Non-conformist Image Processing with the Graph Laplacian Operator

Peyman Milanfar
Google Research

More than 2 billion photos shared on social media per day

That's 23,000 frames/sec

Over 100 million are "selfies"

That's 1,200 frames/sec

Reuters/CBS/TIME/KPCB

Modern Era of Easy Photography

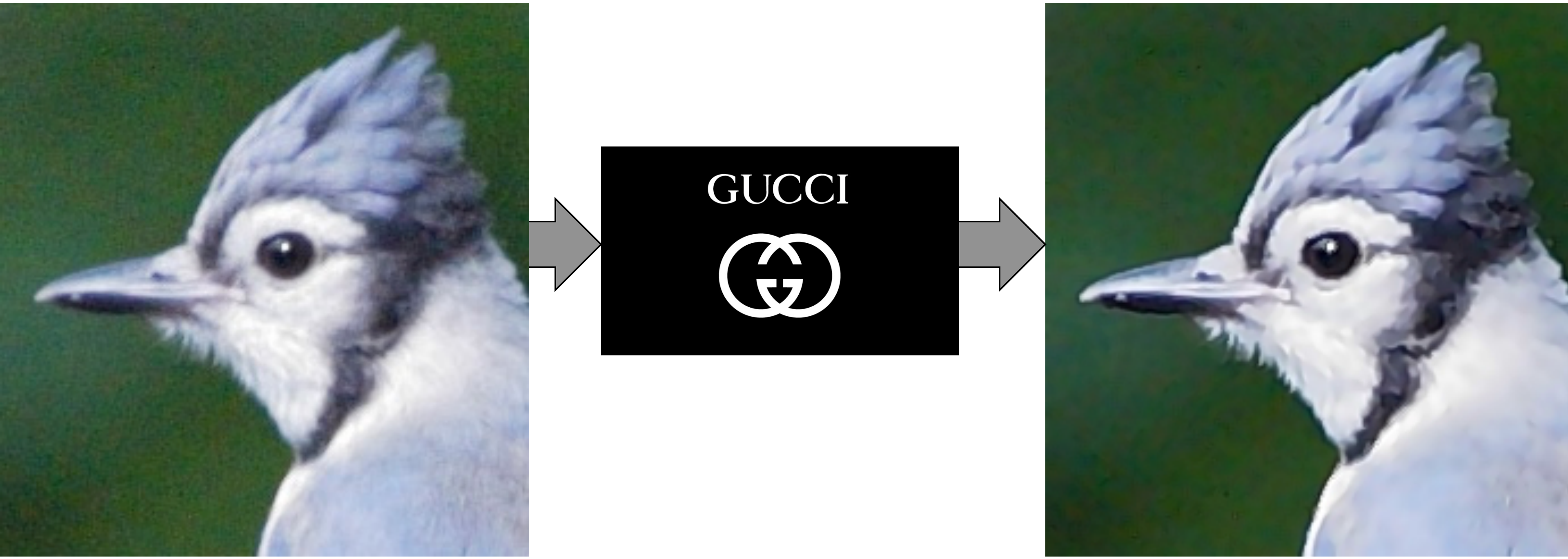
Take it
Change it
Share it

Evolution of (cheap) Photography

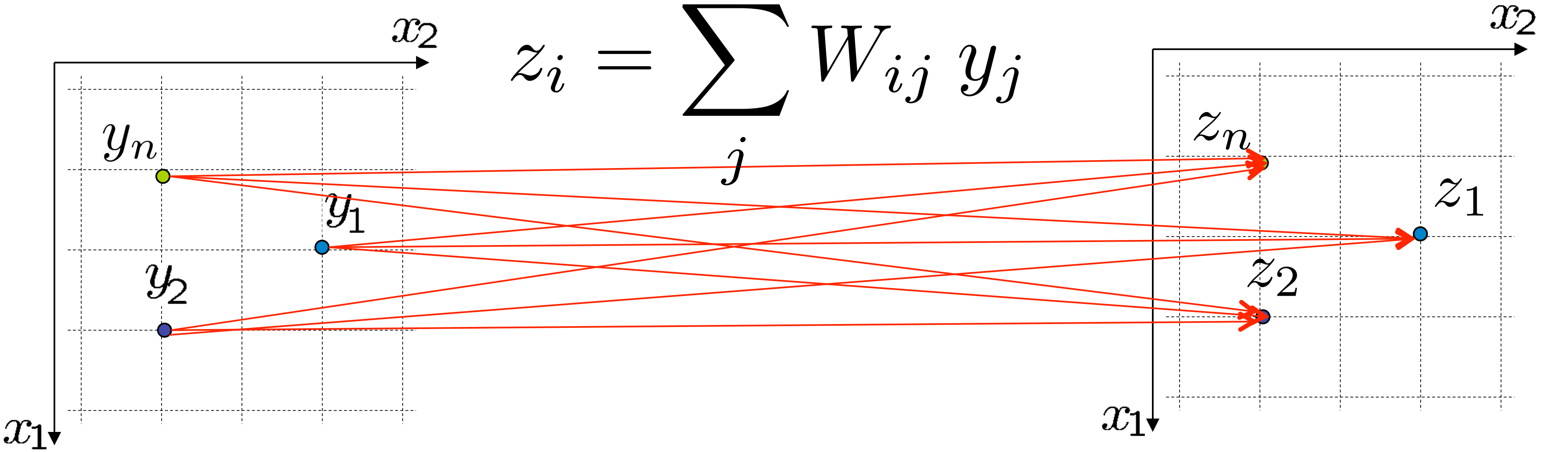
Evolution of (cheap) Photography

Modern Era of Hard Image Processing

Designer Filters



$$z_i = \sum_j W_{ij}\, y_j$$



Global (or Local) Filters

Each output pixel → $$z_i = \sum_j W_{ij}\, y_j$$ ← All input pixels

$$\mathbf{w}_i^T = [W_{1j},\, W_{2j},\, \cdots,\, W_{nj}]$$

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_1^T \\ \mathbf{w}_2^T \\ \vdots \\ \mathbf{w}_n^T \end{bmatrix}$$

