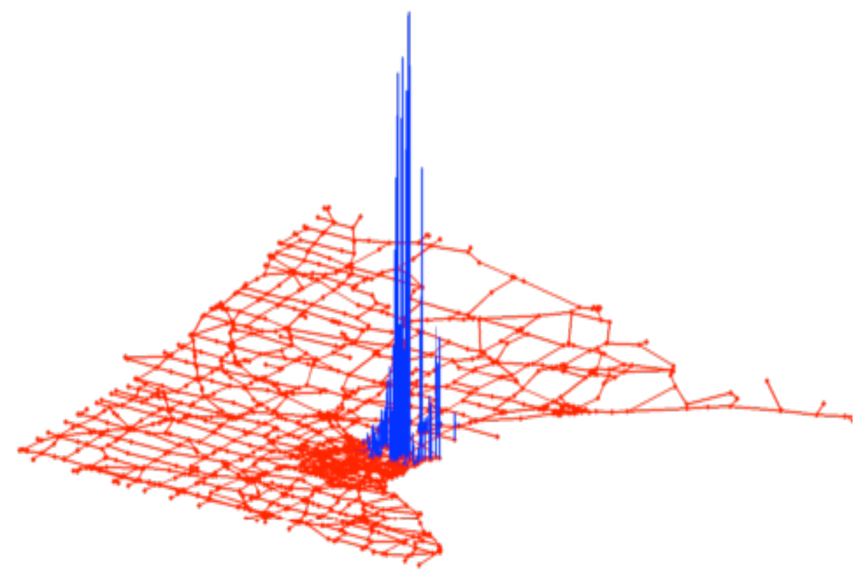
$$\text{argmin}_a \left\{ ||f - W^* a||_2^2 + \gamma ||a||_{1,\mu} \right\}$$

Simple De-Noising with Wavelets

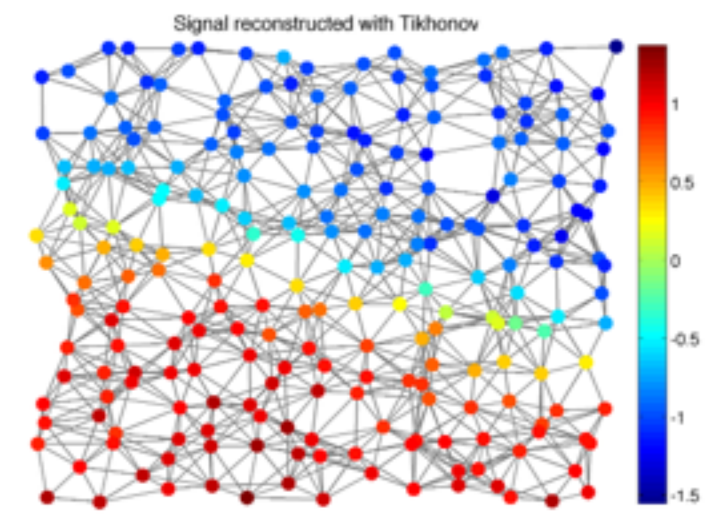
$$\operatorname{argmin}_f \left\{ ||f - y||_2^2 + \gamma f^T \mathcal{L} f \right\}$$



Original

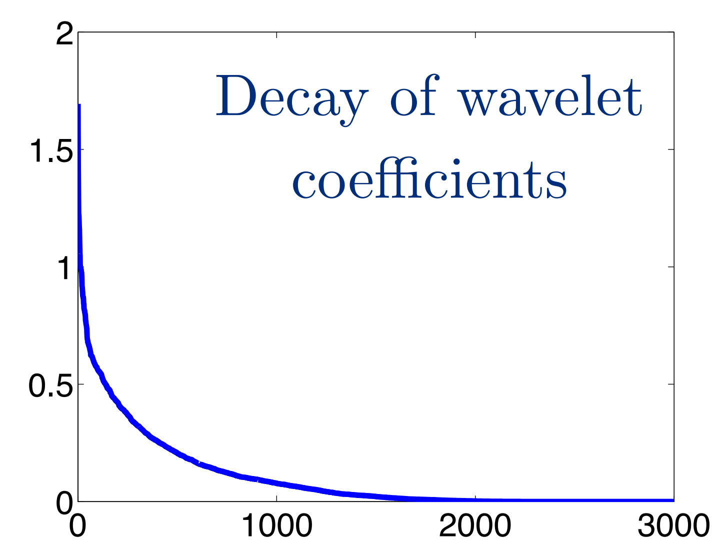


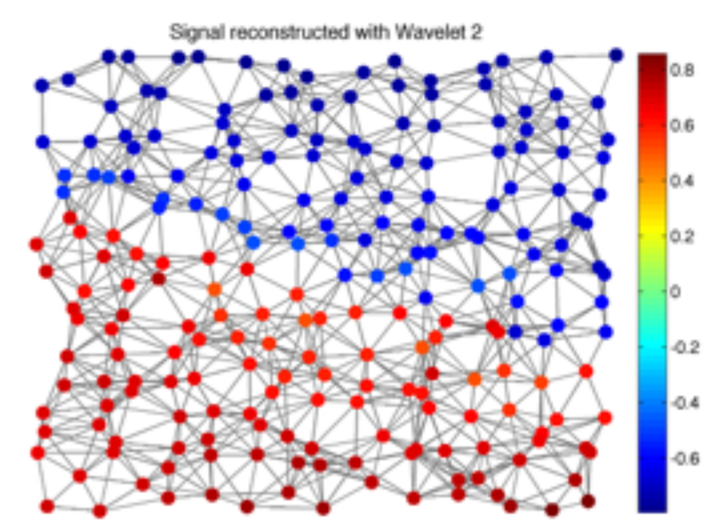
Noisy



Denoised

$$\operatorname{argmin}_a \left\{ ||f - W^* a||_2^2 + \gamma ||a||_{1,\mu} \right\}$$



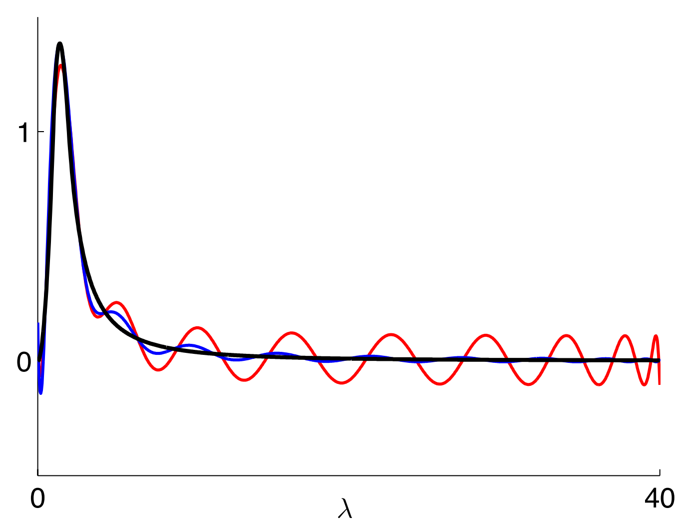


Remark on Implementation

Not necessary to compute spectral decomposition for filtering

Polynomial approximation : $$g(t\omega) \simeq \sum_{k=0}^{K-1} a_k(t) p_k(\omega)$$

ex: Chebyshev, minimax



$$\tilde{W}_f(t_n, j) = \left(\frac{1}{2} c_{n,0} f^{\#} + \sum_{k=1}^{M_n} c_{n,k} \overline{T}_k(\mathcal{L}) f^{\#} \right)_j$$

$$\overline{T}_k(\mathcal{L}) f = \frac{2}{a_1} (\mathcal{L} - a_2 I) \left(\overline{T}_{k-1}(\mathcal{L}) f \right) - \overline{T}_{k-2}(\mathcal{L}) f$$

Computational cost dominated by matrix-vector multiply with (sparse) Laplacian matrix. In particular $O(\sum_{n=1}^{J} M_n |E|)$

