Locally Adaptive, Nonparametric Visual Signal Processing

Peyman Milanfar

EE Department
University of California, Santa Cruz

http://www.ee.ucsc.edu/~milanfar

PSIVT, 2010 Singapore
Modern “Image Processing”

- Computational Photography (SIGGRAPH, ICCP, PAMI)
- Applied Mathematics (SIAM Imaging)
- Machine Vision (CVPR, PAMI)
- Machine Learning (NIPS)
- Non-parametric Statistics (Annals of Statistics)
- Graphics (SIGGRAPH)
- Classical Signal Processing (Trans. on Image Proc.)
Many Deeply Related Concepts

- Bilateral Filter
- Anisotropic Diffusion
- Moving Least-Squares
- Non-local Means
- Locally Adaptive Kernel Regression
- Spectral Clustering
- Bregman Iterations, Boosting
The Many Problems of Imaging

A real scene

Blurring effect

Down-sampling effect

Noise effect

A limited number of pixels

Sensor/Compression

Measurements

Atmosphere

Lens

Deblurring problem

Upscaling problem (or Interpolation)

Denoising problem
The Common Framework

• A data-fitting problem

\[ y_i = z(x_i) + e_i, \quad \text{for} \quad i = 1, \cdots, n, \]

- The regression function
- Zero-mean, i.i.d noise (No other assump.)
- Given samples
- The sampling position
- The number of samples

• The particular form of \( z(x) \)
  may remain unspecified for now.
Kernel Regression

• The data model:

\[ y_i = z(x_i) + e_i, \quad \text{for} \quad i = 1, \ldots, n, \]

• The non-parametric (point) estimate:

\[ \hat{z}(x_j) = \arg \min_{z(x_j)} \sum_{i=1}^{n} [y_i - z(x_j)]^2 K(x_i, x_j, y_i, y_j) \]

Measure of similarity between two data points \( i \) and \( j \).
Solving for the (Point) Estimate: Matrix Formulation

• It’s just a weighted Least Squares problem:

\[
\hat{z}(x_j) = \arg \min_{z(x_j)} [y - z(x_j)1_n]^T K_j [y - z(x_j)1_n]
\]

where

\[
y = \begin{bmatrix}
y_1 \\
\vdots \\
y_n
\end{bmatrix}, \quad 1_n = \begin{bmatrix}
1 \\
\vdots \\
1
\end{bmatrix}, \quad K_j = \text{diag}\left[\begin{array}{c}
K(x_1, x_j, y_1, y_j) \\
K(x_2, x_j, y_2, y_j) \\
\vdots \\
K(x_n, x_j, y_n, y_j)
\end{array}\right]
\]
Solution: Locally Adaptive Filters

$$\hat{z}(x_j) = \arg \min_{z(x_j)} [y - z(x_j)1_n]^T K_j [y - z(x_j)1_n]$$

$$\downarrow$$

$$\hat{z}(x_j) = \left(1_n^T K_j 1_n\right)^{-1} 1_n^T K_j y$$

$$= \sum_i \frac{K(x_i, x_j, y_i, y_j)}{\sum_i K(x_i, x_j, y_i, y_j)} y_i$$

$$= \sum_i W_{i,j} y_i$$

$$= w_j^T y.$$
Some Special Cases

- **Classical Gaussian Linear Filters:**

\[
K(x_i, x_j, y_i, y_j) = \exp \left( -\frac{\|x_i - x_j\|^2}{h_x^2} \right)
\]

![Diagram showing spatial distance and kernel function](image)
Some Special Cases

- **Bilateral Filter** (Tomasi, Manduchi, '98)
- **Non-local Means** (Buades. et al. 2005)

\[ K(x_i - x) \cdot K(y_i - y) \]

\( \delta x = |x_i - x| \)

\( \delta y = |y_i - y| \)

\( K(y_i - y) \)

\( K(x_i - x) \)

The photometric distance

The spatial distance

\( \sqrt{\delta x^2 + \delta y^2} \)