rows

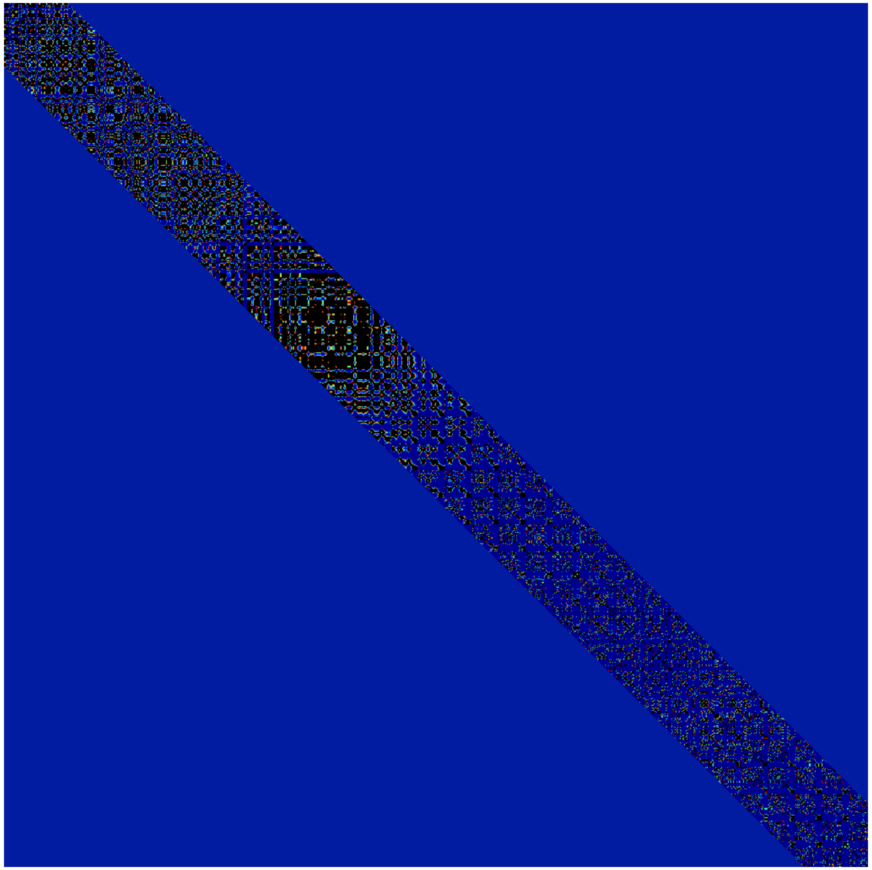
$$\mathbf{z} = \mathbf{W}\mathbf{y}$$

Data-dependent matrix

Image scanned into a vector

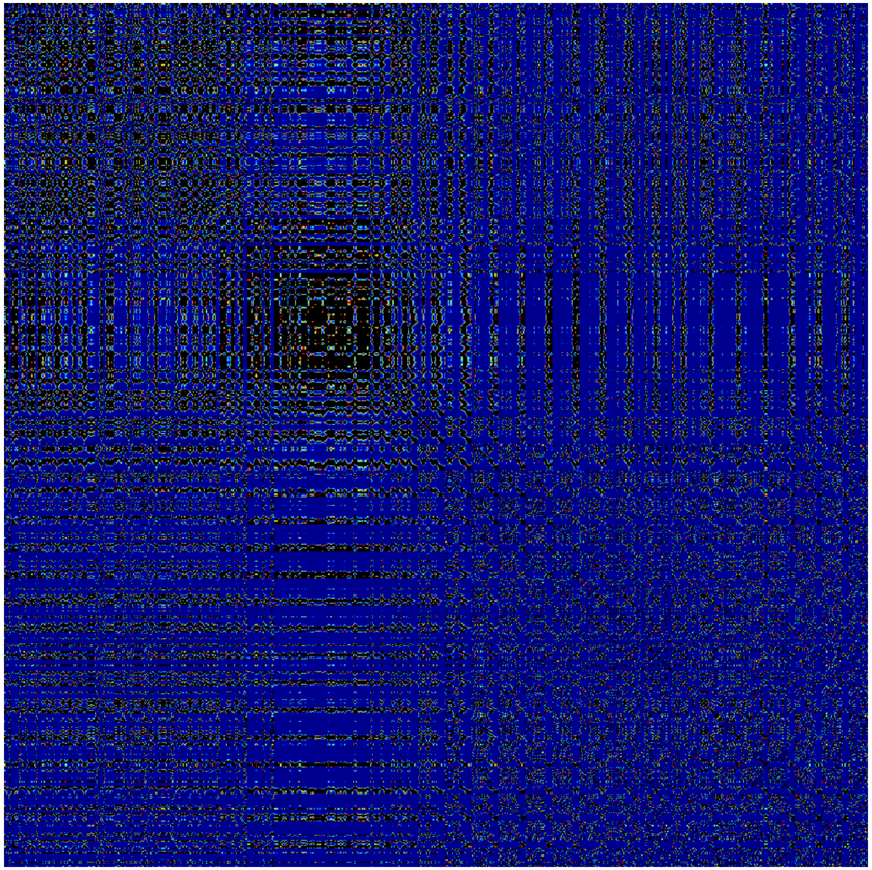
Matrix W

Local Filters



Sparse, but high-rank

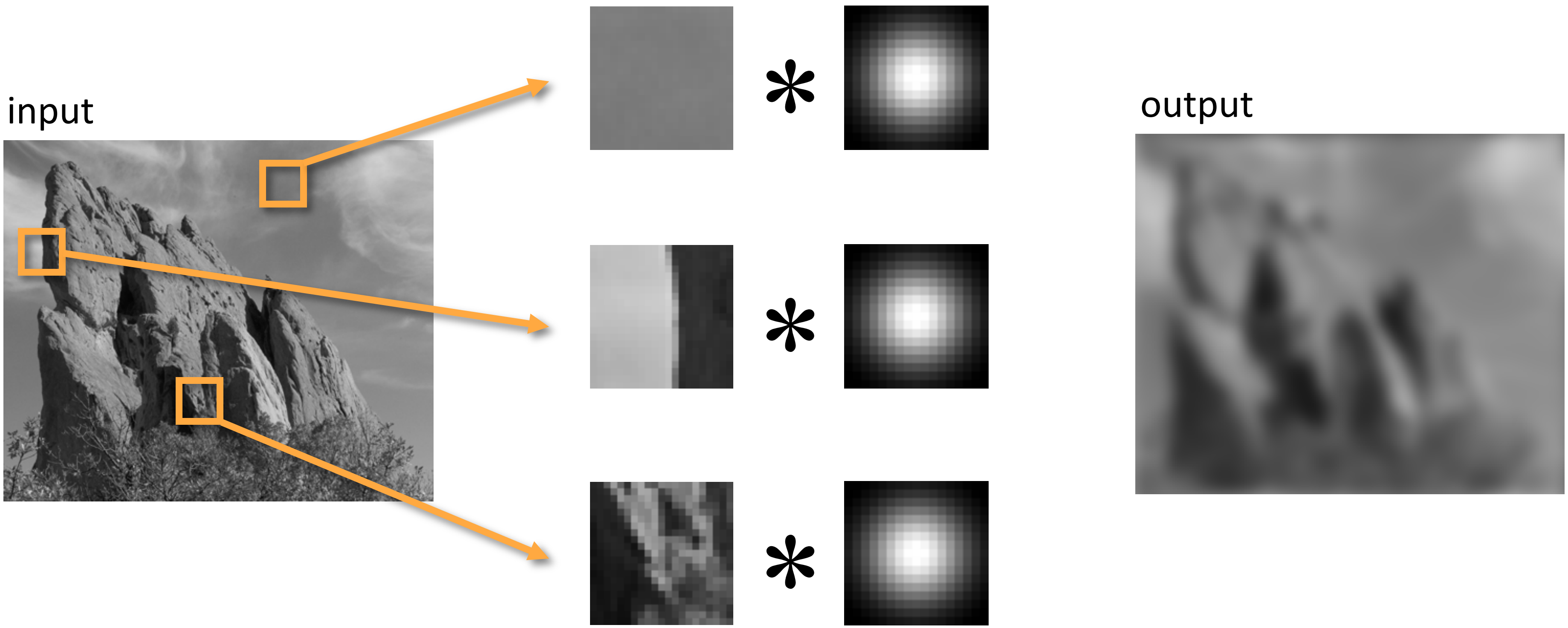
Global Filters



Dense but low-rank

Same weights everywhere

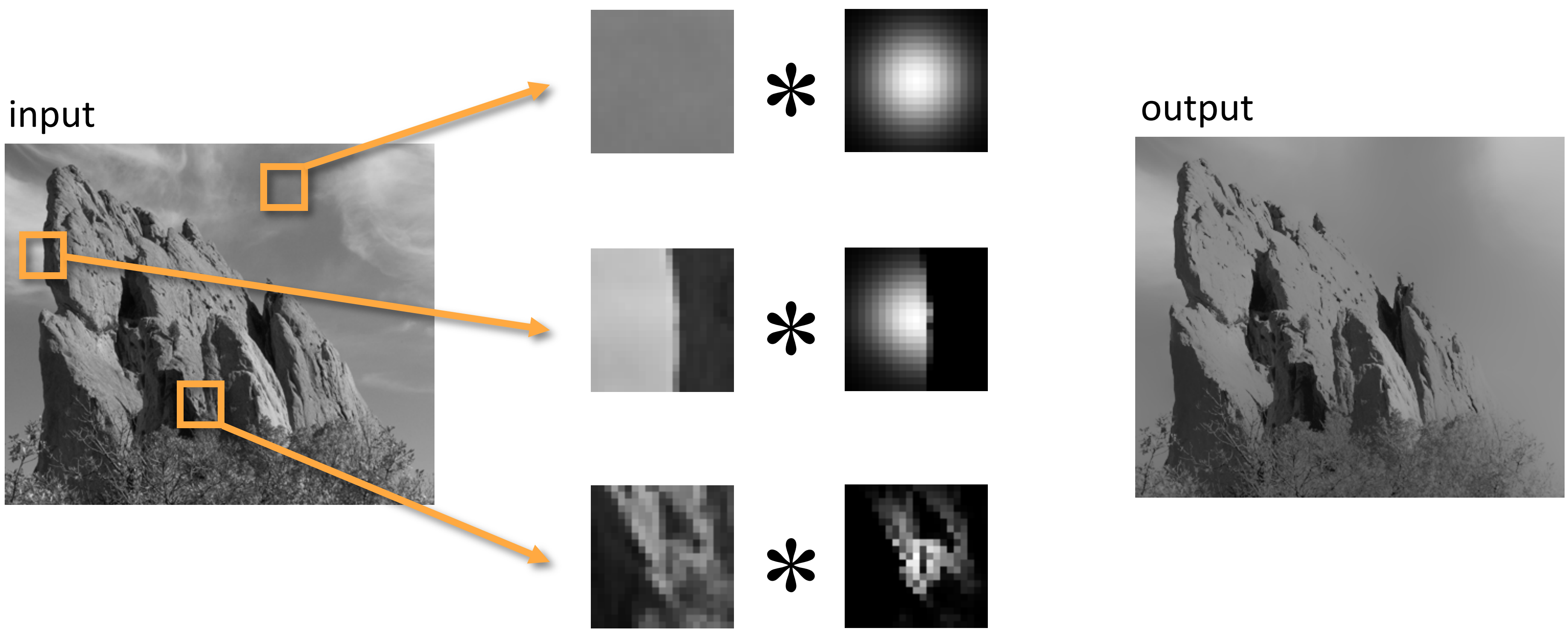
$$z_i = \sum_j W_{ij} \, y_j$$



Same Gaussian kernel everywhere

Data-adaptive weights

$$z_i = \sum_j W_{ij} \, y_j$$



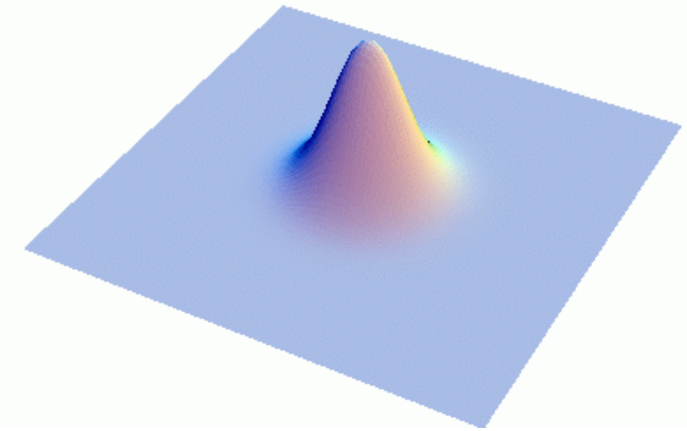
The kernel shape depends on the image content. **But how?**

Image by K. Crane

Laplacian Operators

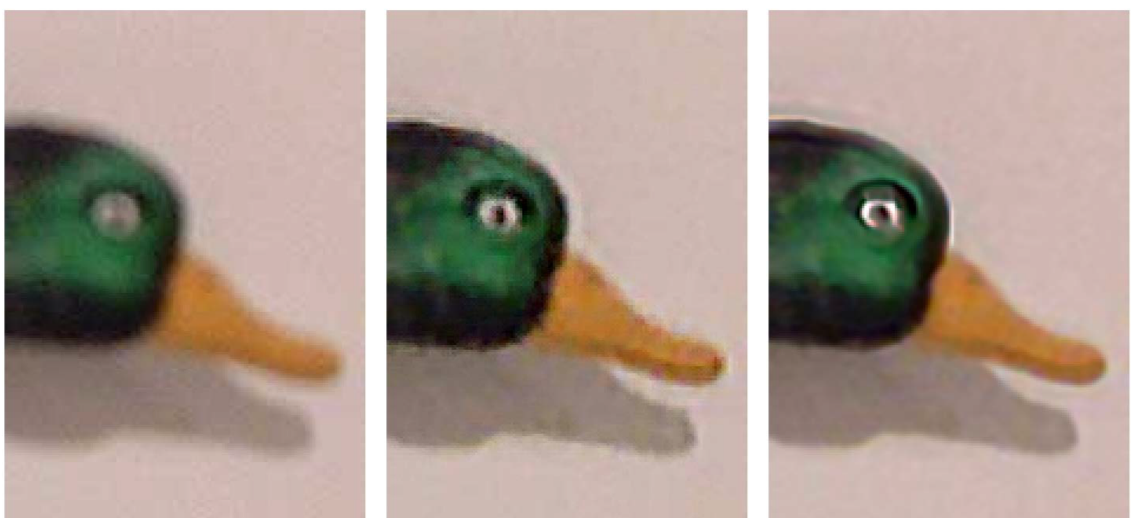
The Laplacian $-\Delta \mathbf{z}(x_1, x_2) = \frac{\partial^2 \mathbf{z}}{\partial x_1^2} + \frac{\partial^2 \mathbf{z}}{\partial x_2^2}$

- It is all over physics
  - Heat equation
  - Wave equation
  - Schrodinger's eqn.
  - Maxwell's eqns
  - Fluid flow
  - .....



The Laplacian in Imaging

- It is all over image processing
  - (Anisotropic) Diffusion
  - Curvature Flow
  - Adaptive Sharpening
  - Deblurring
  - ……



Laplacian, the non-conformist

$$-\Delta \mathbf{z}(x_1, x_2) = \frac{\partial^2 \mathbf{z}}{\partial x_1^2} + \frac{\partial^2 \mathbf{z}}{\partial x_2^2}$$

$$-\Delta \mathbf{z}(x) = \frac{\partial^2 \mathbf{z}}{\partial x^2}$$

$$\approx \frac{\mathbf{z}(x+\epsilon) - 2\mathbf{z}(x) + \mathbf{z}(x-\epsilon)}{\epsilon^2}$$

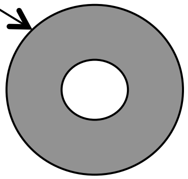
$$= \frac{2}{\epsilon^2}\left[\boxed{\frac{\mathbf{z}(x+\epsilon) + \mathbf{z}(x-\epsilon)}{2}} - \mathbf{z}(x)\right]$$

Average of z around x

The Laplacian in $\mathcal{R}^d$

$$-\Delta \mathbf{z}(x) = \frac{\partial^2 \mathbf{z}}{\partial x_1^2} + \frac{\partial^2 \mathbf{z}}{\partial x_2^2} + \cdots + \frac{\partial^2 \mathbf{z}}{\partial x_d^2}$$

