Filter, Filters, Everywhere

- BM3D
- Moving Least-Squares
- Beltrami Kernel
- Shock Filters, Diffusion
- Gaussian Filtering
- Bilateral Filter
- Wavelet Filtering
- Spline Smoother
- Non-local Means
The Smoothing Problem

Noisy samples

Clean samples

Zero-mean, i.i.d noise (No other assump.)

\[ y_i = z_i + e_i, \quad \text{for} \quad i = 1, \ldots, n \]

The number of samples

\[ z_i = z(x_i) \]
Non-parametric Regression

\[ y_i = z_i + e_i, \quad \text{for} \quad i = 1, \ldots, n \]

- The point estimate (one pixel from many):

\[ \hat{z}_j = \arg \min_{z_j} \sum_{i=1}^{n} (y_i - z_j)^2 K_{ij} \]

This Kernel measure similarity between two data points i and j.
Solution: Locally Adaptive Filters

\[ \hat{z}_j = \arg \min_{z_j} \sum_{i=1}^{n} (y_i - z_j)^2 K_{ij} \]

\[ \hat{z}_j = \sum_i \frac{K_{ij}}{\sum_i K_{ij}} y_i \]

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

\[ \hat{Z} = \mathbf{W} \mathbf{y} \]

Generally data-dependent
n xn matrix
Some Special Cases

\[ K(x_i - x_j) = \exp \left( \frac{-\|x_i - x_j\|^2}{h^2_x} \right) \]

\[ \delta x = |x_i - x| \]

The spatial distance

\[ K(x_i - x) \]

- Classical Gaussian Linear Filters:
Some Special Cases

- **Bilateral Filter** (Tomasi, Manduchi, ‘98)
- **Non-local Means** (Buades. et al. ‘05)
- **LARK** (Takeda, Farsiu, Milanfar, ‘07)

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

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

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

The Euclidean distance:
\[
\sqrt{\delta x^2 + \delta y^2}
\]

The geodesic distance (LARK):
\[
K(x_i - x)
\]
Some Popular Special Cases

- **Bilateral Filter** (Tomasi, Manduchi, ‘98)
  \[
  K_{ij} = \exp \left\{ \frac{-\|x_i - x_j\|^2}{h_x^2} + \frac{-\|y_i - y_j\|^2}{h_y^2} \right\}
  \]

- **Non-local Means** (Buades et al. ‘05)
  \[
  K_{ij} = \exp \left\{ \frac{-\|x_i - x_j\|^2}{h_x^2} + \frac{-\|y_i - y_j\|^2}{h_y^2} \right\}
  \]

- **LARK** (Takeda et al. ‘07)
  \[
  K_{ij} = \exp \left\{ -(x_i - x_j)^T \hat{C}_{ij}(y)(x_i - x_j) \right\}
  \]

- **Patches**

- **“Learned” Geodesic Distance**
Comparisons

Bilateral

Non-local Means

LARK

Shown in non-overlapping patches (for convenience of illustration only)
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
- Focus Stacking
- Deblurring
- .....
Film Grain Reduction
(Real Noise)

Noisy image
Film Grain Reduction (Real Noise)

LARK
Film Grain Reduction (Real Noise)
• What about missing pixels?
  – Photometric distance is undefined!
  – Using a “pilot” estimate, fill the missing pixels:

$$K(x_i, x_j, y_i, y_j)$$
Recovering Sparsely Sampled Images

Randomly delete 85% of pixels

Reconstruction
More Generally

- **Gaussian Kernel with**

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

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

- **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**: \( Q_x = C_{i,j} \) and \( Q_y = 0. \)
Further Generalization

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

\[ Q_{i,j} = \begin{bmatrix} Q_x & 0 \\ 0 & Q_y \end{bmatrix} \quad \text{Symmetric, positive-definite} \]

- Introduce off-diagonal blocks for \( Q \)
- Define the “feature” vector \( t \) more generally
- Use General class of Reproducing Kernels
Generalizations III

• Admissible Kernels
  • \( K(t, s) = K(s, t) \geq 0 \)
  • 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:
    - \( K(t, s) = \alpha K_1(t, s) + \beta K_2(t, s) \quad \alpha, \beta \geq 0 \)
    - \( K(t, s) = K_1(t, s) K_2(t, s) \)
    - \( \ldots \)
The Matrix $W$ (non-negative)