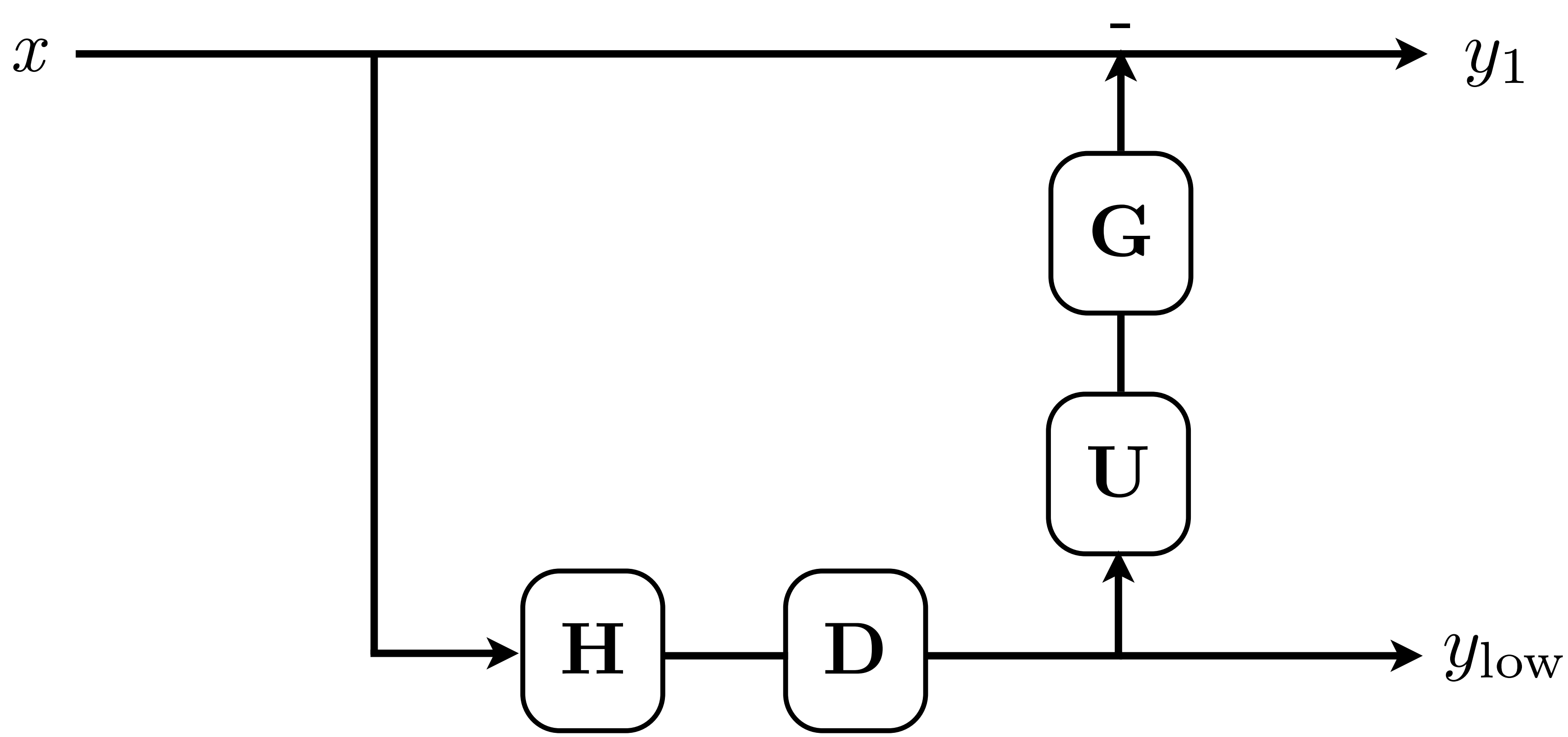
http://wiki.epfl.ch/sgwt

LTS EPFL LTS2

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

The Laplacian Pyramid

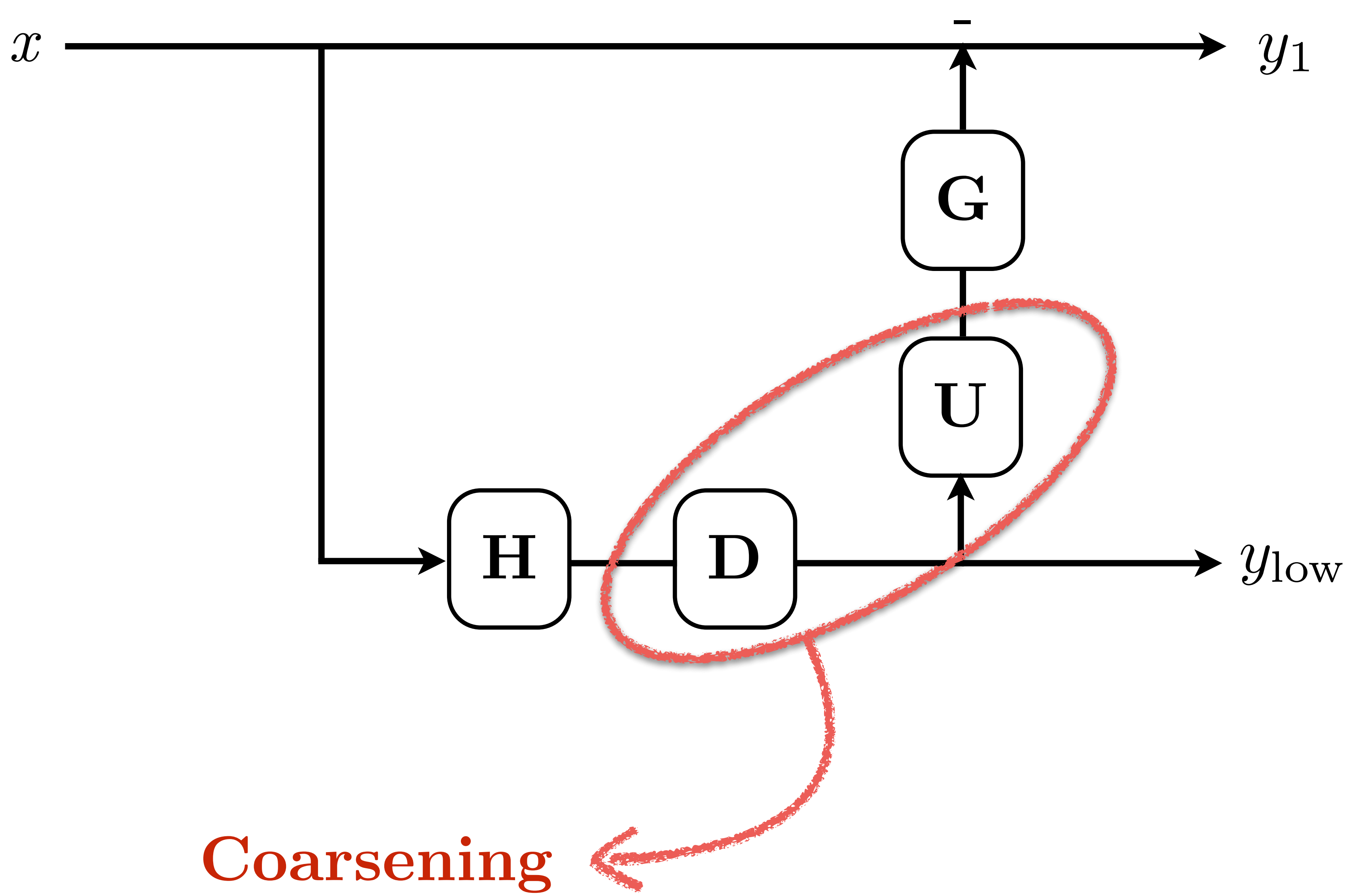
Analysis operator



Shuman, Faraji, VDG, A framework for multiscale transforms on graphs, 2013

The Laplacian Pyramid

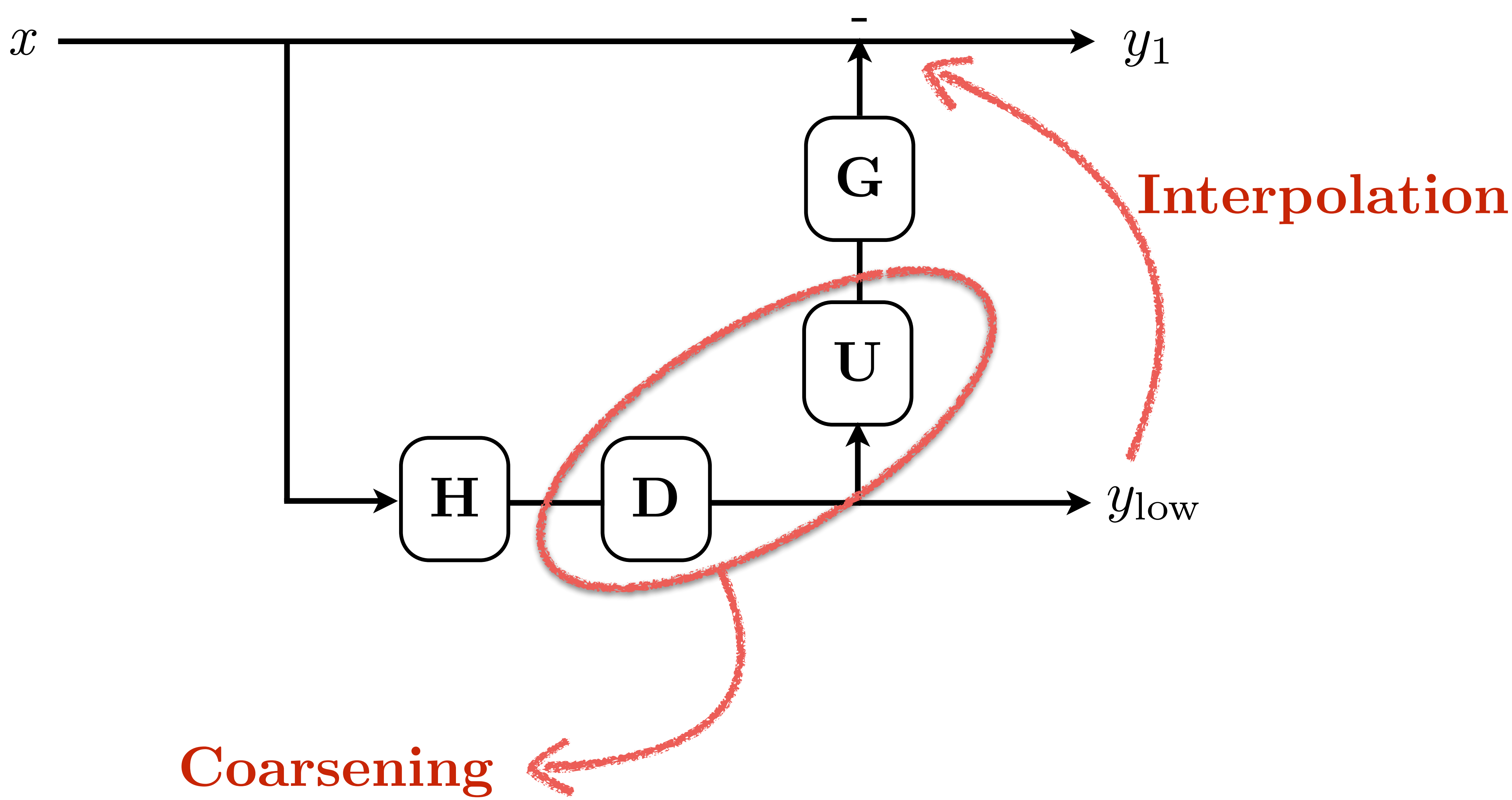
Analysis operator



Shuman, Faraji, VDG, A framework for multiscale transforms on graphs, 2013

The Laplacian Pyramid

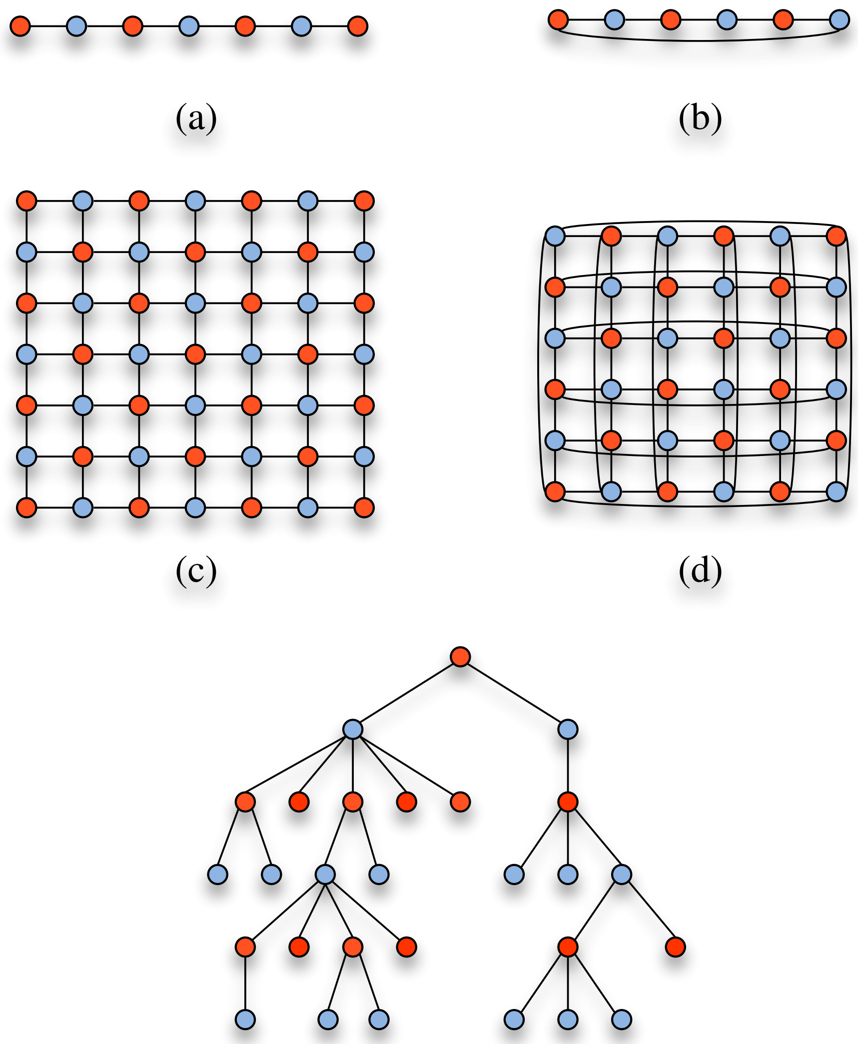
Analysis operator



Shuman, Faraji, VDG, A framework for multiscale transforms on graphs, 2013

Downsampling

$$\mathcal{V}_1 = \mathcal{V}_+ := \{i \in \mathcal{V} : u_{\max}(i) \geq 0\}$$


(a) (b) (c) (d) (e)