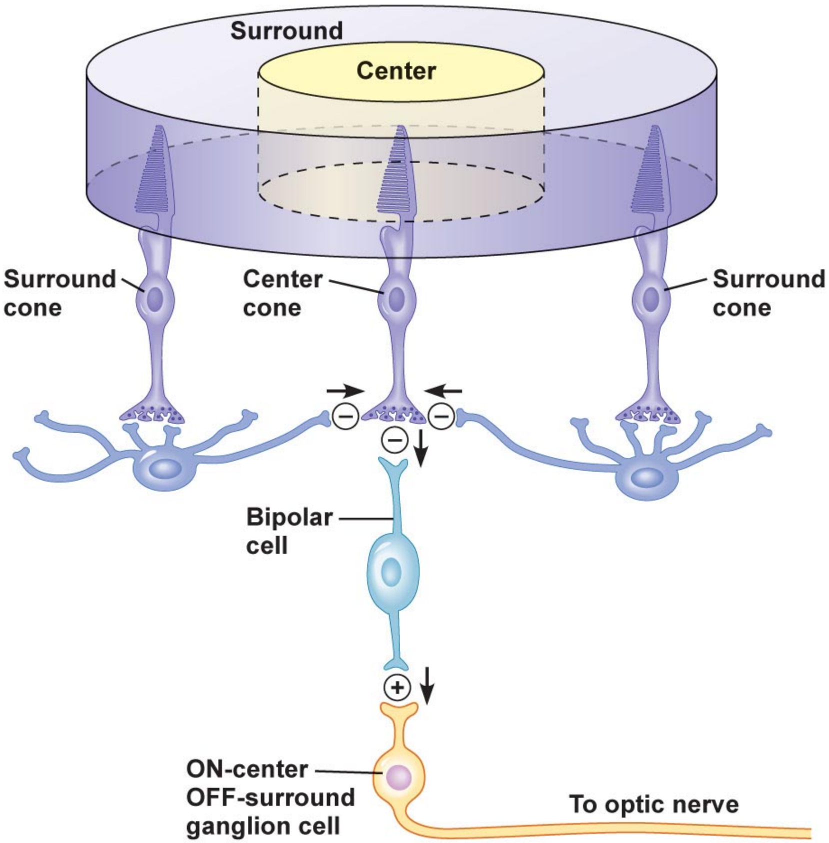
$$-\Delta \mathbf{z}(x) = \lim_{\epsilon \to 0} \frac{2d}{c(\epsilon)} \left(\underset{\mathcal{N}(\epsilon)}{\text{Average}}\{\mathbf{z}\} - \mathbf{z}(x) \right)$$



Laplacian Operator ⟷ Center – Surround average

The Laplacian – In mammalian vision

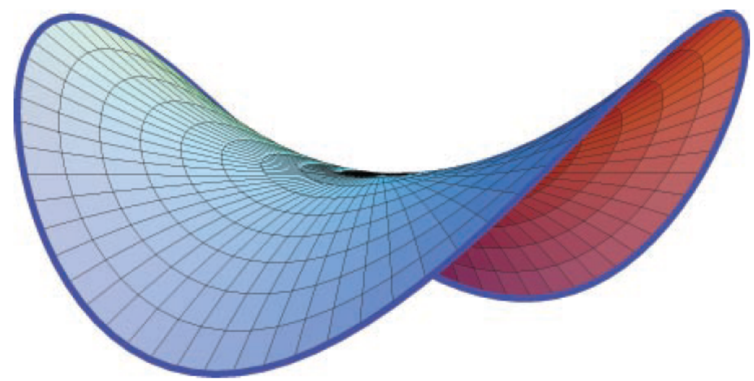
- Center-surround Antagonistic Receptive Fields
  - But these cells have a *nonlinear* response to their input)



The Laplacian -- Properties

- Detects local "*non-conformity*"

- $\Delta \mathbf{z}(x) = 0$ implies *smoothness*
  - Harmonic/analytic functions
  - Mean-value property



- Minimizer of Gradient (Dirichlet) Energy

$$\min_{\mathbf{z}} \int \|\nabla \mathbf{z}(x)\|^2 dx \longleftrightarrow \Delta \mathbf{z}(x) = 0$$

The Laplacian and Diffusion

$$\min_{\mathbf{z}} \int \|\nabla \mathbf{z}\|^2 dx$$

$$\mathbf{z}(x) \longrightarrow \mathbf{z}(x, t)$$

Dirichlet Energy

(Functional) Gradient Descent

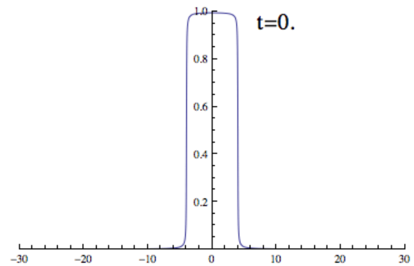
$$\frac{\partial \mathbf{z}(x,t)}{\partial t} = \Delta \mathbf{z}(x,t)$$

Diffusion Eqn

$$\mathbf{z}(x, 0) = \mathbf{z}_0$$

The Laplacian and Smoothing

$$\frac{\partial \mathbf{z}(x,t)}{\partial t} = \Delta \mathbf{z}(x,t) \quad \Rightarrow \quad \mathbf{z}(x,t) = \mathbf{z}_0 * \mathcal{H}(x,t)$$



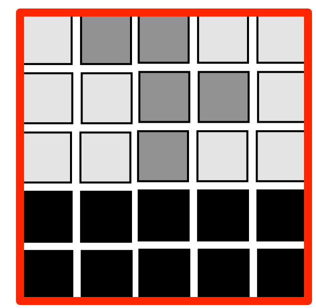
$$\frac{\mathbf{z}(x,t+\delta t) - \mathbf{z}(x,t)}{\delta t} \approx \Delta \mathbf{z}(x,t)$$

$$\mathbf{z}_0 * \frac{\mathcal{H}(x,t+\delta t) - \mathcal{H}(x,t)}{\delta t} \approx \Delta \mathbf{z}(x,t)$$

Laplacian Operator ⟷ Diff. of Smoothing Kernels

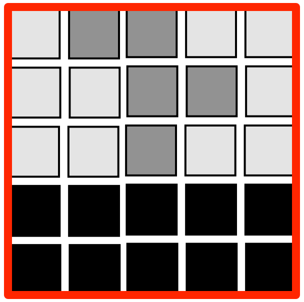
What about practical use?

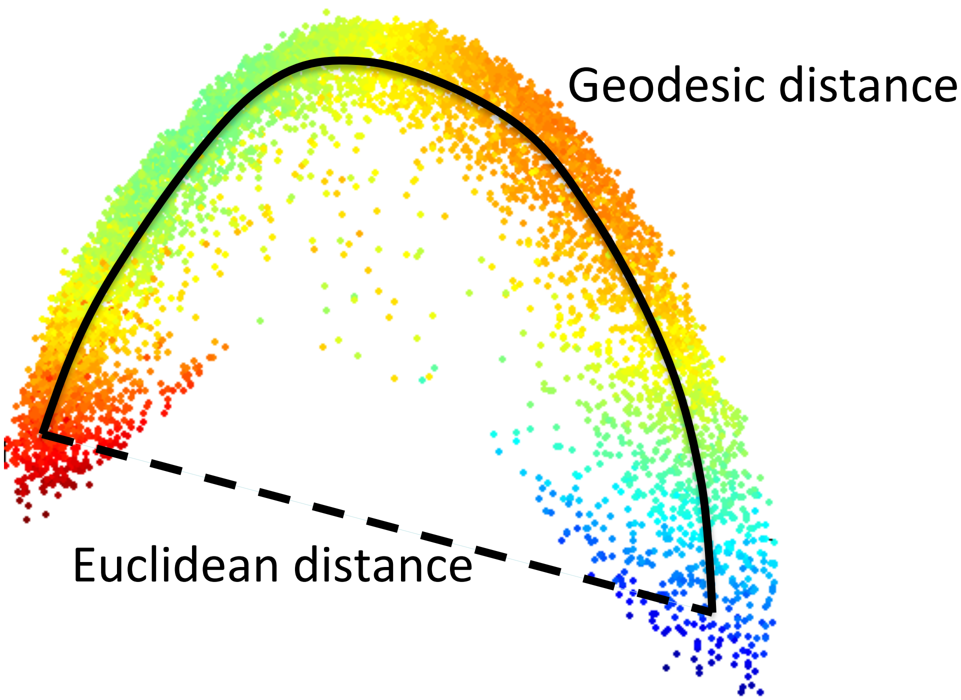
- Data is generally discrete, and non-parametric
  - pixels and patches, instead of functions

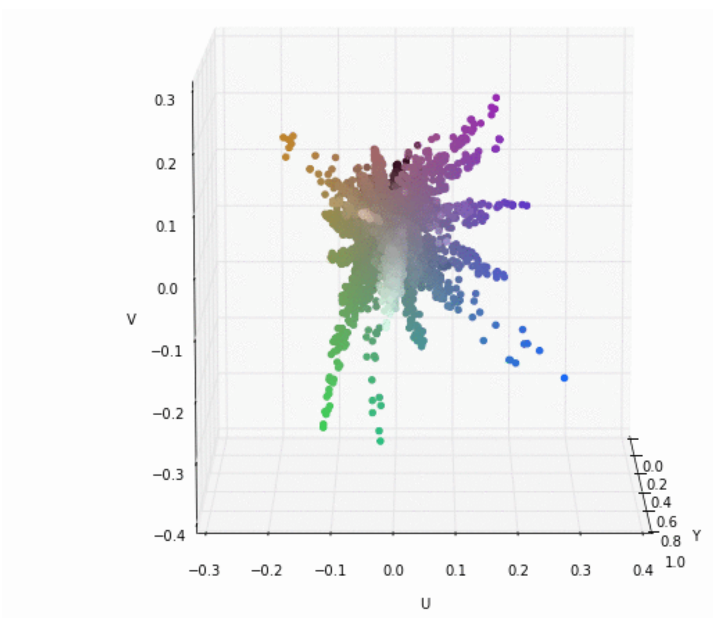


What about practical use?

- Data is generally discrete
  - Pixels and patches, instead of functions
- Image (patches) live on a manifold
- Useful distances are hard to measure

