An Empirical Evaluation of Denoising Techniques for Streaming Data

Jeremy L Thompson

United States Air Force Academy
Lawrence Livermore National Laboratory

jeremy.thompson@usafa.edu

July 28, 2014

Overview

We focus on denoising techniques for streaming data that is analyzed while being collected. We investigate spatial filters, such as the Box filter, Gaussian smoothing, and the Bilateral filter, and a statistical neighborhood filter, Non-Local Means.

We discuss practical concerns for incremental implementation, such as edge treatment, incremental updating, and parameter stability.

We make recommendations, specifically for the use of the Bilateral filter or a combination of the Bilateral and Non-Local Means filters.

Overview

1. Background
2. Overview of Techniques
3. Practical Concerns
4. Parameter Stability
5. Summary

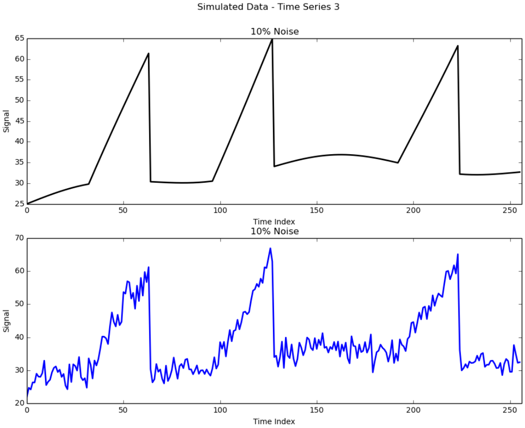
Noise in Real World Time Series

Time Series:

- Weather data
- Test telemetry
- Sample averages

Noise Sources:

- Measurement errors
- Processing errors
- Standard errors of the mean



Notation

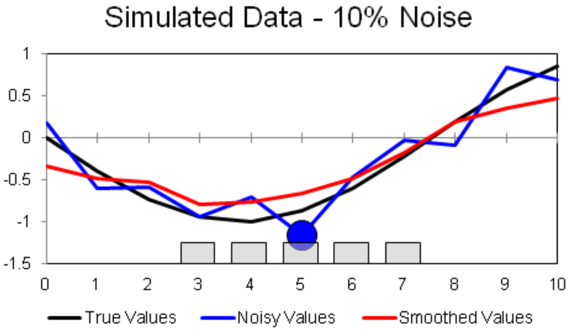
Time Series - temporally orders series of observations

y_i - Time series value at time index i
s_i - Denoised series value at time index i

Additive White Gaussian Noise (AWGN) - error added to true time series, independent identically distributed real values from Gaussian distribution

σ_n - standard deviation of AGWN
$\hat{\sigma}_n$ - estimate of σ_n

Box Filter



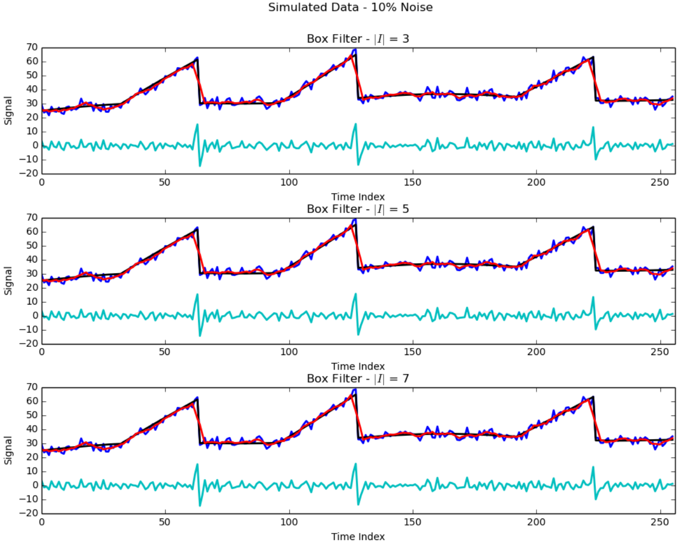
$$s_i = \sum_{j \in I} \frac{1}{|I|} y_j$$

Parameter:

- $|I|$ (Window Size)