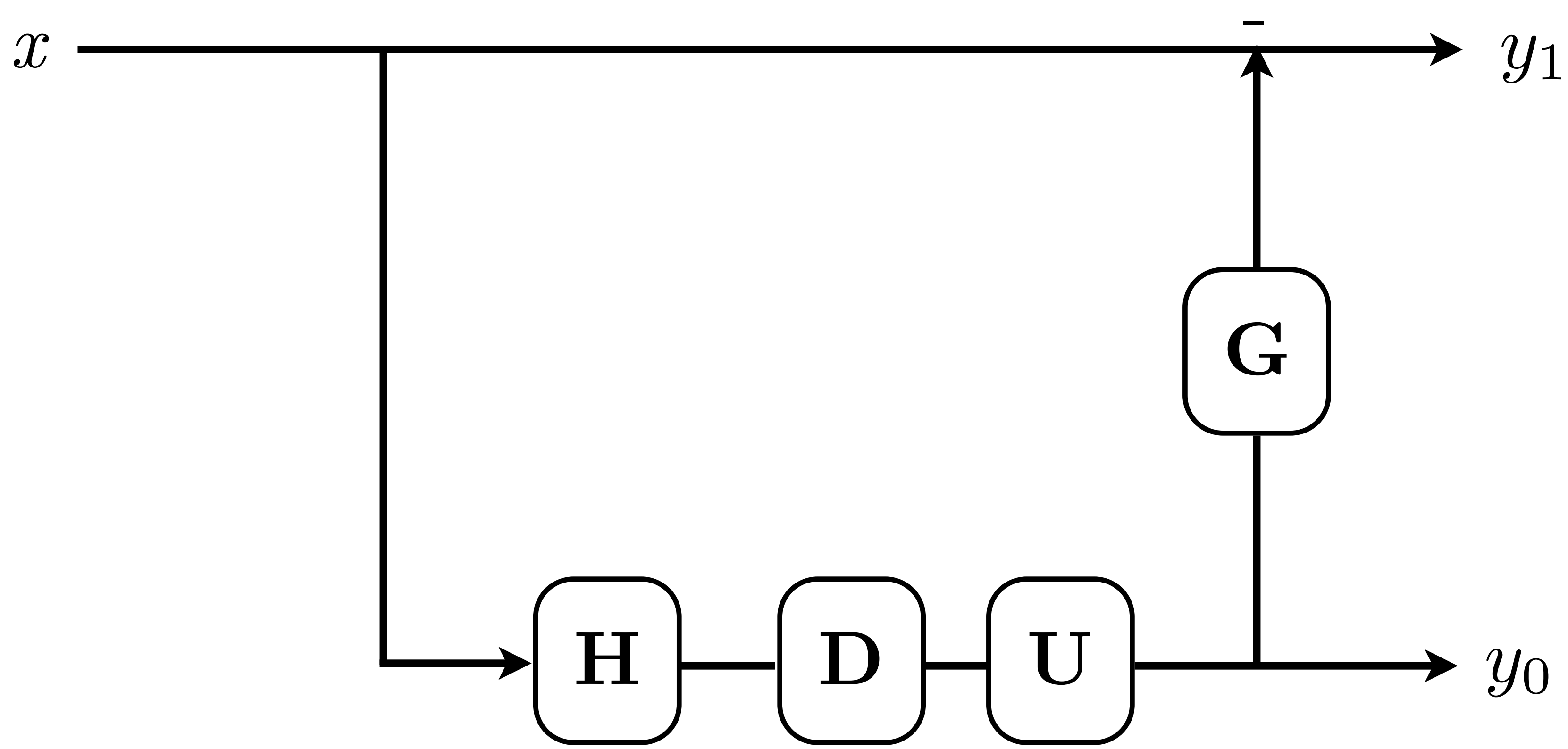
Relaxed solution to 2-coloring for regular graphs