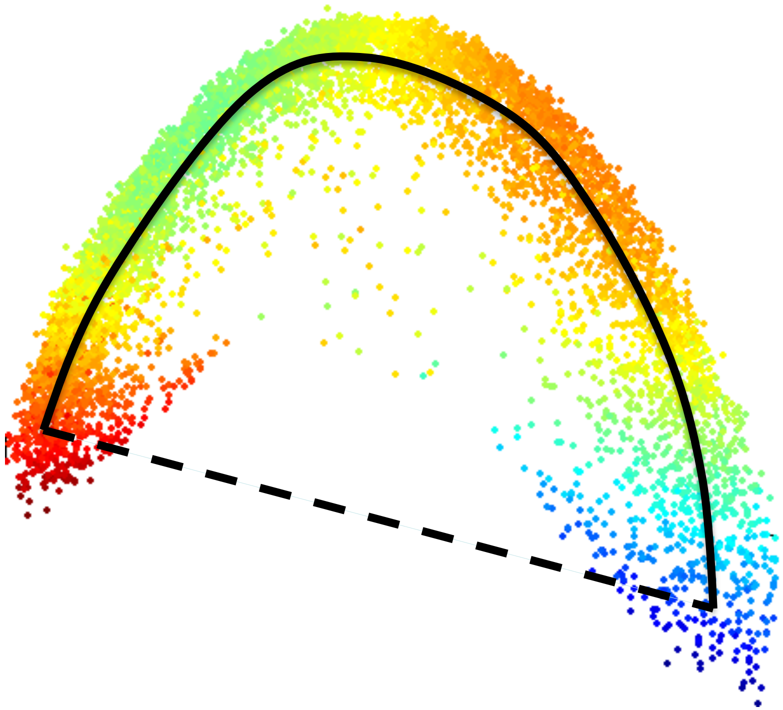






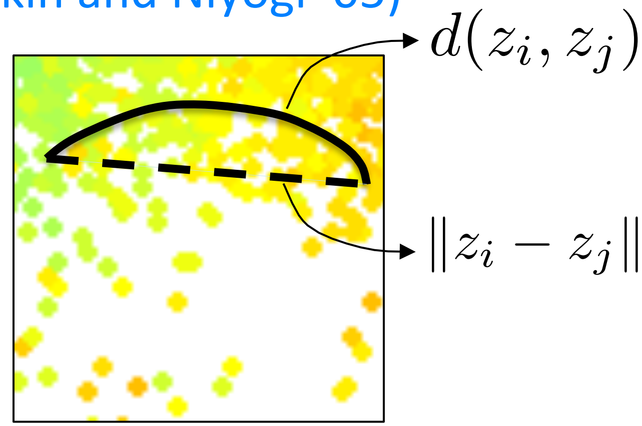
Discrete Laplacian

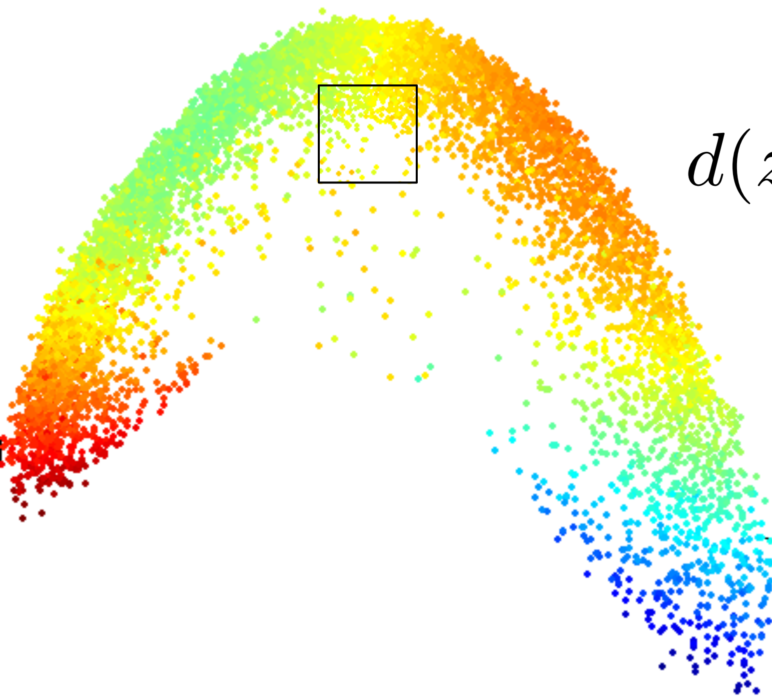
- Just a cloud of point $z_i = \mathbf{z}(x_i)$
  - No explicit manifold
  - No specified metric
  - No geodesic paths



Discrete Laplacian

- Let’s look closely (Belkin and Niyogi ‘05)





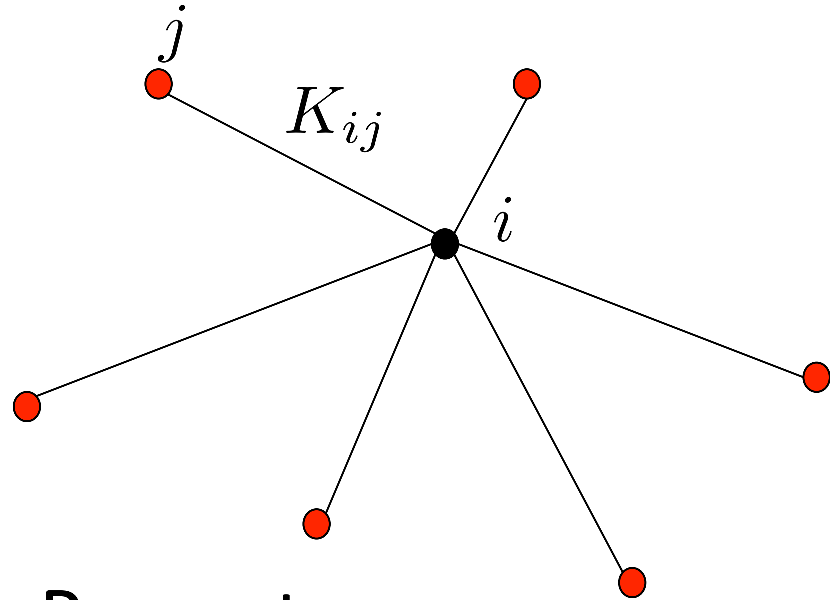
$$d(z_i, z_j) = \|z_i - z_j\| + \mathcal{O}(\|z_i - z_j\|^3)$$

For nearby points, chordal distance approximates the geodesic distance.

Discrete (Graph) Laplacian

$$\mathcal{L}(z_i) = \sum_j K_{ij}(z_i - z_j)$$

$$= z_i \sum_j K_{ij} - \sum_j K_{ij}\, z_j$$

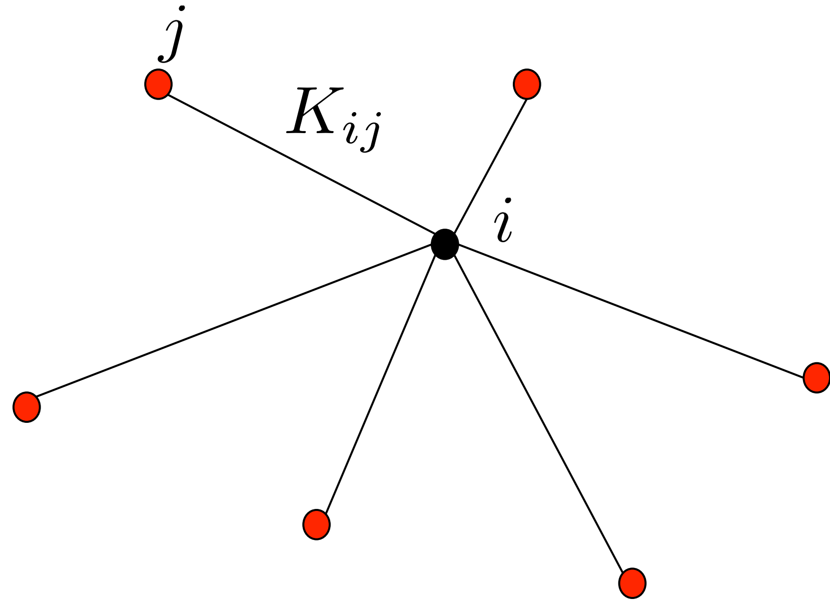


- Center-surround/mean-value Property
- Measure Smoothness
- Notion of Diffusion
- With enough samples, converges to the continuous definition of Laplacian (Lafon '04, Belkin, et. al '05, Hein et al. '05, Singer '06, ….)

Graph Laplacian and Smoothness

$$\mathcal{L}(z_i) = \sum_j K_{ij}(z_i - z_j)$$

$$= z_i \sum_j K_{ij} - \sum_j K_{ij}\, z_j$$



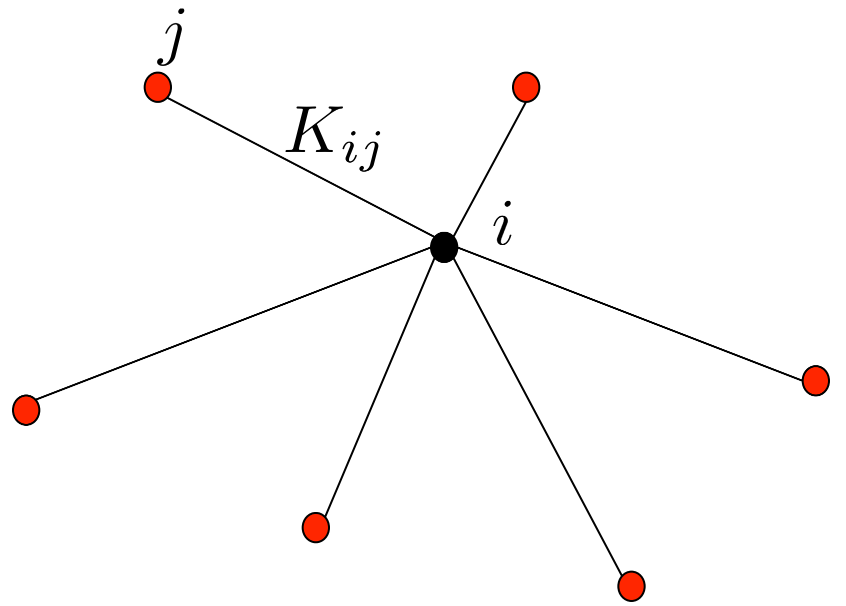
$$\mathcal{L}(z_i) = 0 \quad \Longrightarrow \quad z_i = \frac{\sum_j K_{ij}\, z_j}{\sum_j K_{ij}}$$

"Filter as prior"

Kernel Weighted Graph

Kernel (Affinity) Matrix

$$\mathbf{K} = \{K_{ij}\}$$

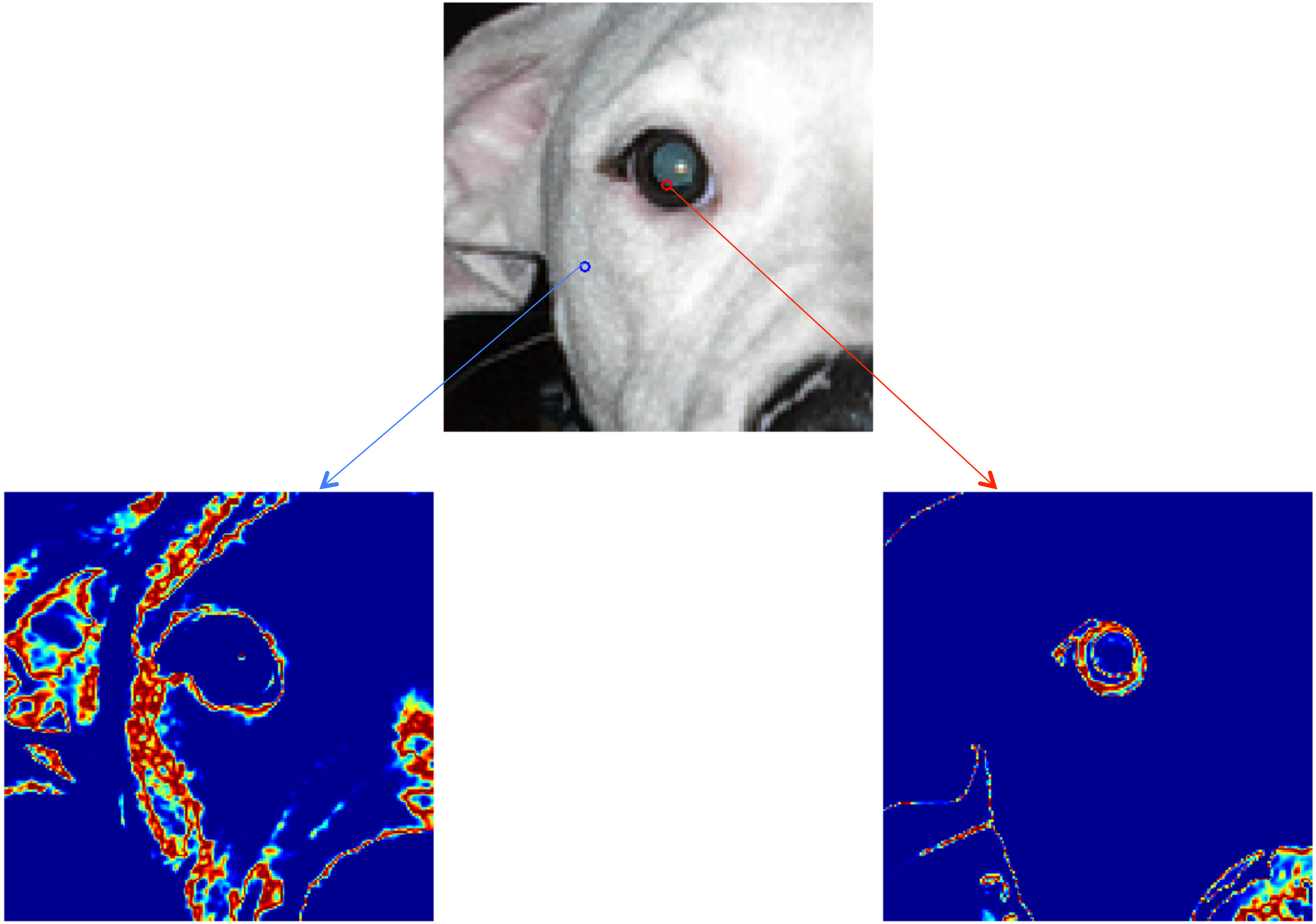


$$K(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / h_x^2)$$ Spatial Gaussian Kernel