The Euclidean distance
Special Cases: Bilateral Filter

\[ K(x_i, x_j, y_i, y_j) = \exp \left\{ -\frac{\|x_i - x_j\|^2}{h_x^2} + \frac{-(y_i - y_j)^2}{h_y^2} \right\} \]

**Spatial similarity**

**Photometric similarity**

Shown in non-overlapping patches (for convenience of illustration only)
Special Cases: Non-local Means

\[ K(x_i, x_j, y_i, y_j) = \exp \left\{ \frac{-\|x_i - x_j\|^2}{h_x^2} + \frac{-\|y_i - y_j\|^2}{h_y^2} \right\} \]

Shown in non-overlapping patches (for convenience of illustration only)
More Special Cases

- **LARK Steering Kernel** (Takeda et al. 2007)

The photometric distance
\[ \delta y = |y_i - y| \]

The spatial distance
\[ \delta x = |x_i - x| \]

The geodesic distance
\[ K(x_i - x) \]
LARK

\[ K(x_i, x_j, y_i, y_j) = \exp \left\{ -(x_i - x_j)^T C_{ij} (x_i - x_j) \right\} \]

Estimated Local gradient covariance
- "Structure Tensor"
- "Metric Tensor"
Image as a Surface Embedded in the Euclidean 3-space

\[ S(x_1, x_2) = \{x_1, x_2, z(x_1, x_2)\} \in \mathbb{R}^3 \]

Arclength on the surface

\[
    ds^2 = dx_1^2 + dx_2^2 + dz^2
    = dx_1^2 + dx_2^2 + (z_{x_1} dx_1 + z_{x_2} dx_2)^2
    = (1 + z_{x_1}^2) dx_1^2 + 2 z_{x_1} z_{x_2} dx_1 dx_2 + (1 + z_{x_2}^2) dx_2^2
\]

\[
    = (dx_1 \quad dx_2) \begin{pmatrix}
        1 + z_{x_1}^2 & z_{x_1} z_{x_2} \\
        z_{x_1} z_{x_2} & 1 + z_{x_2}^2
    \end{pmatrix} \begin{pmatrix}
        dx_1 \\
        dx_2
    \end{pmatrix}
\]

\[
    (x_l - x)^T (C_l + I) (x_l - x) = (\text{Local geodesic distance})^2
    \text{ or Riemannian metric}
\]

**LARK:** \[ K(C_l, x_l, x) = \exp \{ -(x_l - x)^T C_l (x_l - x) \} \]
Comparisons

Bilateral  Non-local Means  LARK
Robustness of LARK Descriptors

Original image
Brightness change
Contrast change
WGN sigma = 10
Data loss (80%)
Generalizations I

- General Gaussian Kernel with \( t = \begin{bmatrix} x \\ y \end{bmatrix} \)

\[
K(t_i, t_j) = \exp \left\{ - (t_i - t_j)^T Q_{i,j} (t_i - t_j) \right\}
\]

\[
Q_{i,j} = \begin{bmatrix} Q_x & 0 \\ 0 & Q_y \end{bmatrix}
\]

Symmetric, positive-definite

- Classical: \( Q_x = \frac{1}{h_x^2} I \) and \( Q_y = 0 \)

- Bilateral: \( Q_x = \frac{1}{h_x^2} I \) and \( Q_y = \frac{1}{h_y^2} \text{diag}[0, 0, \cdots, 1, \cdots, 0, 0] \)

- Non-local Means: \( Q_x = 0 \) and \( Q_y = \frac{1}{h_y^2} G \)

- LARK: \( Q_x = C_i(y) \) and \( Q_y = 0 \).
Generalizations II

\[ K(t_i, t_j) = \exp \left\{ -(t_i - t_j)^T Q_{i,j}(t_i - t_j) \right\} \]

\[ Q_{i,j} = \begin{bmatrix} Q_x & 0 \\ 0 & Q_y \end{bmatrix} \]

• Introduce off-diagonal blocks for Q
• Define the “feature” vector \( t \) more generally
• Reproducing Kernels
  – (Mercer –’09, Aronszajn, Bergman ’50)
Generalizations III

• Admissible Kernels

• $K(t, s) = K(s, t)$

• Positive definiteness:

  For $\{t_i\}_{i=1}^n$, the Gram matrix $K_{i,j} = K(t_i, t_j)$ is symmetric positive definite.