Exact for bipartite graphs

Connections with nodal domains theory for laplacian eigenvectors

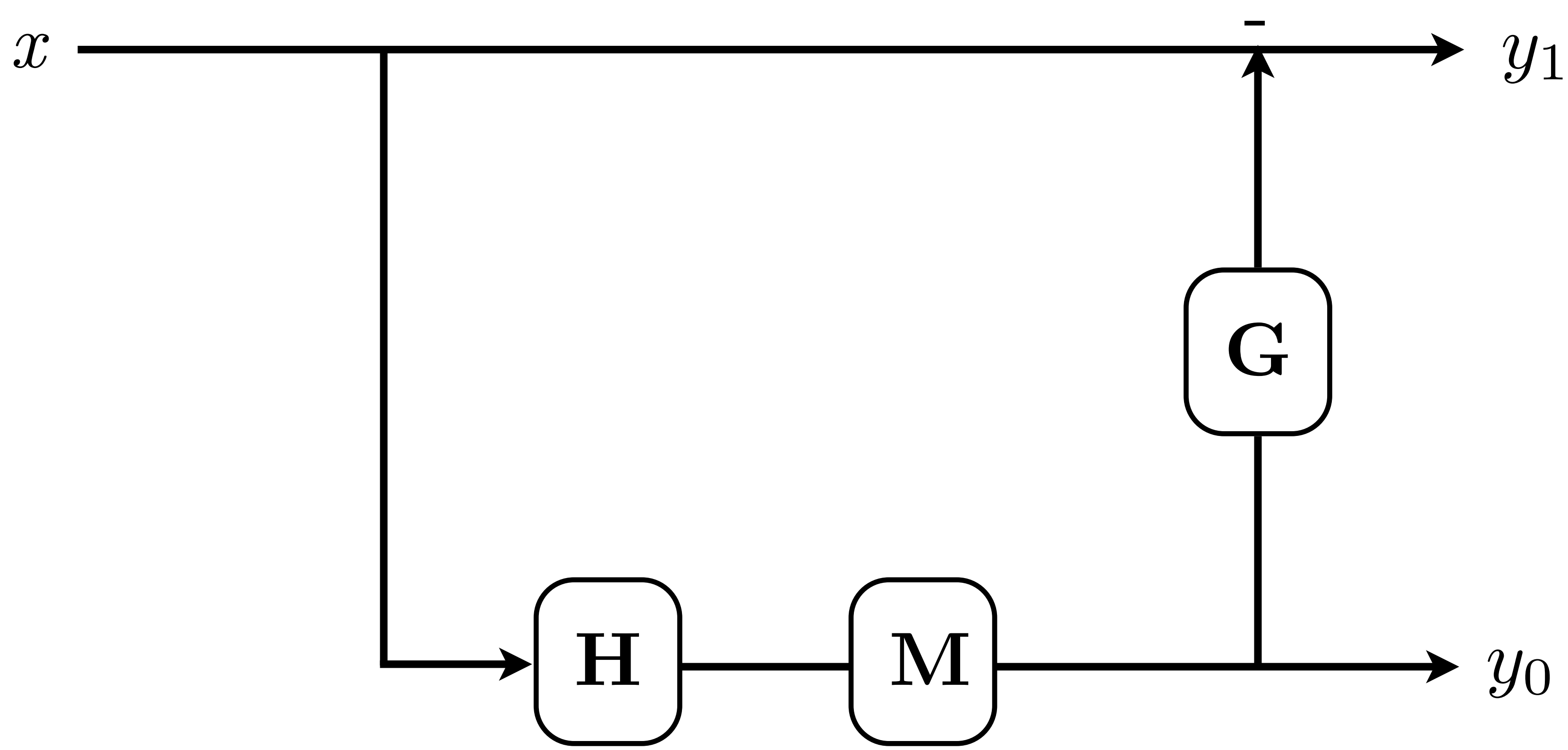
The Laplacian Pyramid

Analysis operator



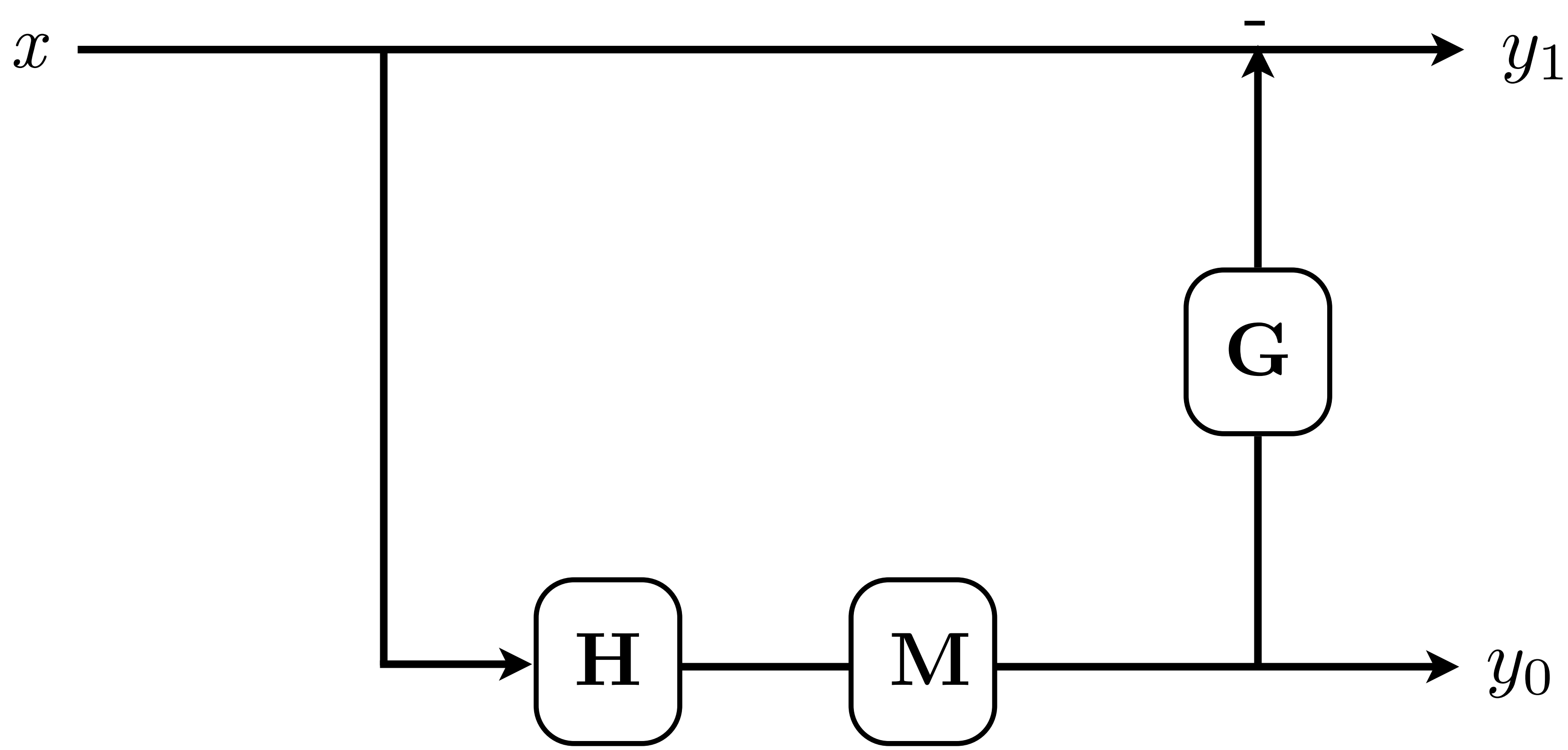
The Laplacian Pyramid

Analysis operator



The Laplacian Pyramid

Analysis operator

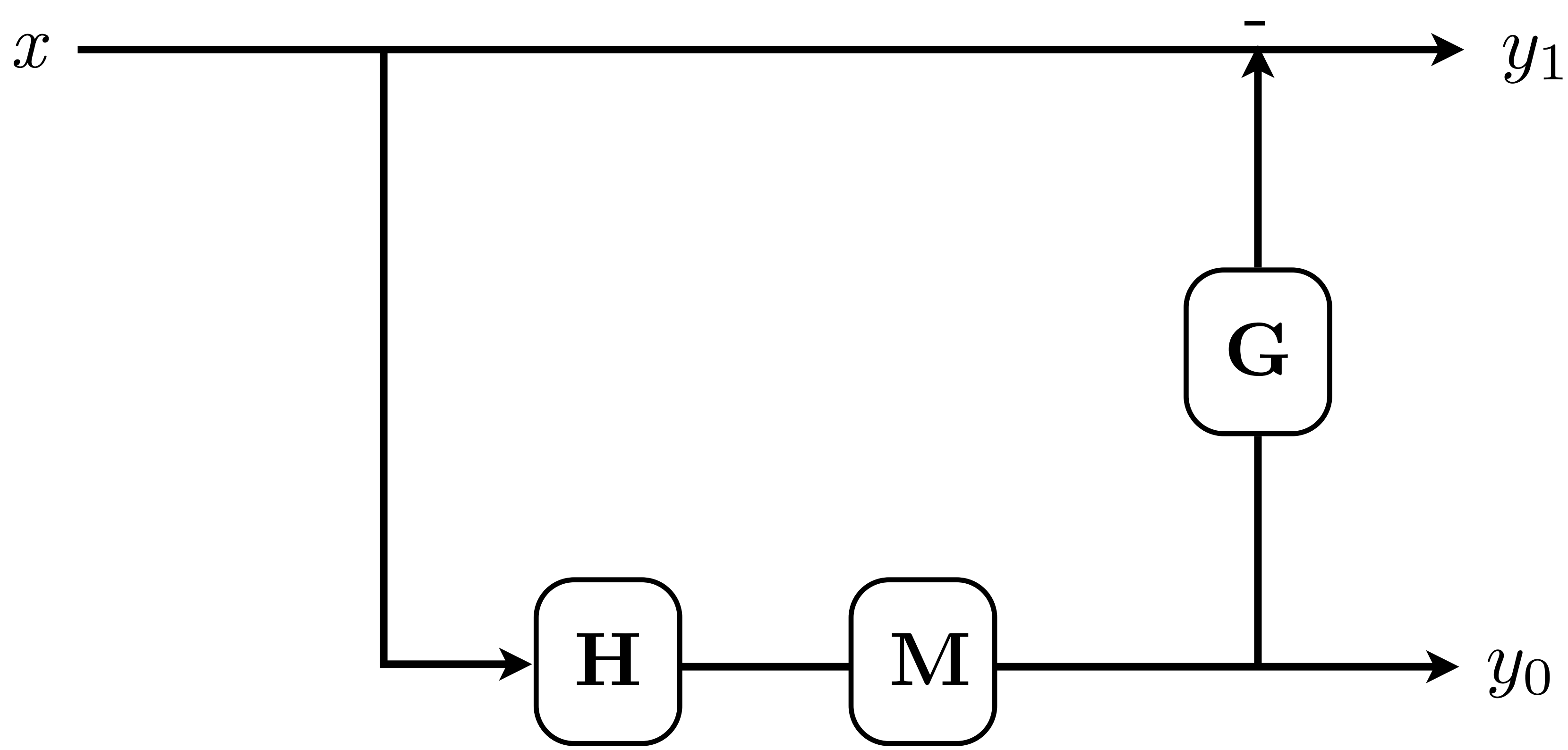


$$y_0 = \mathbf{H_m} x = \mathbf{MH} x$$

$$y_1 = x - \mathbf{G} y_0 = x - \mathbf{GH_m} x$$

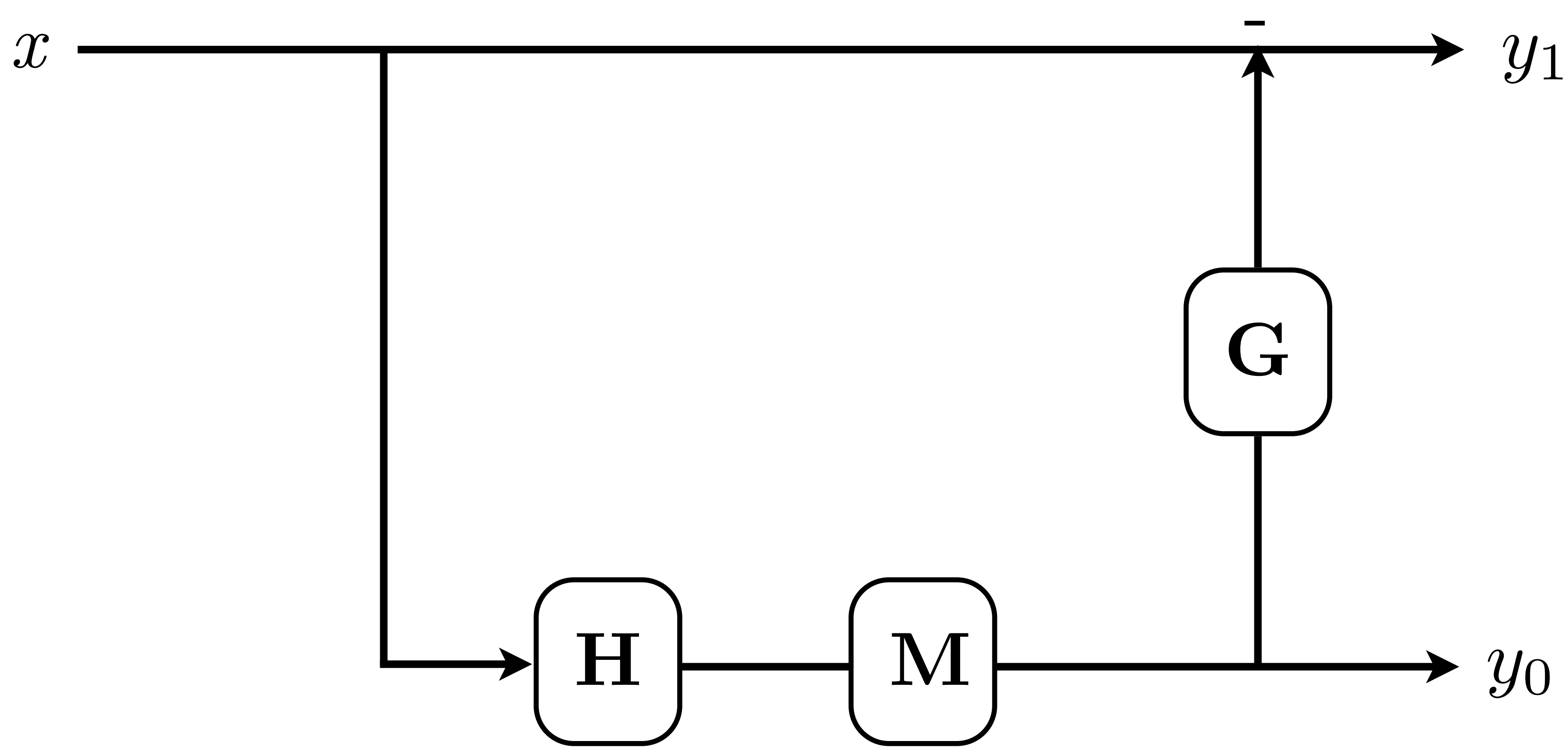
The Laplacian Pyramid

Analysis operator



The Laplacian Pyramid

Analysis operator



$$\underbrace{\begin{pmatrix} y_0 \\ y_1 \end{pmatrix}}_{y} = \underbrace{\begin{pmatrix} \mathbf{H_m} \\ \mathbf{I} - \mathbf{GH_m} \end{pmatrix}}_{\mathbf{T_a}} x,$$

The Laplacian Pyramid

Analysis operator

$$\underbrace{\begin{pmatrix} y_0 \\ y_1 \end{pmatrix}}_{y} = \underbrace{\begin{pmatrix} \mathbf{H_m} \\ \mathbf{I} - \mathbf{G}\mathbf{H_m} \end{pmatrix}}_{\mathbf{T_a}} x,$$