$$K(z_i, z_j) = \exp(-\|z_i - z_j\|^2 / h_z^2)$$ Photometric Gaussian Kernel

$$K_{ij} = K(z_i, z_j) K(x_i, x_j)$$ Bilateral Kernel

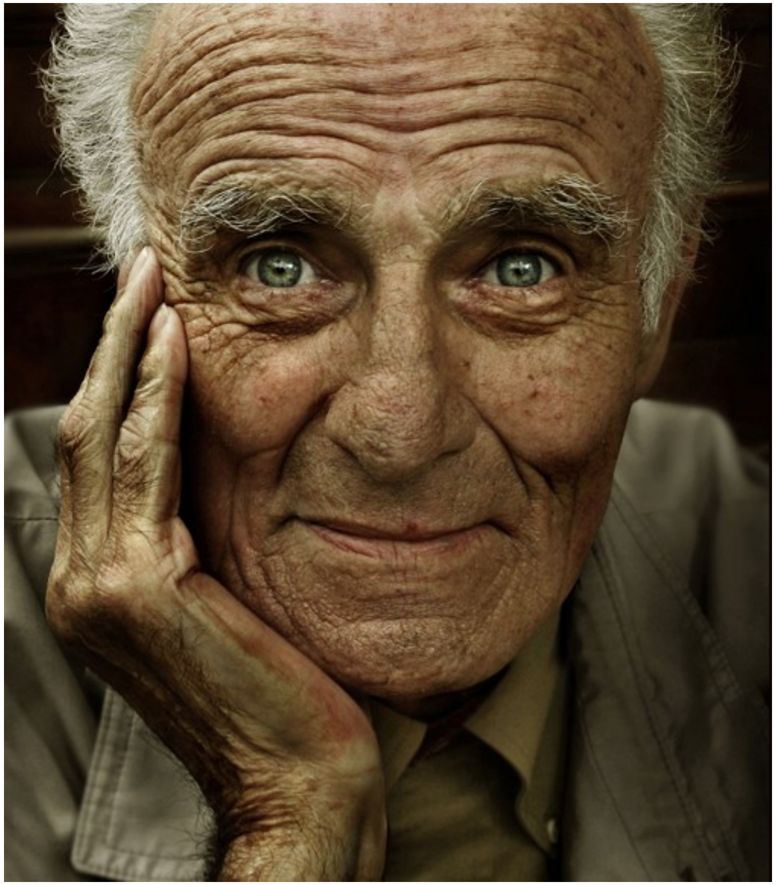
Non-Local Means Affinities



“Degree” of Pixel ~ Number of similar pixels

$$d_i = \sum K_{ij}$$

Image



Laplacians Give Birth to Filters (I)

$$\mathcal{E} = \sum_j K_{ij}(z_i - z_j)^2$$ Dirichlet Energy

Gradient Descent $\mathbf{z}_0 = \mathbf{y}$

$$\mathbf{z}_{k+1} = (\mathbf{I} - \mathbf{L})\,\mathbf{z}_k = \mathbf{W}\;\mathbf{z}_k$$ Diffusion

$$\widehat{\mathbf{z}} = \mathbf{W}\;\mathbf{y}$$

Or… *Define* Laplacians from Filters

$$\widehat{\mathbf{z}} = \mathbf{W}\ \mathbf{y}$$

$$\mathbf{L}\,\mathbf{z} = \mathbf{z} - \mathbf{W}\,\mathbf{z}$$

$$\mathbf{L} = \mathbf{I} - \mathbf{W}$$

Center-surround

$$\mathbf{z}^T \mathbf{L}\,\mathbf{z}$$

Dirichlet Energy

Laplacians Give Birth to Filters (II)

$$(\mathbf{y} - \mathbf{z})^T (\mathbf{I} - \mathbf{L})(\mathbf{y} - \mathbf{z})$$

Dirichlet Energy
On “residuals”

Gradient Descent

$$\mathbf{z}_0 = \mathbf{W}\,\mathbf{y}$$

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \mathbf{W}(\mathbf{y} - \mathbf{z}_k)$$

$$\widehat{\mathbf{z}} = (\mathbf{2I} - \mathbf{W})\mathbf{W}\mathbf{y}$$

Reaction-Diffusion (Nordstrom ‘90)
Twicing (Tukey,’77)
(Kaiser and Hamming ’77)
Boosting (Buhlmann et al. ’03)
Bregman iter. (Osher et al. ’05)

Graph Laplacian so far ....

$$d_i = \sum_j K_{ij} \qquad \mathbf{D} = \mathrm{diag}\{d_i\}_1^n$$

Graph Laplacian	Symmetric	DC eigenvector	Spectral Range
$\mathbf{L} = \mathbf{D} - \mathbf{K}$	Yes	Yes	[0, n]

$$\mathcal{L}(z_i) = \sum_j K_{ij}(z_i - z_j) = z_i \sum_j K_{ij} - \sum_j K_{ij}\, z_j$$

$$= z_i\, d_i - \sum_j K_{ij}\, z_j$$

Un-normalized Laplacian

Other definitions

$$d_i = \sum_j K_{ij} \qquad \mathbf{D} = \mathrm{diag}\{d_i\}_1^n$$

Graph Laplacian	Symmetric	DC eigenvector	Spectral Range
$D - K$	Yes	Yes	[0, n]
$I - D^{-1/2} K D^{-1/2}$	Yes	No	[0,2]

Normalized Laplacian (Chung '97)

Other definitions

$$d_i = \sum_j K_{ij} \qquad \mathbf{D} = \mathrm{diag}\{d_i\}_1^n$$

Graph Laplacian	Symmetric	DC eigenvector	Spectral Range
$D - K$	Yes	Yes	[0, n]
$I - D^{-1/2} K D^{-1/2}$	Yes	No	[0,2]
$I - D^{-1} K$	No	Yes	[0,1]

Random Walk Laplacian

Other definitions

$$d_i = \sum_j K_{ij} \qquad \mathbf{D} = \mathrm{diag}\{d_i\}_1^n$$

Graph Laplacian	Symmetric	DC eigenvector	Spectral Range
$D - K$	Yes	Yes	[0, n]
$I - D^{-1/2} K D^{-1/2}$	Yes	No	[0,2]
$I - D^{-1} K$	No	Yes	[0,1]
$\mathbf{I} - \mathbf{C}^{-1/2} \mathbf{K} \mathbf{C}^{-1/2}$	**Yes**	**Yes**	[0,1]

"Sinkhorn" Laplacian (M.,'13)

Stick to one definition of Laplacian

- Un-normalized $\mathbf{L}_u = \mathbf{D} - \mathbf{K}$

$$\mathcal{L}_u(z_i) = z_i \, d_i - \sum_j K_{ij} \, z_j$$

- Random Walk $\mathbf{L}_r = \mathbf{I} - \mathbf{D}^{-1}\mathbf{K}$

$$\mathcal{L}_r(z_i) = z_i - \frac{1}{d_i} \sum_j K_{ij} z_j$$

Stick to one definition of Laplacian

- *Re*-normalized $\mathbf{L}_u = \alpha\left(\mathbf{D} - \mathbf{K}\right)$

$\alpha = \mathcal{O}(n^{-1})$ $\mathcal{L}_u(z_i) = \alpha\, z_i\, d_i - \alpha \sum_j K_{ij}\, z_j$

- Random Walk $\mathbf{L}_r = \mathbf{I} - \mathbf{D}^{-1}\mathbf{K}$

$$\mathcal{L}_r(z_i) = z_i - \frac{1}{d_i}\sum_j K_{ij} z_j$$

Corresponding Filters

- Re-normalized $\mathbf{W}_u = \mathbf{I} - \alpha(\mathbf{D} - \mathbf{K})$

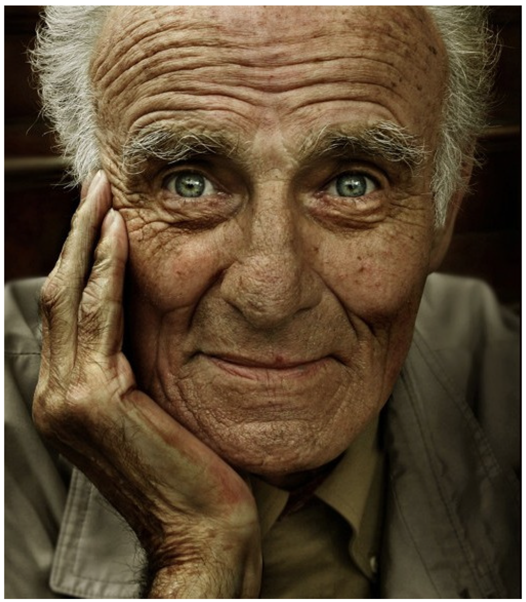
$$\widehat{z}_i = (1 - \alpha d_i)\, y_i + \alpha \sum_j K_{ij}\, y_j$$

- Random Walk $\mathbf{W}_r = \mathbf{D}^{-1}\mathbf{K}$

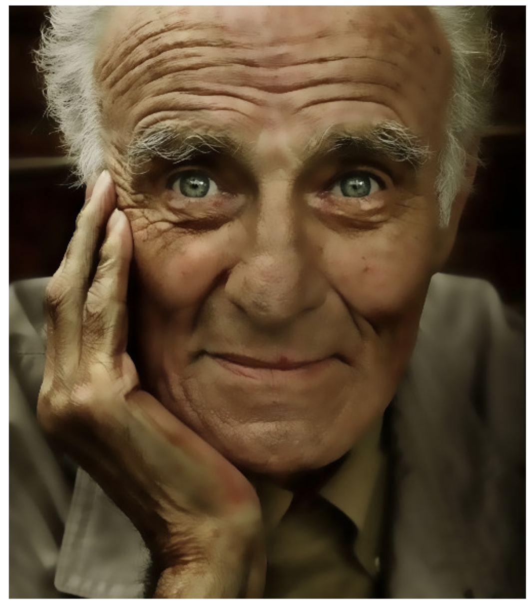
$$\widehat{z}_i = \frac{1}{d_i} \sum_j K_{ij}\, y_j$$