• Given $K_1(t, s)$, and $K_2(t, s)$

  – Endless new constructions are possible:
    
    $K(t, s) = \alpha K_1(t, s) + \beta K_2(t, s)$  \quad $\alpha, \beta \geq 0$
    
    $K(t, s) = K_1(t, s) K_2(t, s)$
    
    $\cdots$
Generalizations IV

- Describe the regression function $z$ in a basis

$$\hat{z}(x_j) = \arg \min_{z(x_j)} \sum_{i=1}^{n} [y_i - z(x_j)]^2 \ K(x_i, x_j, y_i, y_j)$$

$$\hat{z}(x_j) = \arg \min_{\beta_l(x_j)} \sum_{i=1}^{n} \left[ y_i - \sum_{l=0}^{N} \beta_l(x_j) \phi_l(x_i, x_j) \right]^2 \ K(y_i, y, x_i, x_j)$$

Basis functions can be:
  - polynomials, wavelets, etc.
  - learned from data/examples
    - (e.g. K-SVD, Elad et al. ’08)
An Example: Using a Polynomial Basis + Classic Kernel

Zeroth order (N = 0): Constant model $\beta_0$

First order (N = 1): Linear model $\beta_0 + \beta_1 (x - x_i)$

Second order (N = 2): Quadratic model $\beta_0 + \beta_1 (x - x_i) + \beta_2 (x - x_i)^2$

Regression order N trades of bias/variance
Generalizations IV

- In matrix form:

\[ \hat{z}(x_j) = \arg \min_{z(x_j)} [y - z(x_j)1_n]^T K_j [y - z(x_j)1_n] \]

\[ \hat{\beta}(x_j) = \arg \min_{\beta(x_j)} [y - \Phi_j \beta(x_j)]^T K_j [y - \Phi_j \beta(x_j)] , \]

\[ \hat{z}(x_j) = \phi_j^T \hat{\beta}(x_j) \]

- Weighted combination of all the data:
  - weight $\sim$ similarity + basis
  - weights add up to 1
  - No longer positive!
What Do the Weights Look Like?
To summarize so far ……

• **Classic Kernel Regression:**
  Locally *Linear, Shift-varying* Filters:

  \[
  \hat{z}(x_j) = \sum_i W(x_i, x_j) y_i
  \]

• **Data-Adaptive Kernel:**
  Locally *Non-Linear, Shift-varying* Filters:

  \[
  \hat{z}(x_j) = \sum_i W(x_i, x_j, y_i, y_j) y_i
  \]
Many Applications…..

• Denoising
• Interpolation
• Super-resolution
• Deblurring
• …..
Film Grain Reduction (Real Noise)

Noisy image
Film Grain Reduction
(Real Noise)

LARK
Film Grain Reduction (Real Noise)
The State of the Art

- Several Tightly Competitive Algorithms
- Still some room for improvement
  - “Is Denoising Dead?” [Chatterjee, M., TIP 2010]
Recovering Sparsely Sampled Images

Randomly delete 85% of pixels

Reconstruction
Adaptive Kernels for Interpolation

- What about missing pixels?
  - Photometric distance is undefined!
  - Using a “pilot” estimate, fill the missing pixels:
Super-resolution

Motion Estimation → Adaptive Kernel Regression

Image Reconstruction (Interpolation, Denoising, and Deblurring)
An Immodest Plug

• New Book:
The Matrix Formulation

- Collect the vector formulation for all $j = 1, \ldots, n$

$$\hat{z}(x_j) = \sum_i W_{i,j} y_i = w_j^T y.$$
The Matrix \( W \)