Do, Vetterli, Framing Pyramids, IEEE TSP, 2003

The Laplacian Pyramid

Analysis operator

$$\underbrace{\begin{pmatrix} y_0 \\ y_1 \end{pmatrix}}_{y} = \underbrace{\begin{pmatrix} \mathbf{H_m} \\ \mathbf{I} - \mathbf{G}\mathbf{H_m} \end{pmatrix}}_{\mathbf{T_a}} x,$$

Simple (traditional) left inverse

$$\hat{x} = \underbrace{\begin{pmatrix} \mathbf{G} & \mathbf{I} \end{pmatrix}}_{\mathbf{T_s}} \underbrace{\begin{pmatrix} y_0 \\ y_1 \end{pmatrix}}_{y}$$

$\mathbf{T_s}\mathbf{T_a} = \mathbf{I}$ with no conditions on $\mathbf{H}$ or $\mathbf{G}$

Do, Vetterli, Framing Pyramids, IEEE TSP, 2003

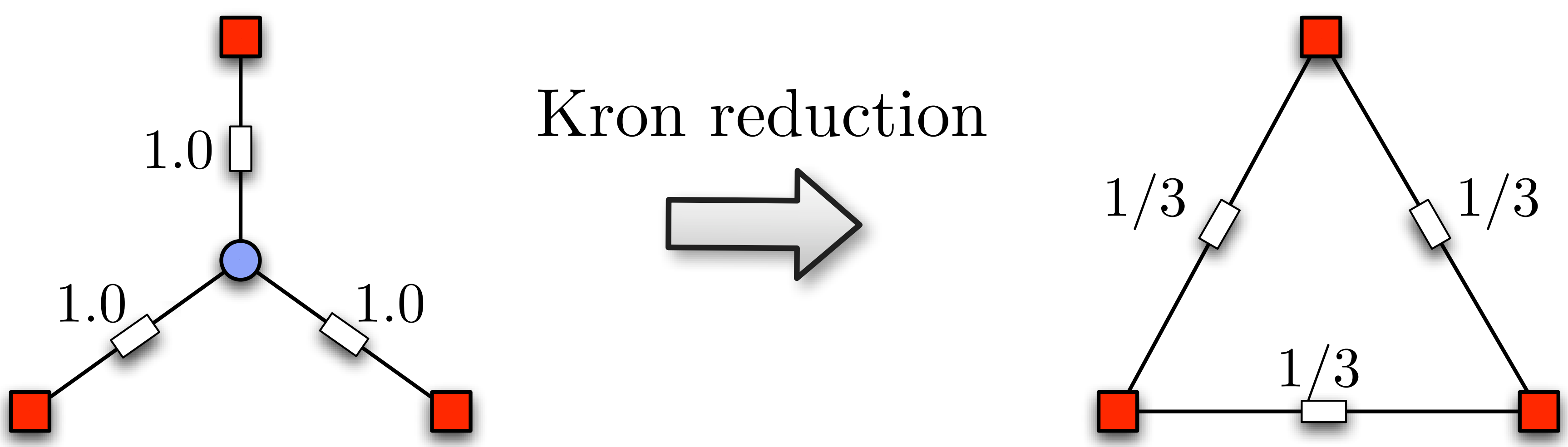
Coarsening by Kron Reduction

In order to iterate the construction, we need to construct a graph on the reduced vertex set.

$$\mathbf{A}_{\mathrm{r}} = \mathbf{A}[\alpha,\alpha] - \mathbf{A}[\alpha,\alpha)\mathbf{A}(\alpha,\alpha)^{-1}\mathbf{A}(\alpha,\alpha]$$

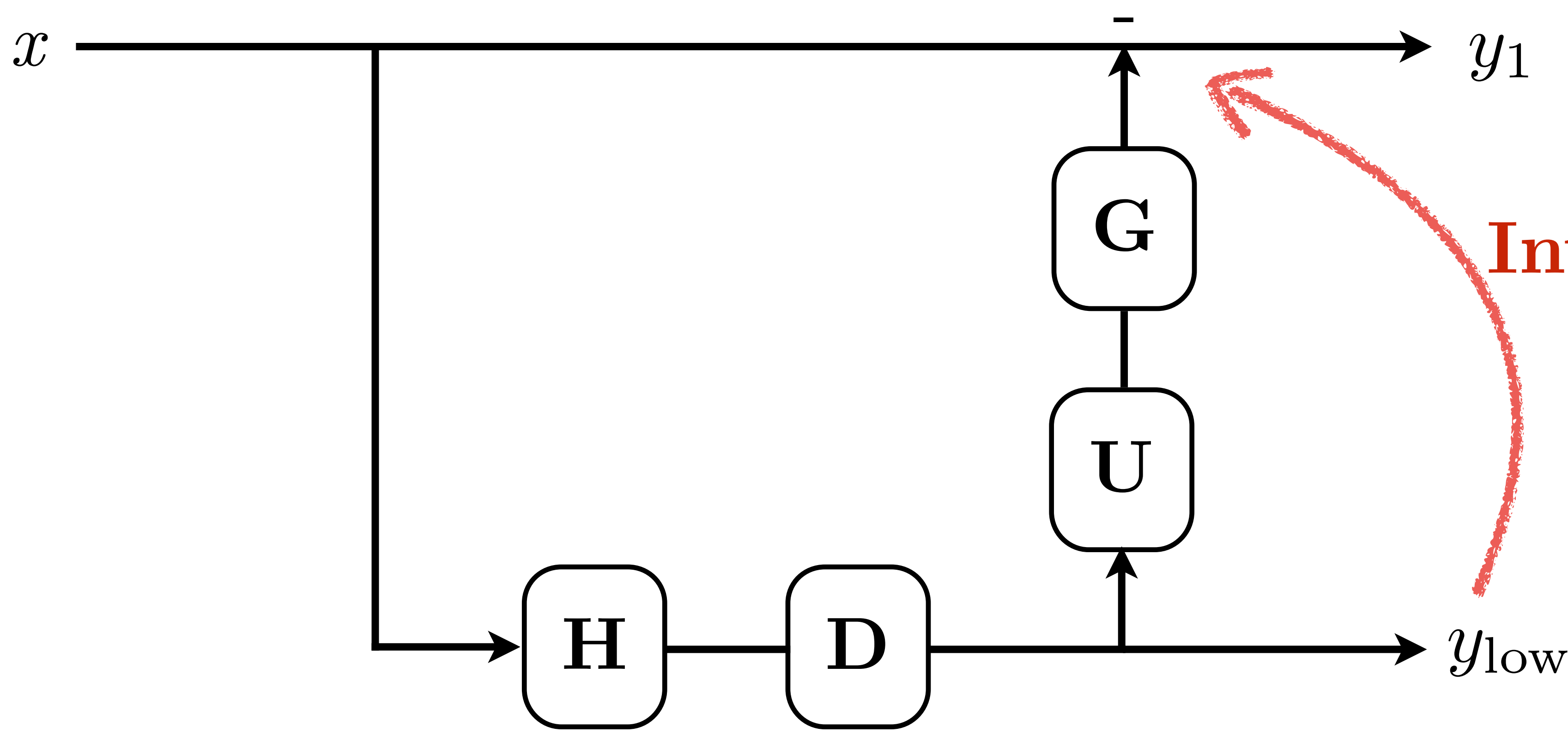
Schur complement

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}[\alpha,\alpha] & \mathbf{A}[\alpha,\alpha) \\ \mathbf{A}(\alpha,\alpha] & \mathbf{A}(\alpha,\alpha) \end{bmatrix}$$



Dorfler et al., ArXiV, 2011

Reduction-aware interpolation



Idea: Optimize interpolation for reduction:

$$y[u] = \sum_{v \in V_1} \alpha[v] \varphi^v[u]$$ Shifted Green's function of L at vertex v

$$y[v'] = \sum_{v \in V_1} \alpha[v] \varphi^v[v'] = x[v'] \quad \forall v' \in V_1$$

Spline-like interpolation

Simple linear model:

$$f_{\text{interp}}(i) = \sum_{j \in \mathcal{V}_r} \alpha[j] \varphi_j(i) \qquad\qquad f_{\text{interp}} = \mathbf{\Phi}\alpha$$

With: $\varphi_j(i) = \big(T_j \varphi\big)(i) \qquad\qquad \mathbf{\Phi}[i,j] = \varphi_i(j)$

Spline-like interpolation

Simple linear model:

$$f_{\text{interp}}(i) = \sum_{j \in \mathcal{V}_r} \alpha[j] \varphi_j(i) \qquad f_{\text{interp}} = \mathbf{\Phi} \alpha$$

With: $\varphi_j(i) = \big(T_j \varphi\big)(i) \qquad \mathbf{\Phi}[i, j] = \varphi_i(j)$

Interpolation condition:

On the known vertices: $f_r = \mathbf{\Phi}_{\mathcal{V}_r} \alpha$

Solution depends on efficient, robust inversion of: $\alpha = \mathbf{\Phi}_{\mathcal{V}_r}^{-1} f_r$

Those weights can be computed using only filtering !

Spline-like interpolation

Regularized Laplacian: $\tilde{\mathcal{L}} = \mu^{-1}\mathcal{L} + \mathbf{I}_{|\mathcal{V}|}$

Stable pseudo-inverse: $\tilde{\mathcal{L}}^{-1}[i,j] = \sum_{\ell=0}^{|\mathcal{V}|-1} \frac{1}{1+\mu^{-1}\lambda_\ell} u_\ell(i) u_\ell(j)$

Spline-like interpolation

Regularized Laplacian: $\tilde{\mathcal{L}} = \mu^{-1}\mathcal{L} + \mathbf{I}_{|\mathcal{V}|}$

Stable pseudo-inverse: $\tilde{\mathcal{L}}^{-1}[i,j] = \sum_{\ell=0}^{|\mathcal{V}|-1} \frac{1}{1+\mu^{-1}\lambda_\ell} u_\ell(i) u_\ell(j)$

$$\varphi_j(i) = \sum_{\ell=0}^{|\mathcal{V}|-1} \frac{1}{1+\mu^{-1}\lambda_\ell} u_\ell(i) u_\ell(j)$$

Shifted Green's functions

Spline-like interpolation

Regularized Laplacian: $\tilde{\mathcal{L}} = \mu^{-1}\mathcal{L} + \mathbf{I}_{|\mathcal{V}|}$

Stable pseudo-inverse: $\tilde{\mathcal{L}}^{-1}[i,j] = \sum_{\ell=0}^{|\mathcal{V}|-1} \frac{1}{1+\mu^{-1}\lambda_\ell} u_\ell(i) u_\ell(j)$

$$\varphi_j(i) = \sum_{\ell=0}^{|\mathcal{V}|-1} \frac{1}{1+\mu^{-1}\lambda_\ell} u_\ell(i) u_\ell(j)$$

Shifted Green's functions

$$\begin{aligned} \tilde{\mathcal{L}}_r f_{\text{interp}}(i) &= \tilde{\mathcal{L}}_r f_r(i), \ \forall i \in \mathcal{V}_r \\ &= \sum_{j \in \mathcal{V}_r} \alpha[j] (\tilde{\mathcal{L}}_r \varphi_j)(i) \end{aligned}$$

Note: $$\begin{aligned} \tilde{\mathcal{L}}\varphi_j(i) &= \tilde{\mathcal{L}}\tilde{\mathcal{L}}^{-1}\delta_j(i) \\ &= \delta_j(i) \end{aligned}$$

Does this property carry over to the Kron reduced Laplacian?