Example: Bilateral Kernel

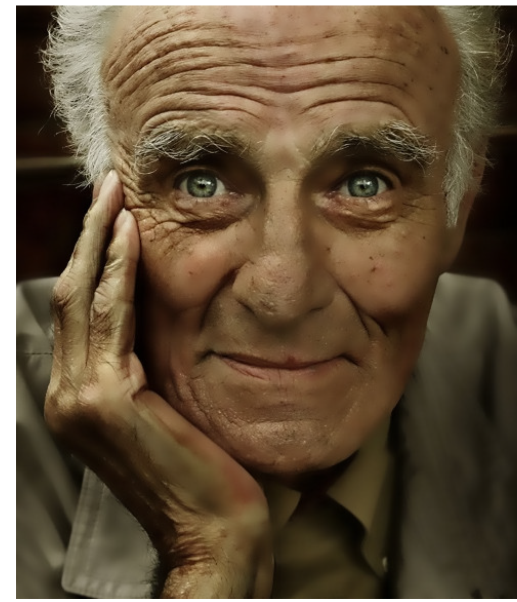
Input Image



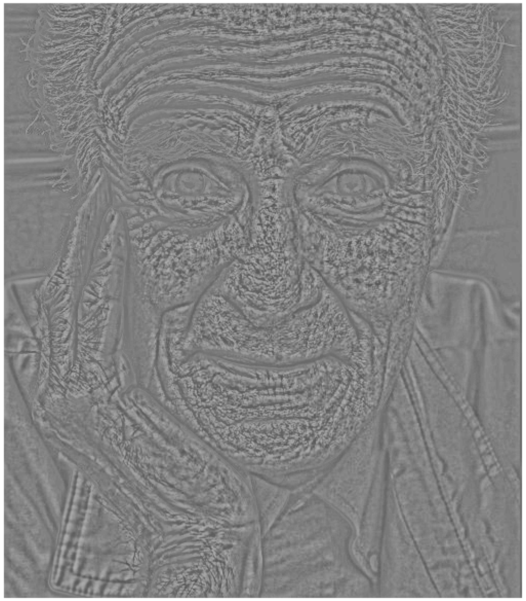
Random Walk Filter



Re-normalized Filter



Residuals



More Generally

$$\underbrace{(\mathbf{y}-\mathbf{A}\mathbf{z})^T(\mathbf{I}+\beta\mathbf{L})(\mathbf{y}-\mathbf{A}\mathbf{z})}_{\text{Data Fidelity}} + \underbrace{\eta\,\mathbf{z}^T\mathbf{L}\mathbf{z}}_{\text{Regularization}}$$

More Generally

$$(\mathbf{y} - \mathbf{A}\mathbf{z})^T(\mathbf{I} + \beta\mathbf{L})(\mathbf{y} - \mathbf{A}\mathbf{z}) + \eta\, \mathbf{z}^T\mathbf{L}\mathbf{z}$$

"Blur"

Regularization

Deblurring – Estimated PSF

$$(\mathbf{y} - \mathbf{A}\mathbf{z})^T(\mathbf{I} + \beta\mathbf{L})(\mathbf{y} - \mathbf{A}\mathbf{z}) + \eta\, \mathbf{z}^T\mathbf{L}\mathbf{z}$$

Adaptive Sharpening as a Special Case

$$(\mathbf{y} - \mathbf{A}\mathbf{z})^T(\mathbf{I} + \beta\mathbf{L})(\mathbf{y} - \mathbf{A}\mathbf{z}) + \eta\, \mathbf{z}^T\mathbf{L}\mathbf{z}$$

- Special Case: $\mathbf{A} = \mathbf{I} \qquad \eta = 0$

$$\widehat{\mathbf{z}} = (\mathbf{I} + \beta\mathbf{L})\,\mathbf{y} \qquad \beta > 0$$

Adaptive Sharpening "nonlinear unsharp mask"
(No knowledge of PSF)

No knowledge of PSF

Adaptive Sharpening

Linear Embedding Using Laplacian

$$\min \frac{\mathbf{z}^T \mathbf{L} \mathbf{z}}{\mathbf{z}^T \mathbf{z}} \quad \Longleftrightarrow \quad \max \frac{\mathbf{z}^T \mathbf{W} \mathbf{z}}{\mathbf{z}^T \mathbf{z}}$$

$$\Downarrow$$

$$\widehat{\mathbf{z}} \approx \sum_i \alpha_i \times \text{ top eigenvectors of } \mathbf{W}$$

$$\widehat{\mathbf{z}} \approx \sum_i \alpha_i \times \text{ bottom eigenvectors of } \mathbf{L}$$

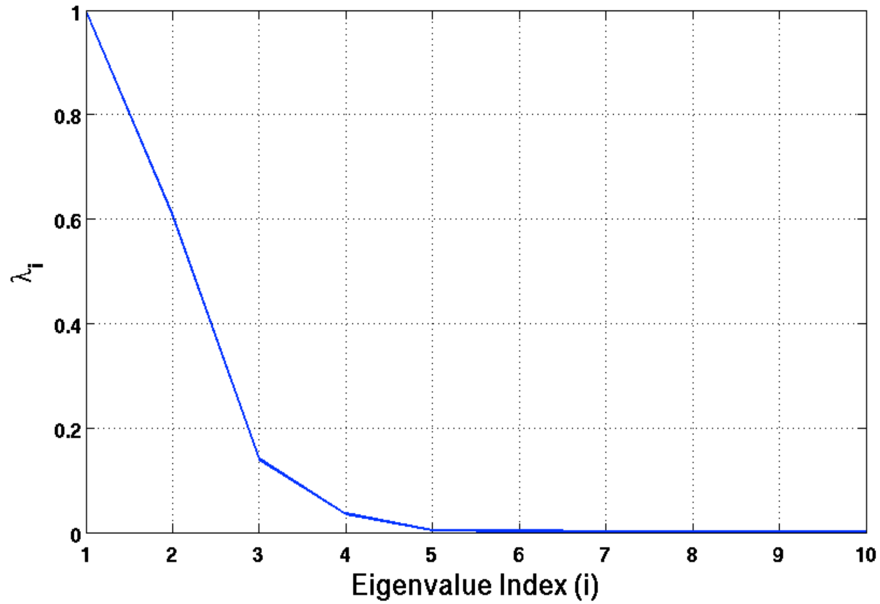
The (Orthonormal) Eigenvectors

Why? How many eigenvectors?

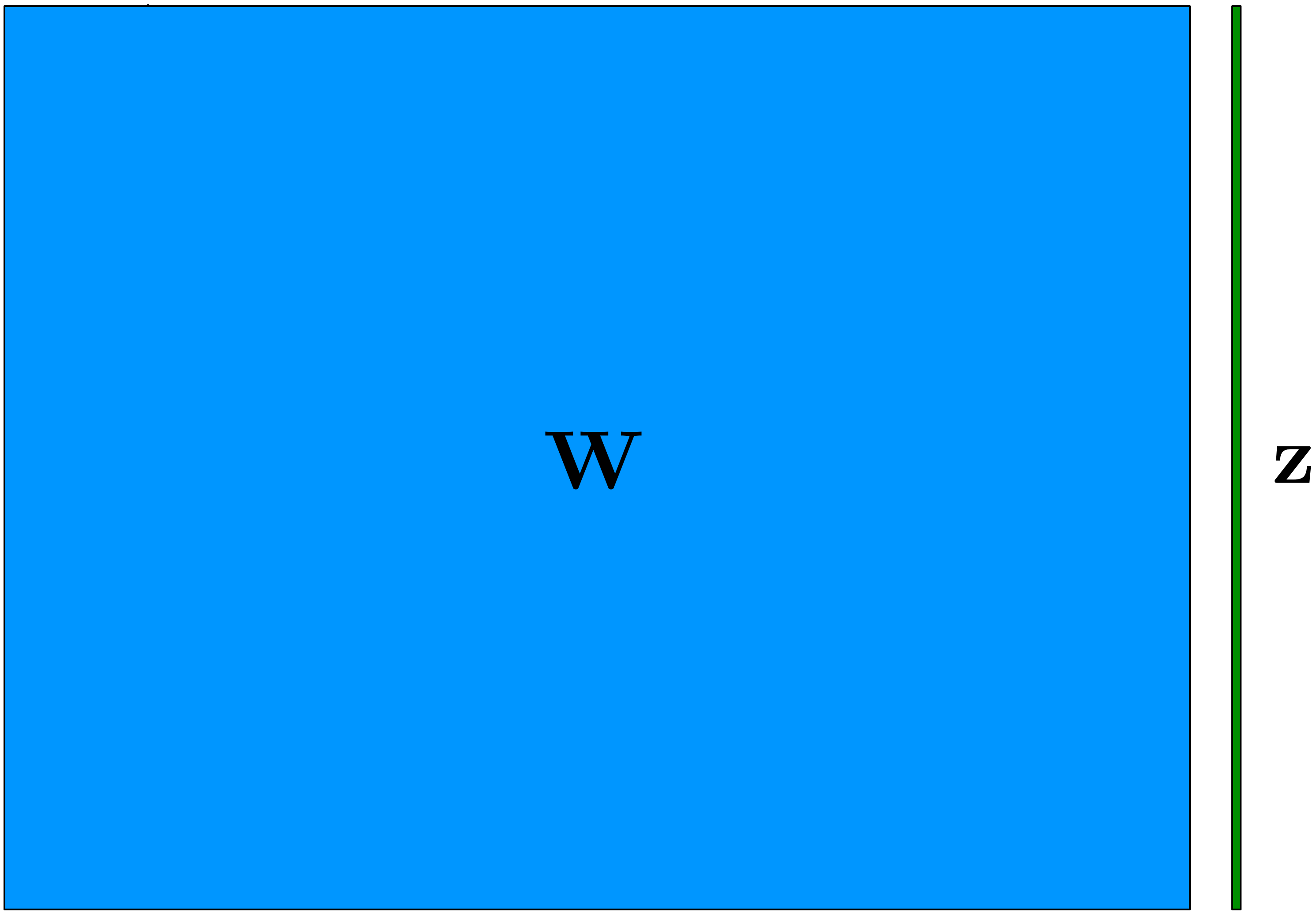
- Eigenvectors of the Laplacian are *optimal* in approximating functions with L_2 bounded gradient. (Aflalo, Brezis, Kimmel '15)

- Affinity eigenvalues decay (very) rapidly. (Talebi,and M. '15, Meyer and Shen, '12)