• ..... Is very special:

$$w_j^T y = \sum_i W_{ij} y_i = \sum_i \frac{K_{ij}}{\sum_i K_{ij}} y_i$$

$$\Rightarrow \quad W = D^{-1} K,$$  where  $$D_{jj} = \text{diag}\{\sum_i K_{ij}\}$$

• $W$ is not symmetric
  – though it is almost ..... (more on this later)
Properties of $W$

- Key properties (Perron-Frobenius Thm.):
  - $W$ is row-stochastic ($w_j^T 1 = 1$)
  - $W$ has spectral radius $1 = \lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n \geq 0$
  - Dominant left, right eigen-vector: $v = \frac{1}{\sqrt{n}} 1$ and $u > 0$
  - Ergodicity: $\lim_{k \to \infty} W^k = vu^T > 0$

- Repeatedly applying $W$ gives a constant vector $c1$. 
Graph Interpretation

\[ W = D^{-1} K \]

\[ \mathcal{L} = D^{1/2} (I - W) D^{-1/2} \]

“Graph Laplacian”

- Eigenvalues of Laplacian are 1- eigenvalues of \( W \)
- Eigenvectors of the Laplacian are \( \nu_i = D^{1/2} v_i \)
- Once you have \( W \) you have everything .......
Spectra of the LARK Filter

Eigenvalue Index

Eigenvalue

Flat

Edge 1

Corner 1

Corner 2

Texture 1

Texture 2

Texture 3

Edge 2
Some “challenges” with \( \hat{z} = W y \)

- **Ad-hoc design**
  - pick a kernel, generate \( W \), filter
  - Adjust parameters, iterate, ....

- **Asymmetry of \( W \)**
  - Filters are “inadmissible” (Cohen ’66)
  - No Bayesian interpretation (Hastie & Tibshirani ’00)
  - No orthogonal decomposition

- **We have been fixing these .......**
Remedy 1: Normalizing $W$

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

$$\text{for } k = 1 : \text{iter;}$$

 Normalize Columns
 Normalize Rows

 end

$C = \text{diag}(c); \ R = \text{diag}(r);$  
$\hat{W} = R W C$

- **Matrix Balancing**
  - Sinkhorn and Knopp ('67)
- **Iterative Proportional Scaling**
  - Darroch and Ratcliff ('72)

- Convergence is guaranteed for non-negative $W$

- $\hat{W}$ is symmetric, positive-definite, and doubly-stochastic
Is the Approximation Any Good?

• Yes! Minimizes the cross-entropy:

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

• Smaller error with increasing dimension……..

\[ \frac{1}{n} \left\| \mathbf{W} - \hat{\mathbf{W}} \right\|_F = O(n^{-1/2}) \]

RMS difference in the elements
Dominant Eigenvectors of Original W

(Non Orthogonal Basis)
Dominant Eigenvectors of Symmetrized W

(Orthogonal Basis)
Performance Analysis (W Symm.)

• Spectral Decomposition: \( W = VSV^T \)

\[ z = Vb \]

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

\[ 0 \leq \lambda_n \leq \cdots \leq \lambda_1 = 1. \]

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

Signal coefficients in the basis given by eigenvectors of \( W \)

\[ \text{Bias}^2 \quad \text{Variance} \]
An Observation

• What is the “ideal” (oracle) spectrum for W?

• Minimize the Mean-Squared Error w.r.t. $\lambda_i$

$$\text{MSE}(\lambda_i) = \sum_{i=1}^{n} (\lambda_i - 1)^2 b_i^2 + \sigma^2 \lambda_i^2$$

• Optimal Spectrum:

$$\lambda_i^* = \frac{b_i^2}{b_i^2 + \sigma^2} = \frac{1}{1 + \text{snr}_i^{-1}}$$

• Explains performance of state of the art denoising
Remedy 2: Iterations

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

\[
\hat{Z}_k = W \hat{Z}_{k-1}
\]

\[
\frac{\partial z}{\partial t} = \hat{Z}_k - \hat{Z}_{k-1} = (W - I) \hat{Z}_{k-1}
\]