Spline-like interpolation

Lemma: Inversion/Reduction commute for the (regularized) Laplacian

$$\left(\tilde{\mathcal{L}}^{-1}\right)_{\mathcal{V}_r} = \left(\tilde{\mathcal{L}}_r\right)^{-1}$$

This implies invariance of the Green's functions via reduction and therefore

$$\alpha = \tilde{\mathcal{L}}_r f_r \qquad\qquad f_{\text{interp}} = \mathbf{\Phi}\alpha$$

Spline-like interpolation

Lemma: Inversion/Reduction commute for the (regularized) Laplacian

$$\left(\tilde{\mathcal{L}}^{-1}\right)_{\mathcal{V}_r} = \left(\tilde{\mathcal{L}}_r\right)^{-1}$$

This implies invariance of the Green's functions via reduction and therefore

$$\alpha = \tilde{\mathcal{L}}_r f_r \qquad f_{\text{interp}} = \mathbf{\Phi}\alpha$$

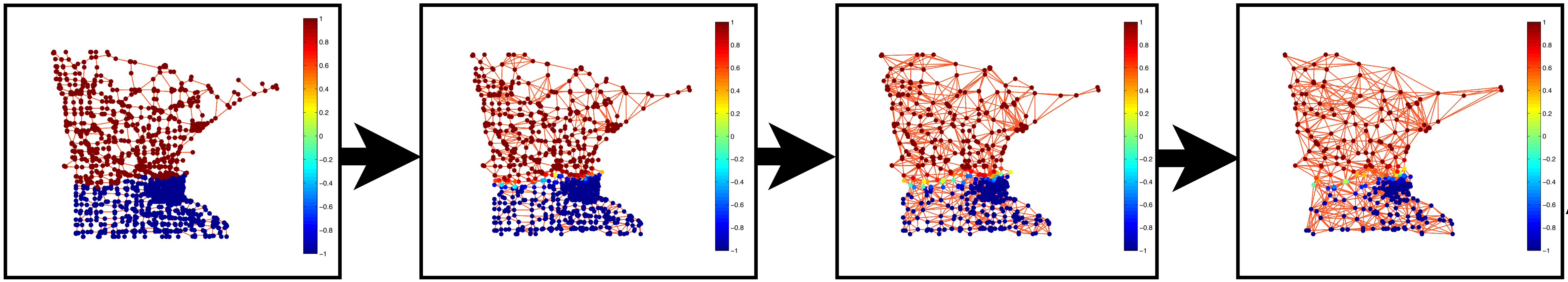
Algorithm: Reduce graph

Apply reduced Laplacian to vertex data

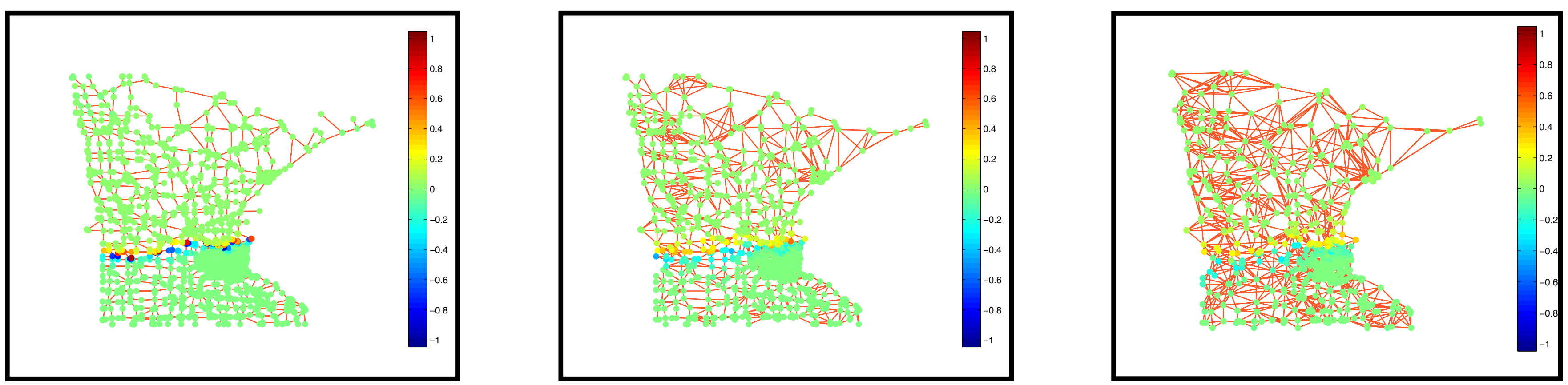
Replace old data with newly calculated coefficients