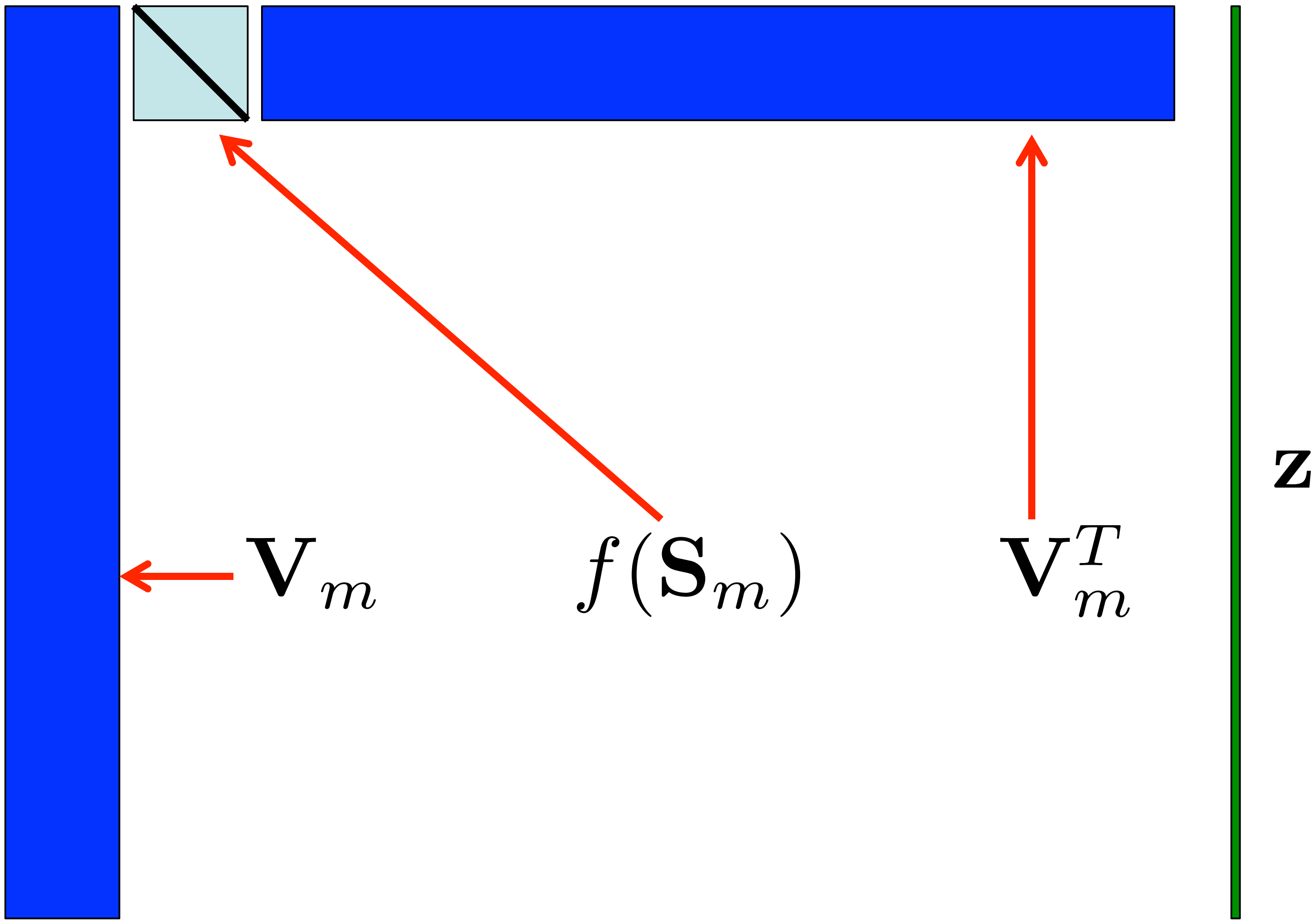
$$f(\mathbf{z}) = \mathbf{z}^T \mathbf{L}\, \mathbf{z} = \sum_i \alpha_i^2 \lambda_i$$



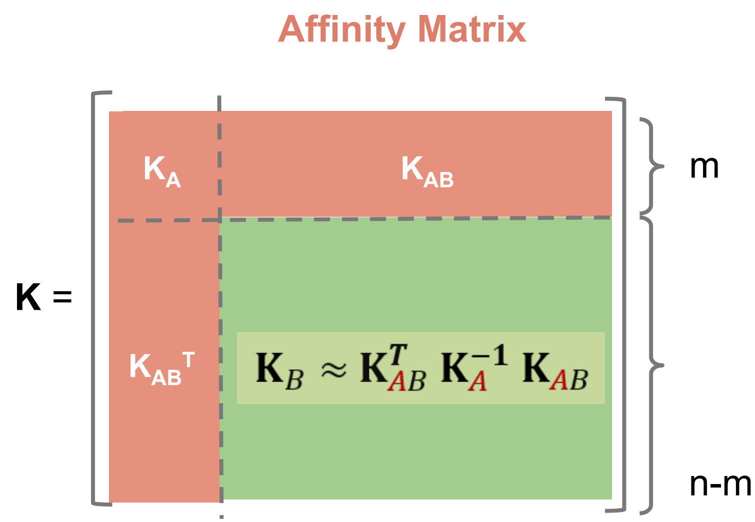
Spectral Filtering in Lower Dimension



Spectral Filtering in Lower Dimension



Nystrom Approximation



Explicitly computed weights

Approximated weights

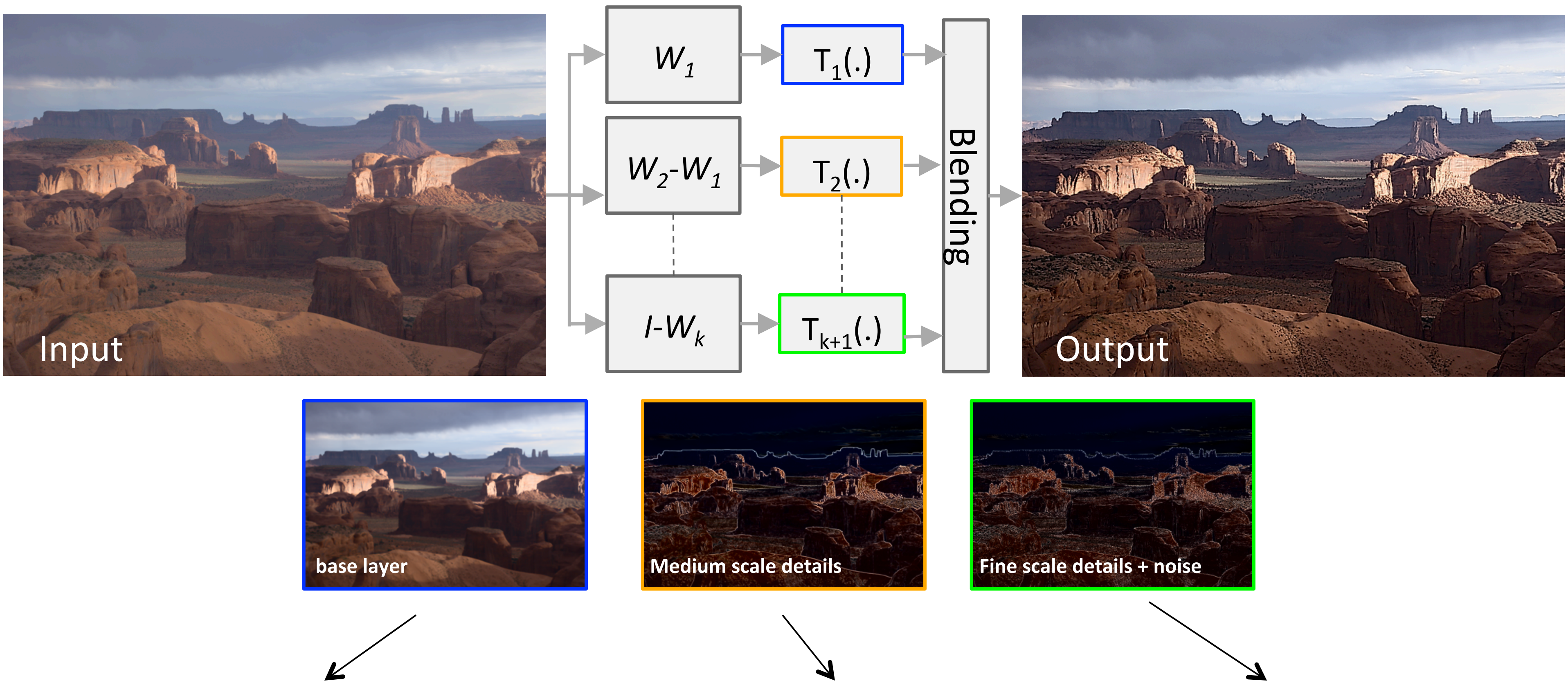
Spatially uniform sampling

A: Sampled pixels (m)

B: Remaining pixels (n-m)

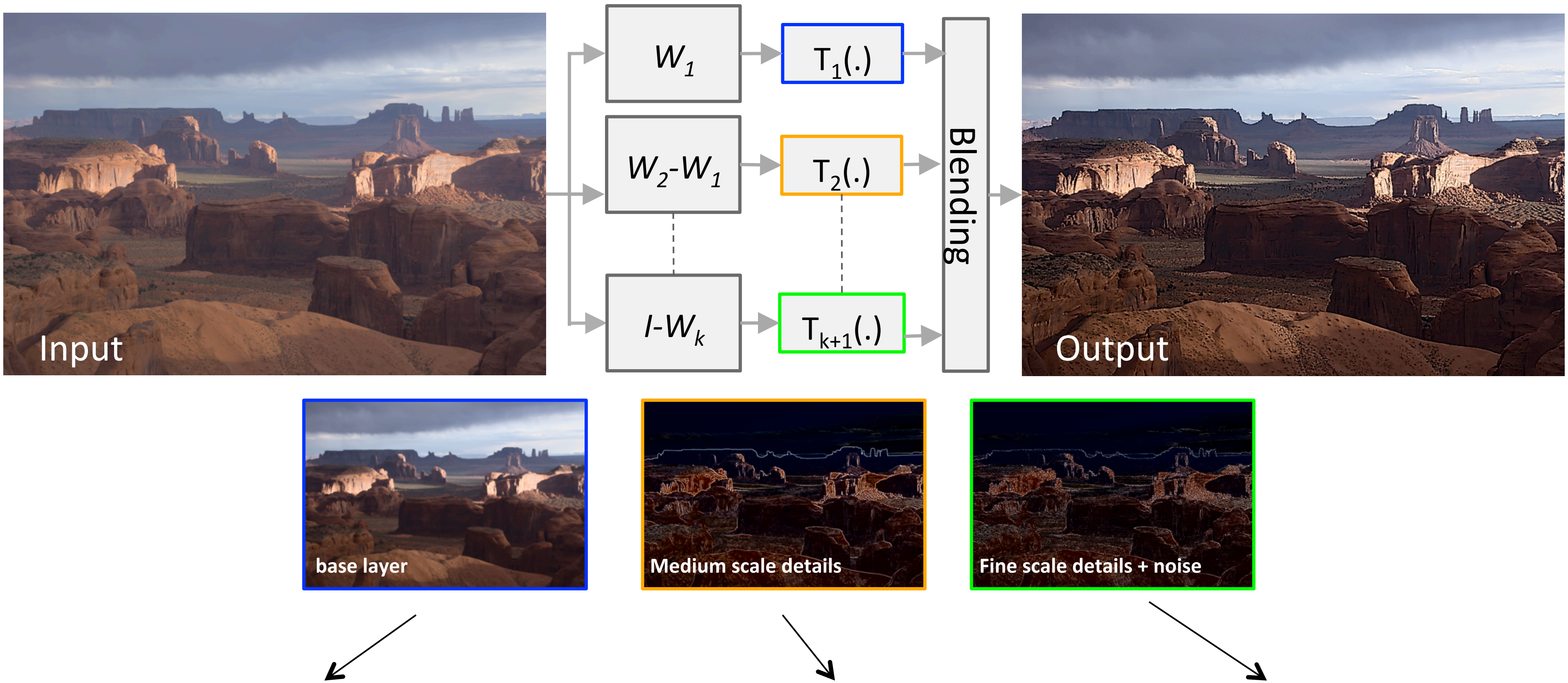
We go from sampled affinities directly to the eigenvectors.