- \( W \) is very special:

\[
    w_j^T y = \sum_i W_{i,j} y_i = \sum_i \frac{K_{i,j}}{\sum_j K_{i,j}} y_i
\]

\[\longrightarrow\]

\[ W = D^{-1} K, \quad \text{where } D_{j,j} = \text{diag}\{\sum_i K_{i,j}\} \]

\[
    W = D^{-1} K = D^{-1/2} \underbrace{D^{-1/2} K D^{-1/2}}_{L} D^{1/2}
\]

- So \( W \) is positive definite, but \underline{NOT} symmetric
  – though it is almost .....
The Matrix $W$ is Special....

\[ w_j^T y = \sum_i W_{i,j} y_i = \sum_i \frac{K_{i,j}}{\sum_i K_{i,j}} y_i \]

- $W$ is positive definite and row-stochastic ($w_j^T 1 = 1$)

- $W$ has spectral radius $0 \leq \lambda(W) \leq 1$

- $\lambda_1(W) = 1$, eig. vector: $v_1 = \frac{1}{\sqrt{n}} [1, 1, \ldots, 1]^T = \frac{1}{\sqrt{n}} 1_n$

- Ergodicity: $\lim_{k \to \infty} W^k = v_1 u_1^T = \frac{1}{\sqrt{n}} 1_n u_1^T$.

- $W$ is also ......
  - Probability Transition Matrix for a Markov Chain
  - Graphical Models: Affinity Matrix in Spectral Clustering
    - Graph Laplacian
Spectrum of the Bilateral Filter
Spectrum of Non-Local Means Filter

![Diagram showing the spectrum of non-local means filter with eigenvalue and eigenvalue index. The diagram includes various features such as Edge 1, Corner 1, Flat 1, Flat 2, Edge 2, Corner 2, Texture 1, Texture 2, and Texture 3.](image-url)
Spectrum of the LARK Filter

Eigenvalue Index

Flat 1
Flat 2
Edge 1
Corner 1
Corner 2
Edge 2
Texture 1
Texture 2
Texture 3

Eigenvalue

Eigenvalue Index
W is Almost Symmetric

• How to make it symmetric & retain its properties?

**Algorithm 1** Diagonal scaling of $W$

```plaintext
for $k = 1 : \text{iter};$
    Normalize *Columns*
    Normalize *Rows*
end
$C = \text{diag}(c); \ R = \text{diag}(r);$  
$\hat{W} = R \ W \ C$
```

- Convergence is guaranteed for *any* strictly positive $W$
- $\hat{W}$ is symmetric, positive-definite, and doubly-stochastic

*Sinkhorn’s Iterative Scaling Algorithm (’67)*
Symmetric Approximation Loses Little

• In what sense is the approximation good?
  – Darroch and Ratcliff (’72)

\[ \widehat{W} = R W C \] minimizes the cross-entropy

\[ \sum_{i,j} \widehat{W}_{i,j} \log \frac{\widehat{W}_{ij}}{W_{ij}} \]

over all doubly-stochastic \( \widehat{W} \).
Spectral Analysis

- Eigen-decomposition of $W$

$$W = VS\!V^T$$

where $S = \text{diag}[\lambda_1, \cdots, \lambda_n]$

$$0 \leq \lambda_n \leq \cdots \leq \lambda_1 = 1.$$
Statistical Analysis of Filters

- **Bias**
  \[ \| \text{bias} \|^2 \approx \| (W - I)z \|^2 \]
  
  Huang, Jordan, et al. (NIPS ’08)

- **Variance**
  \[ \text{cov}(\hat{z}) = \text{cov}(Wy) \approx \text{cov}(We) = \sigma^2 WW^T \]

- **Mean-Squared Error**
  \[ \text{MSE} = \| \text{bias} \|^2 + \text{tr}(\text{cov}(\hat{z})) \]
Statistical Analysis of Filters

- With $\mathbf{W} = \mathbf{VSV}^T$ write $\mathbf{z} = \mathbf{Vb}_0 \leftarrow \mathbf{z}$ in terms of eigenvectors of $\mathbf{W}$

- Bias

$$||\text{bias}||^2 = \sum_{i=1}^{n} (\lambda_i - 1)^2 b_{0i}^2$$

- Variance

$$\text{var}(\hat{\mathbf{z}}) = \text{tr}(\text{cov}(\hat{\mathbf{z}})) = \sigma^2 \sum_{i=1}^{n} \lambda_i^2$$

- Mean-Squared Error

$$\text{MSE} = \sum_{i=1}^{n} (\lambda_i - 1)^2 b_{0i}^2 + \sigma^2 \lambda_i^2$$
An Observation

- What is the “ideal” spectrum for \( W \)?