Filter with Green's kernel

Example



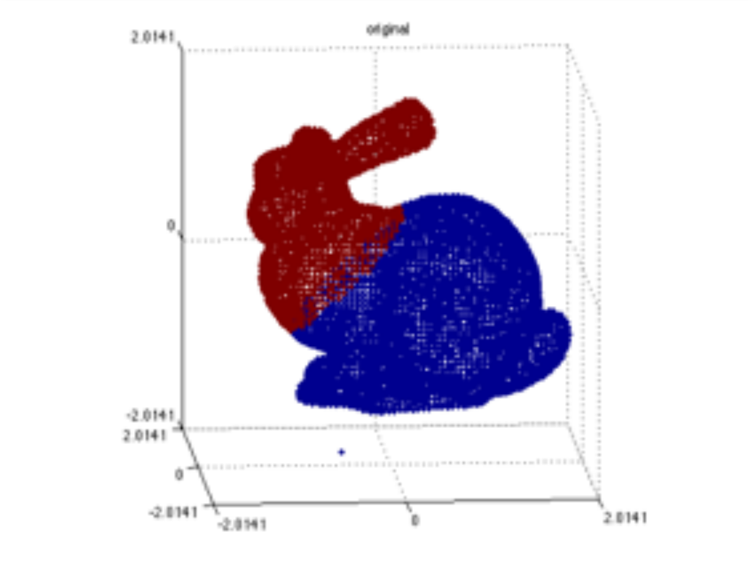
Coarse Approximations

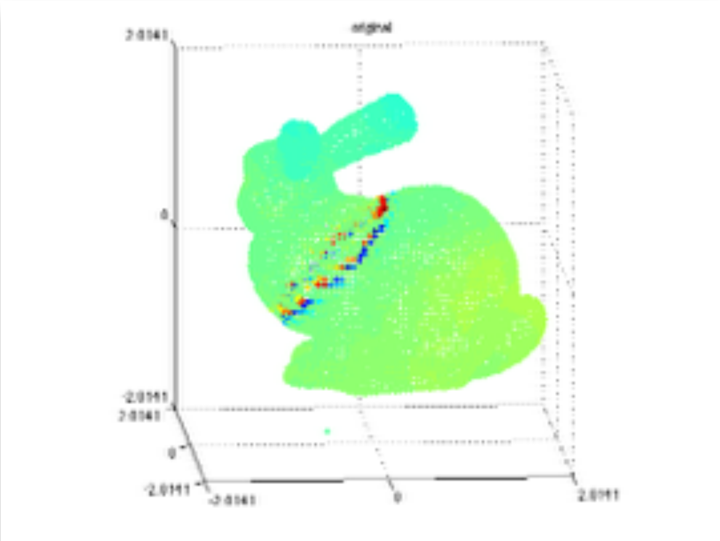


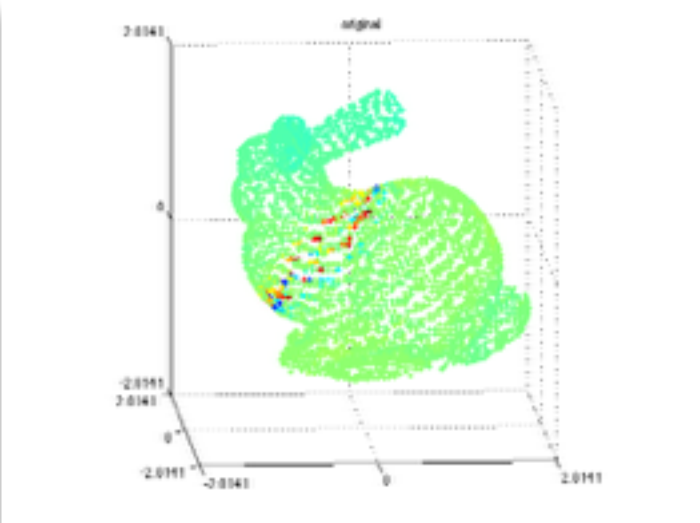
Prediction Errors

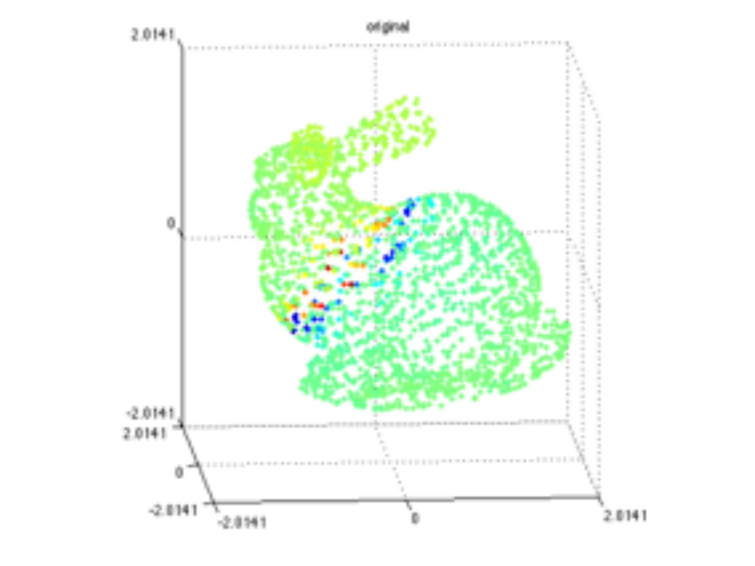
2642 1334 669 337

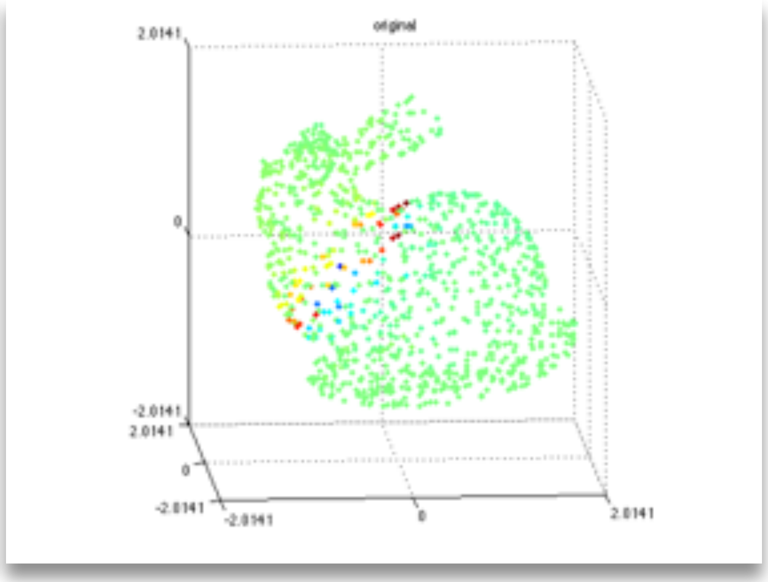
Exact Representation

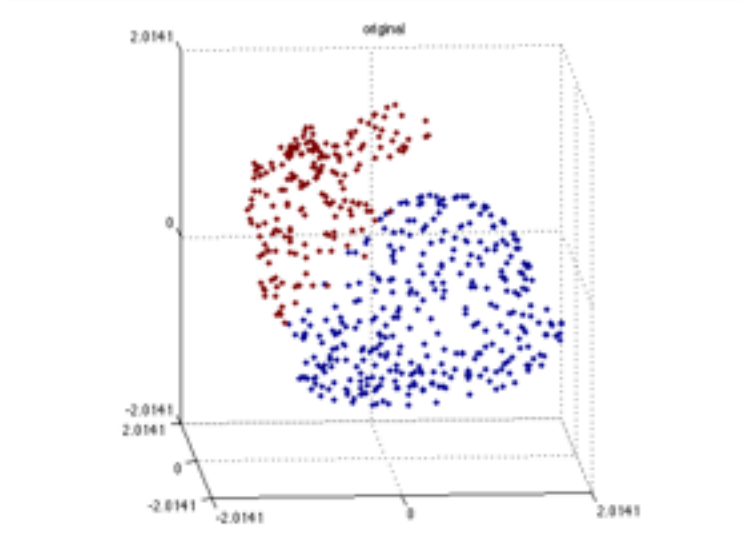

original


original

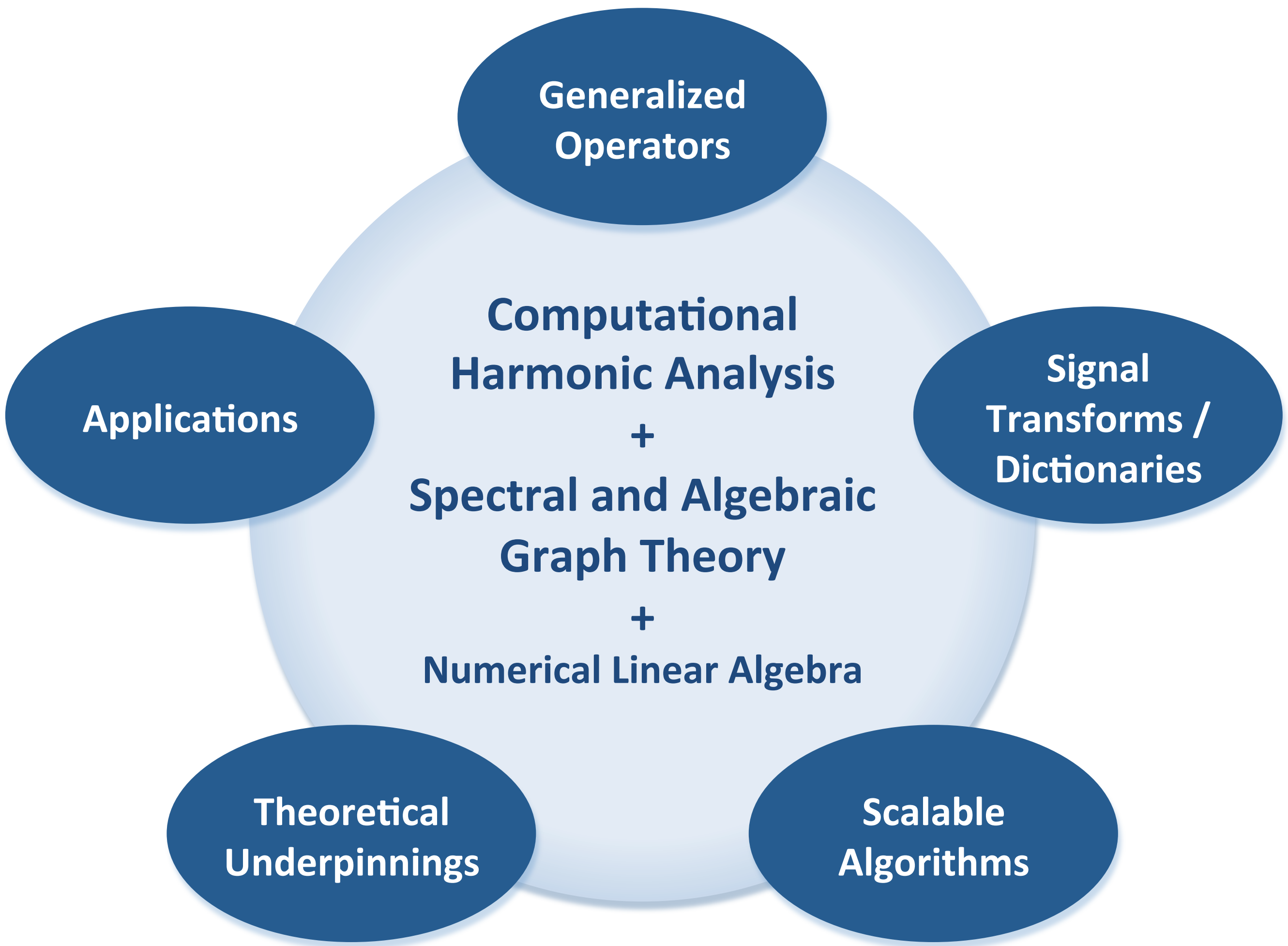

original


original

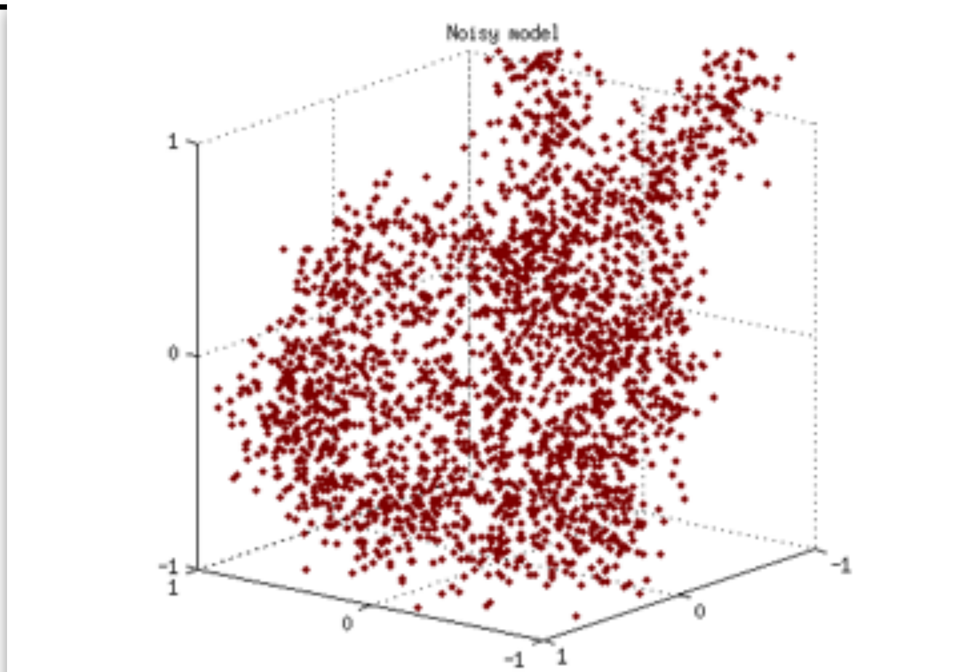

original

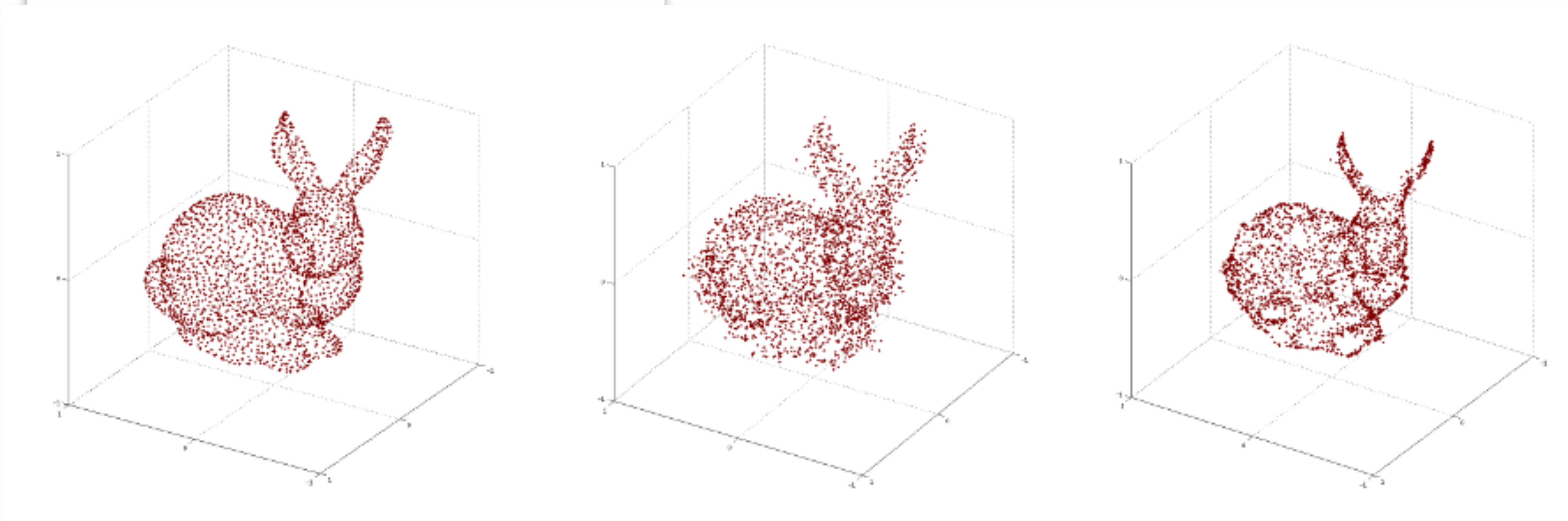

original