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

- **Performance:**

\[
MSE_k = \sum_{i=1}^{n} \left( \lambda_i^k - 1 \right)^2 b_i^2 + \sigma^2 \lambda_i^{2k}
\]

Bias \hspace{1cm} Variance
Remedy 2: Iterations again

- Twicing (Tukey ‘77), (L_2-) Boosting (Buhlmann, Yu ‘03), Bregman Iteration (Osher et al. ‘05), Reaction-Diffusion (Nordstrom ‘90)

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

- Example:

\[
\hat{z}_1 = \hat{z}_0 + W(y - \hat{z}_0)
= Wy + W(y - Wy)
= (2I - W) Wy
\]

\[
\text{MSE}_{k} = \sum_{i=1}^{n} \left(1 - \lambda_i\right)^{2k+2} b_i^2 + \sigma^2 \left(1 - \left(1 - \lambda_i\right)^{k+1}\right)^2
\]

- Bias
- Variance

Residuals
Sharpening (inv. diffusion) step
Blurring (diffusion) step
Statistical Performance Analysis

• Diffusion

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

Bias $\uparrow$

Variance $\downarrow$

• Residual

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

Bias $\downarrow$

Variance $\uparrow$
Which to Use?

- Depends on (1) the filter, (2) the latent image, and (3) the noise level.

- **Diffusion** if W “under-smooths”
  - Doesn’t do much denoising

- **Residual** if W “over-smooths”
  - Signal is left in residuals
  - *Weak learner* in boosting
Computational Matters

• The elements of $W$ can be computed in parallel.

• $W$ can be sparsified.

• $W$ is low rank, and therefore....

• Eigen-space of $W$ can be computed by sampling (Nystrom Method)
Nystrom Approximation

- **Nyström Extension**
  - Pre-filter & sampling
  - Approx. Kernel
  - $\tilde{K} = \begin{bmatrix} K_A & K_{AB} \\ K_{AB}^T & K_{AB} \end{bmatrix}$

- **Filter Eigen-decomposition Approximation**
  - Eigenvectors
  - $\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4, \ldots, \mathbf{v}_p$
  - Eigenvalues
  - $\lambda_2, \lambda_3, \lambda_4, \ldots, \lambda_p$

- **Filter Optimization**
  - Iteration & Truncation
  - $\hat{k}, \hat{m} = \arg \min_{k,m} \text{MSE}(k,m)$
  - Optimal Filter
  - $\hat{W} = \sum_{j=1}^{\hat{m}} \lambda_j^2 \mathbf{v}_j \mathbf{v}_j^T$

- The global filter matrix can be closely approximated with a **low-rank** matrix.

- Iteration and truncation of the filter eigenmodes can improve the estimation accuracy.
Bayesian Interpretation

- Regularization

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

- Steepest Descent Iteration:

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

Empirical log-Prior

Step size

Gradient
Given $W$, what is $R$?

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

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

\[ \nabla R(z_k) = \frac{1}{\mu \lambda} (W - (1 - \mu) I) (y - \hat{z}_k) - \frac{1}{\mu \lambda} (I - W) \hat{y} \]
Given $W$, what is $R$?

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

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

With symmetric $W$ !!

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

Simplify......

\[ \hat{z}_k = z \quad \text{and} \quad \mu = 1 \]

\[ \hat{p}(z) = c \exp \left[ -\hat{R}(z) \right] \]

• Residuals:

\[ \hat{p}(z) = c \exp \left[ -\frac{1}{2} z^T (W - I) z + y^T (W - I) z \right] \]

• Diffusion:

\[ \hat{p}(z) = c \exp \left[ -\frac{1}{2} z^T W z + y^T z \right] \]

The converse is an open problem .......

Given an (empirical) prior, what kernel is implied?
Conclusions

• The $\hat{z} = W y$ framework is very general.

• Many applications

• Kernel filters can almost always be improved.

• Lot of nice theory and many open problems:
  – Local models for signal
  – $W$ negative;
  – Kernelizing Bayesians
  – Global Filtering and Convergence Results
Further Reading