Multi-Laplacian Scale Decomposition



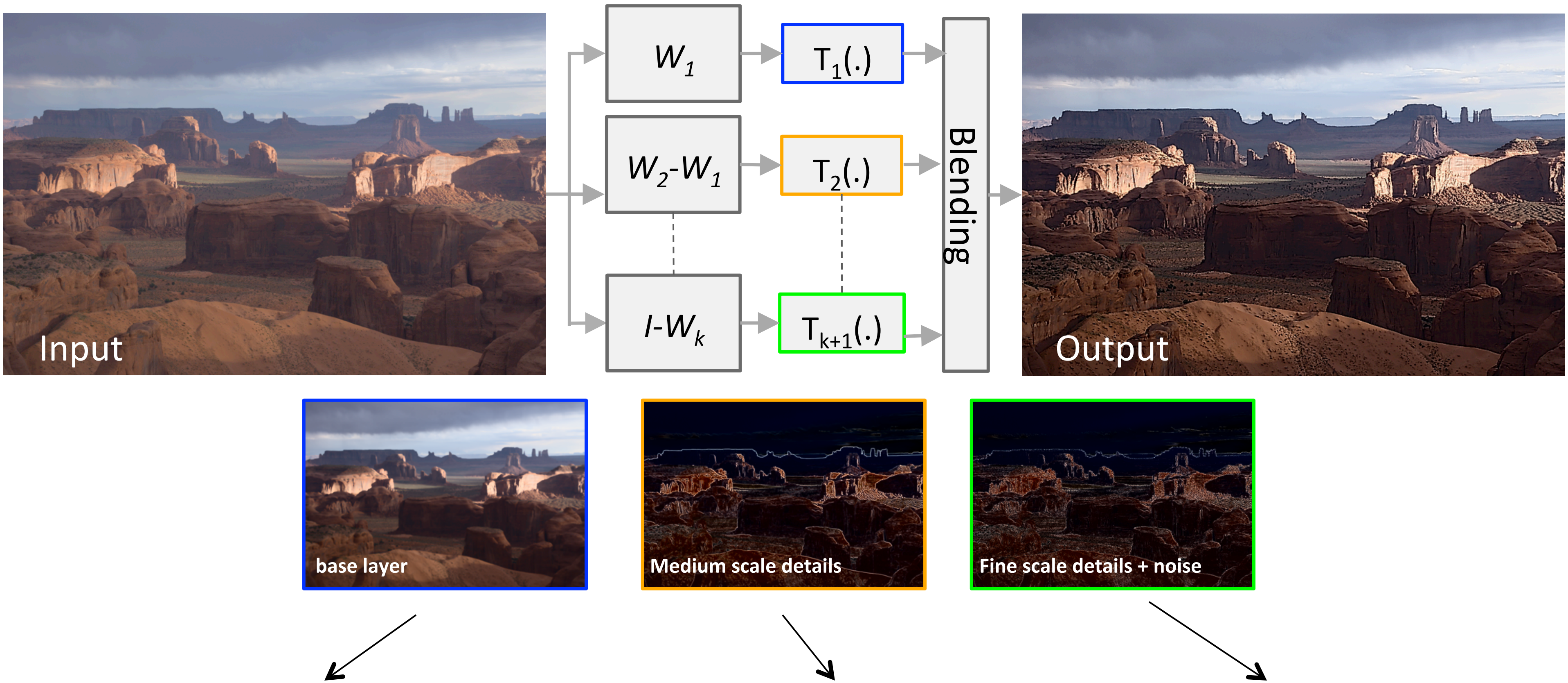
$$\mathbf{z} = \beta_1\, \mathbf{y}_{smooth} + \beta_2\, \mathbf{y}_{detail_1} + \cdots + \beta_{k+1}\, \mathbf{y}_{detail_k}$$

Multi-Laplacian Scale Decomposition



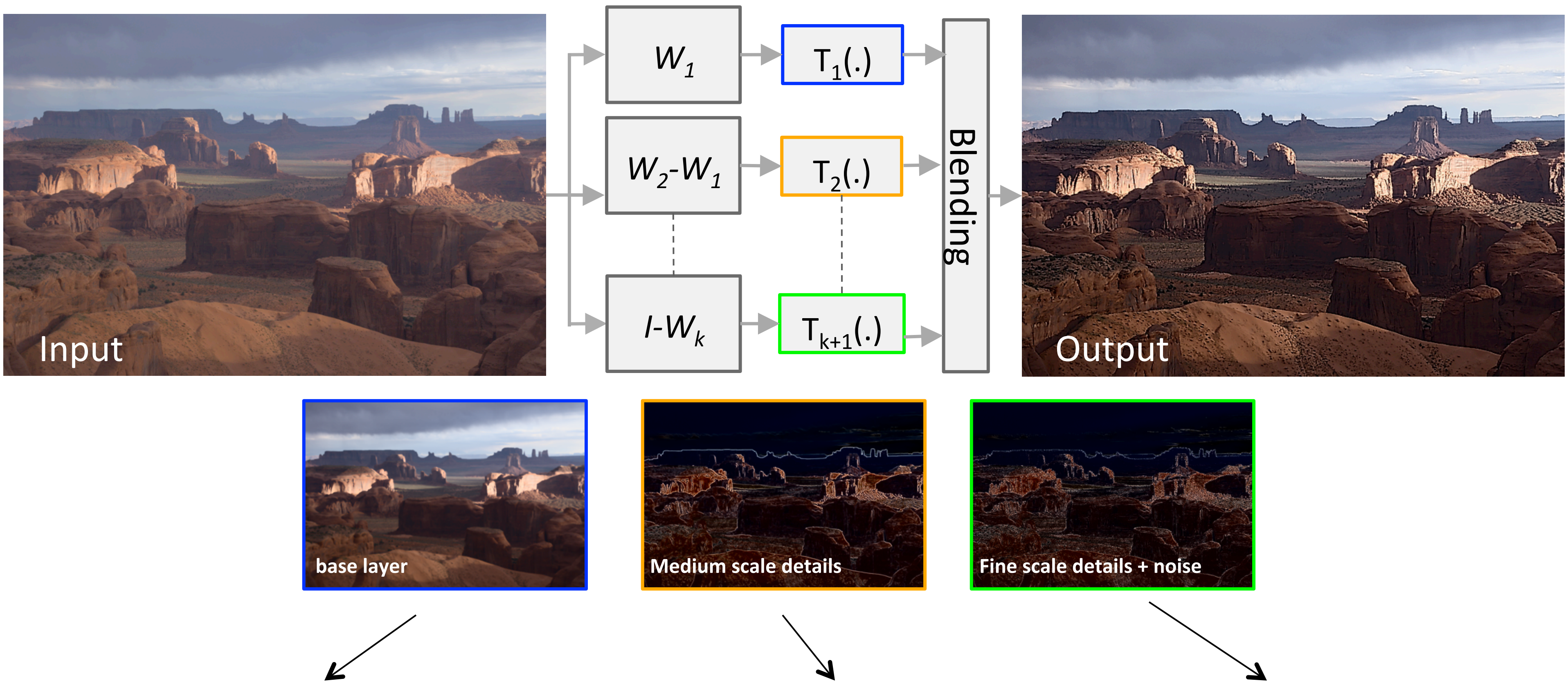
$$\mathbf{z} = \beta_1 \mathbf{W}_1 \mathbf{y} + \beta_2 (\mathbf{W}_2 - \mathbf{W}_1)\mathbf{y} + \cdots + \beta_{k+1}(\mathbf{I} - \mathbf{W}_k)\mathbf{y}$$

Multi-Laplacian Scale Decomposition



$$\mathbf{z} = \beta_1 \mathbf{W}^k \mathbf{y} + \beta_2 \mathbf{W}^{k-1}(\mathbf{I} - \mathbf{W})\mathbf{y} + \cdots + \beta_{k+1}(\mathbf{I} - \mathbf{W})\mathbf{y}$$

Multi-Laplacian Scale Decomposition



$$\mathbf{z} = \beta_1 \mathbf{y} + (\beta_1 - \beta_2)\mathbf{L}_1 \mathbf{y} + \cdots + (\beta_k - \beta_{k+1})\mathbf{L}_k \mathbf{y}$$

Some Examples

Low-light Imaging

Low-light Imaging

Low-light Imaging

Low-light Imaging

Dehazing

Dehazing

Dynamic Range Expansion

Dynamic Range Expansion

Dynamic Range Expansion

Dynamic Range Expansion

HDR Tone Mapping

HDR Tone Mapping

HDR Tone Mapping

HDR Tone Mapping

HDR Tone Mapping

HDR Tone Mapping

Compression Artifact Removal

Compression Artifact Removal

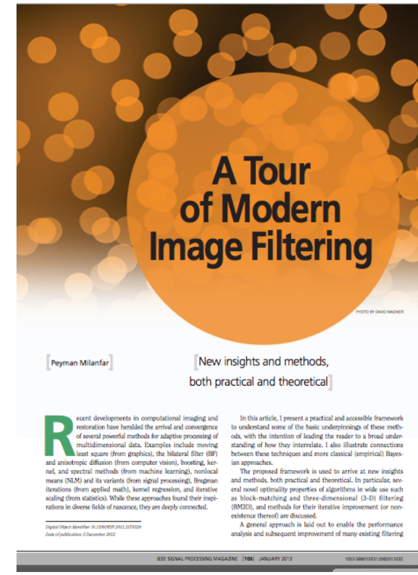
Noise-aware Sharpening/Contrast+

Noise-aware Sharpening/Contrast+

Thank you.

Relevant Papers

- "A Tour of Modern Image Filtering", P. Milanfar, IEEE Signal Processing Magazine, no. 30, pp. 106–128, Jan. 2013
- "A General Framework for Regularized, Similarity-based Image Restoration", A. Kheradmand, and P. Milanfar, IEEE Trans on Image Processing, vol. 23, no. 12, Dec. 2014
- "Global Image Denoising", H. Talebi, and P. Milanfar, IEEE Trans on Image Processing, vol. 23, no. 2, pp. 755-768, Feb. 2014
- "Nonlocal Image Editing", H. Talebi, and P. Milanfar, IEEE Trans on Image Processing, vol. 23, no. 10, Oct. 2014



A Tour of Modern Image Filtering

[Peyman Milanfar]

[New insights and methods, both practical and theoretical]

Recent developments in computational imaging and restoration have heralded the arrival and convergence of several powerful methods for adaptive processing of multidimensional data. Examples include moving least square (from graphics), the bilateral filter (BF) and anisotropic diffusion (from computer vision), boosting, kernel, and spectral methods (from machine learning), nonlocal means (NLM) and its variants (from signal processing), Bregman iterations (from applied math), kernel regression, and iterative scaling (from statistics). While these approaches found their inspirations in diverse fields of nascence, they are deeply connected.

In this article, I present a practical and accessible framework to understand some of the basic underpinnings of these methods, with the intention of leading the reader to a broad understanding of how they interrelate. I also illustrate connections between these techniques and more classical (empirical) Bayesian approaches.

The proposed framework is used to arrive at new insights and methods, both practical and theoretical. In particular, several novel optimality properties of algorithms in wide use such as block-matching and three-dimensional (3-D) filtering (BM3D), and methods for their iterative improvement (or nonexistence thereof) are discussed.

A general approach is laid out to enable the performance analysis and subsequent improvement of many existing filtering

Digital Object Identifier 10.1109/MSP.2011.2179329
Date of publication: 5 December 2012

IEEE SIGNAL PROCESSING MAGAZINE [106] JANUARY 2013 1053-5888/13/$31.00©2013IEEE

http://milanfar.org