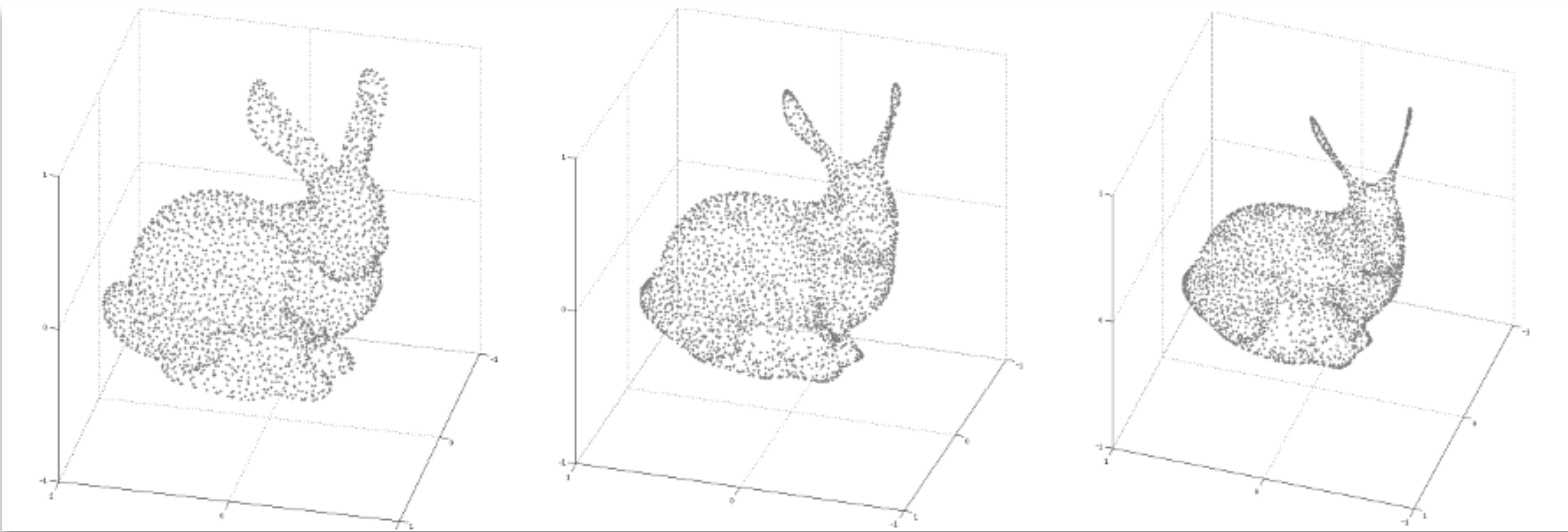
Outlook

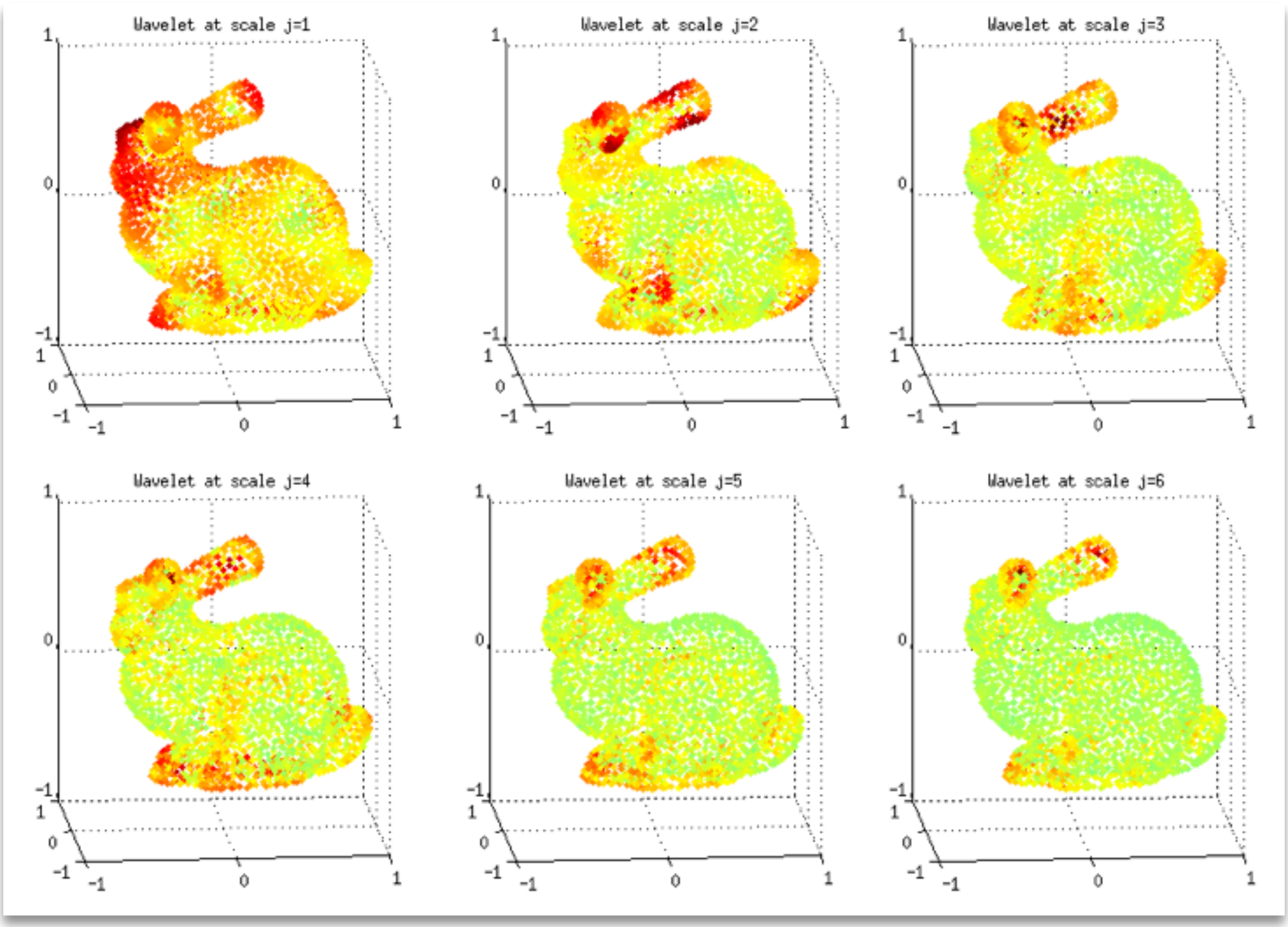


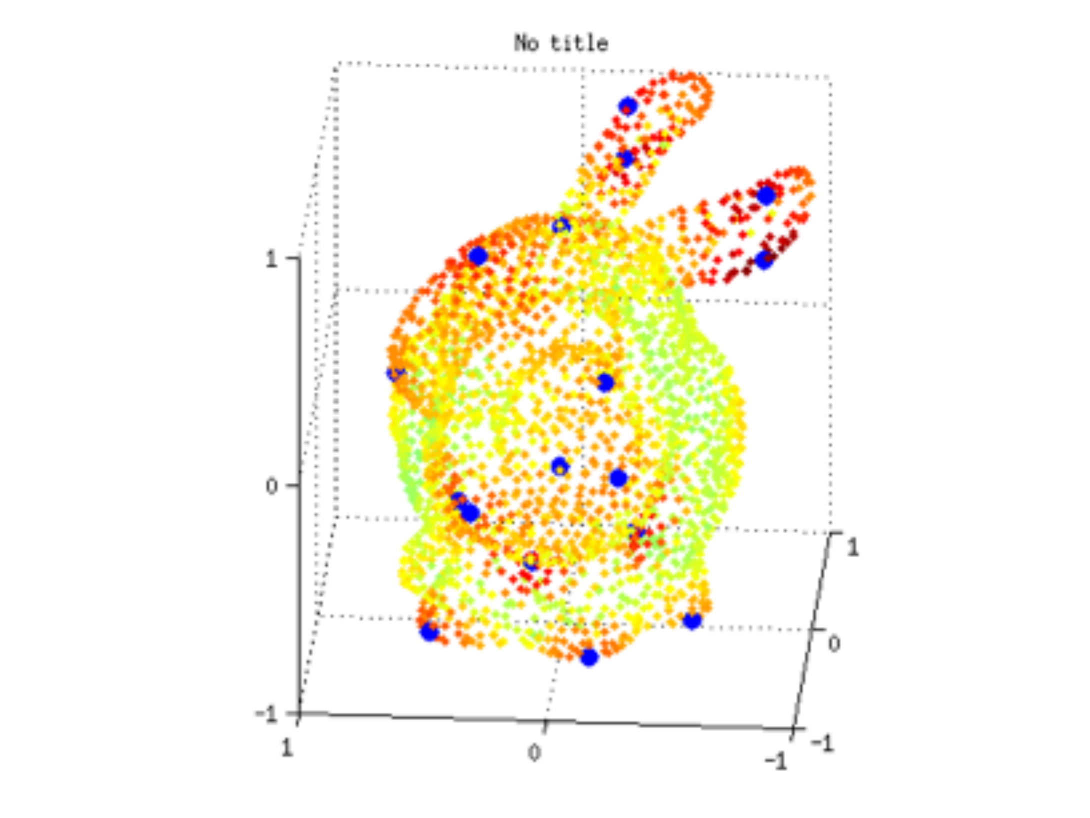
- Application of graph signal processing techniques to real science and engineering problems is in its infancy
- Theoretical connections between classes of graph signals, the underlying graph structure, and sparsity of transform coefficients


Noisy model






Wavelet at scale j=1
Wavelet at scale j=2
Wavelet at scale j=3
Wavelet at scale j=4
Wavelet at scale j=5
Wavelet at scale j=6


No title
1
0
-1
1
0
-1
1
0
-1