- Minimize the Mean-Squared Error w.r.t. \( \lambda_i \)

\[
\text{MSE}(\lambda_i) = \sum_{i=1}^{n} (\lambda_i - 1)^2 b_{0i}^2 + \sigma^2 \lambda_i^2
\]

- Optimal Spectrum:

\[
\lambda_i^* = \frac{b_{0i}^2}{b_{0i}^2 + \sigma^2} = \frac{1}{1 + \text{snr}_i^{-1}}
\]

- State of the art denoising

---

“Ideal” Wiener Filter
Improving the “non-ideal” Filter \( \hat{z} = Wy \)

- **Diffusion** (Perona, Malik ’90, Coifman et al ’06, ....)

\[
\hat{z}_k = W \hat{z}_{k-1} = W^k y \\
\hat{z}_k = \hat{z}_{k-1} + (W - I) \hat{z}_{k-1}
\]

\[
\hat{z}_k - \hat{z}_{k-1} = \left[ D^{-1/2} L D^{-1/2} \right] \hat{z}_{k-1}
\]

\[
\bar{z}_k - \bar{z}_{k-1} = L \bar{z}_{k-1} \quad \longleftrightarrow \quad \frac{\partial \bar{z}(t)}{\partial t} = \nabla^2 \bar{z}(t)
\]

with \( \bar{z}_k = D^{1/2} \hat{z}_k \)
Improving the Estimates II

- **Twicing** (Tukey ’77), **L₂-Boosting** (Buhlmann, Yu ’03), **Bregman Iteration** (Osher et al. ’05)
- Adding “roughness” to the estimate, using residuals

\[
\hat{z}_k = \hat{z}_{k-1} + W(y - \hat{z}_{k-1})
\]

- **Example:**

\[
\hat{z}_1 = \hat{z}_0 + W(y - \hat{z}_0)
\]

\[
= Wy + W(y - Wy)
\]

\[
= (2I - W) Wy
\]

Sharpening (inv. diffusion) step

Blurring (diffusion) step
Statistical Performance Analysis

• Diffusion

\[ \text{MSE}_k = \sum_{i=1}^{n} (\lambda_i^k - 1)^2 b_{0i}^2 + \sigma^2 \lambda_i^{2k} \]

Bias ↑  
Variance ↓

• Residual

\[ \text{MSE}_k = \sum_{i=1}^{n} (1 - \lambda_i)^{2k+2} b_{0i}^2 + \sigma^2 (1 - (1 - \lambda_i)^{k+1})^2 \]

Bias ↓  
Variance ↑
Examples (LARK Filter):

Diffusion Iterations

Residual Iterations
Examples (NLM Filter):
Which to Use?

• Consider the MSE per each “channel”

\[ \text{MSE}_k = \sum_{i=1}^{n} \text{MSE}_k(i) \]

• **Diffusion** improves the i-th channel iff:

\[ \log(1 + \text{snr}_i) < \log \left( \frac{1}{1 - \lambda'_i} \right) \]

  - **Channel Capacity**
  - **Filter Entropy**

• **Residual** improves the i-th channel iff:

\[ \log(1 + \text{snr}_i) > \log \left( \frac{1}{1 - \lambda'_i} \right) \]

  \[ \lambda'_i = \begin{cases} 
  \frac{\lambda_i^k}{1 - (1 - \lambda_i)^{k+1}} & \text{Diff.} \\
  \text{Resid.} 
  \end{cases} \]
Some Insights

• If the filter $\hat{z} = W y$ is
  – “ineffective” (output contains signal) ..... use residuals
  – otherwise, use diffusion

• The optimal number of iterations $k^*$ is achieved when ..... 

\[
\lambda_i' \approx \frac{1}{1 + \text{snr}_i^{-1}}
\]

“Ideal” Wiener Filter

\[
\lambda_i' = \begin{cases} 
\lambda_i^{k^*} & \text{Diff.} \\
1 - (1 - \lambda_i)^{k^*+1} & \text{Resid.}
\end{cases}
\]
Relations to (Empirical) Bayes

- Regularization (Maximum a-Posteriori)

\[ \hat{z} = \arg \min_z \frac{1}{2} \|y - z\|^2 + \frac{\lambda}{2} \mathcal{R}(z) \]

- Steepest Descent Iteration:

\[ \hat{z}_k = \hat{z}_{k-1} - \mu \left[ (\hat{z}_{k-1} - y) + \lambda \nabla \mathcal{R}(z_{k-1}) \right] \]
Relations to Bayes: Iterations

1. MAP SD:
\[ \hat{z}_{k+1} = \hat{z}_k - \mu \left[ (\hat{z}_k - y) + \lambda \nabla \mathcal{R}(\hat{z}_k) \right] \]

2. Residuals:
\[ \hat{z}_{k+1} = \hat{z}_k + W(y - \hat{z}_k) \]

3. Diffusion:
\[ \hat{z}_{k+1} = \hat{z}_k + (W - I) \hat{z}_k \]

\[ \nabla \mathcal{R}(z_k) = \frac{-1}{\mu \lambda} (W - \mu I) (y - \hat{z}_k) \]

\[ \nabla \mathcal{R}(z_k) = \frac{1}{\mu \lambda} (W - (1 - \mu) I) (y - \hat{z}_k) - \frac{1}{\mu \lambda} (I - W) \hat{y} \]
Relations to Bayes: Iterations

1. MAP SD: \[ \hat{z}_{k+1} = \hat{z}_k - \mu \left( (\hat{z}_k - y) + \lambda \nabla R(\hat{z}_k) \right) \]

2. Residuals: \[ \hat{z}_{k+1} = \hat{z}_k + W (y - \hat{z}_k) \]

3. Diffusion: \[ \hat{z}_{k+1} = \hat{z}_k + (W - I) \hat{z}_k \]

\[ R(z_k) = \frac{1}{2 \mu \lambda} (y - \hat{z}_k)^T (W - \mu I) (y - \hat{z}_k) \]

With proper Conditions on W

\[ R(z_k) = \frac{1}{2 \mu \lambda} (y - \hat{z}_k)^T ((1 - \mu)I - W) (y - \hat{z}_k) + \frac{1}{\mu \lambda} y^T (I - W) \hat{z}_k \]
The Weights as Visual Descriptors

“Visual Search”: Robustly detect objects/actions of interest within images/videos from a single query (Seo, M., ’09)
Summary: Object Detection with Local Regression Kernels

Face detection results:
Some Examples

Query

Target

query

target
The Flying LARK: Space –Time Descriptors

- Setup is similar to 2-D, but.....
- Samples from nearby frames
- Covariance matrix is now 3x3
  - Contains implicit motion information
- Space-time processing

\[ J_i = \begin{bmatrix}
  z_{x1}(x_1) & z_{x2}(x_1) & z_{x3}(x_1) \\
  \vdots & \vdots & \vdots \\
  z_{x1}(x_P) & z_{x2}(x_P) & z_{x3}(x_P)
\end{bmatrix} \]

Spatial gradients
Temporal gradients
Space-Time Descriptors:
Action Detection Example

- No Motion Estimation
- No Segmentation
- No Learning
- No Prior Information
Conclusions

• In literature, there are only seven basic plots
  – Comedy, Tragedy, Quest, Rebirth, ......

• Basic plots of modern “Image Processing?”
  – Adaptive, Non-parametric

• Many Applications