Complexity: Low - $O\left(n \cdot \left(\frac{|I|-1}{2}\right)^2\right)$

Box Filter

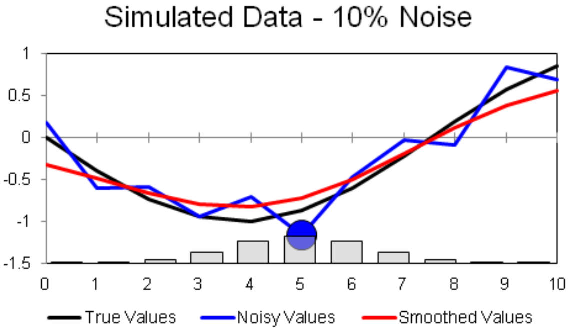


$|I| = 3$

$|I| = 5$

$|I| = 7$

Gaussian Filter



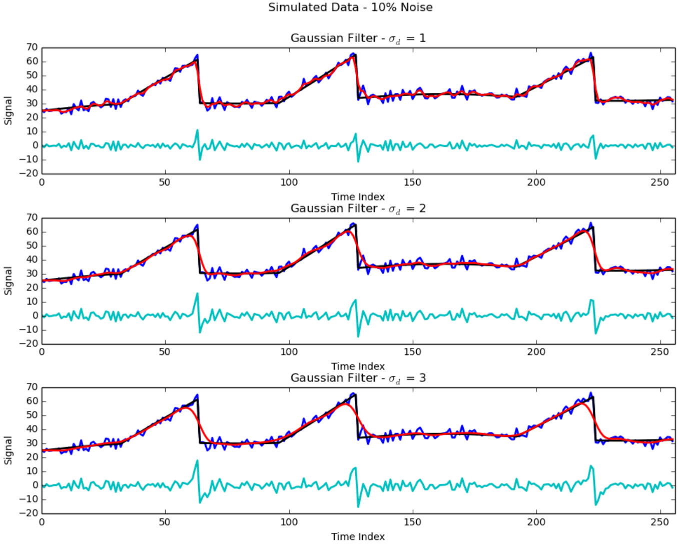
$$s_i = \sum_{j \in I} \frac{1}{z_i} e^{-\frac{|i-j|}{2\sigma_d^2}} y_j$$

Parameter:

- σ_d (Spatial Kernel)

Complexity: Low - $O\left(n \cdot \lfloor 5\sigma_d \rfloor^2\right)$

Gaussian Filter

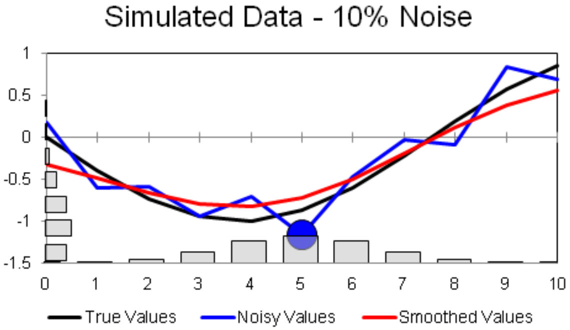


$\sigma_d = 1.0$

$\sigma_d = 2.0$

$\sigma_d = 3.0$

Bilateral Filter



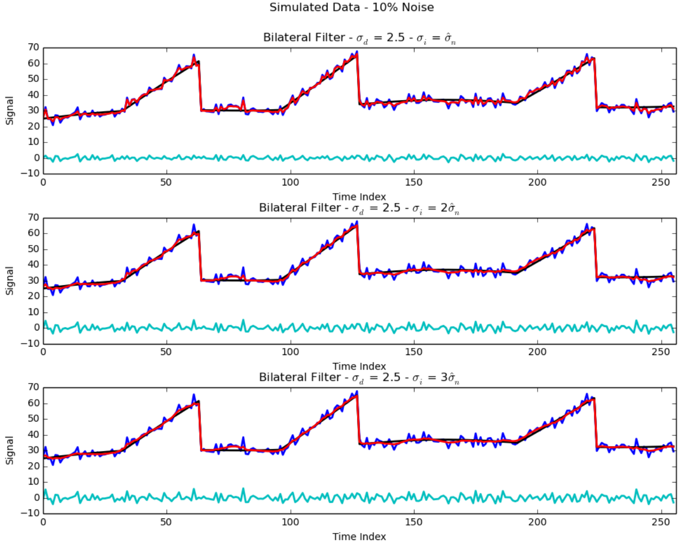
$$s_i = \sum_{j \in I} \frac{1}{z_i} e^{-\frac{|i-j|}{2\sigma_d^2}} e^{-\frac{|y_i - y_j|}{2\sigma_i^2}} y_j$$

Parameters:

- σ_d (Spatial Kernel)
- $\sigma_i = k\hat{\sigma}_n$ (Intensity Kernel)

Complexity: Moderate - $O\left(n \cdot \log\lfloor 5\sigma_d \rfloor\right)$

Bilateral Filter

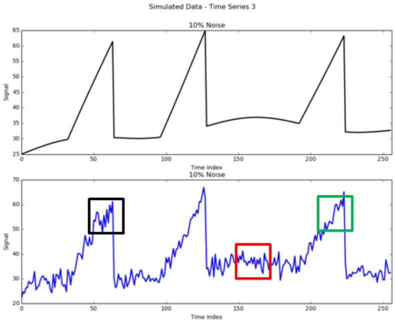


$$\sigma_d = 2.5$$
$$\sigma_i = 1.0\hat{\sigma}_n$$

$$\sigma_d = 2.5$$
$$\sigma_i = 2.0\hat{\sigma}_n$$

$$\sigma_d = 2.5$$
$$\sigma_i = 3.0\hat{\sigma}_n$$

Non-Local Means Filter



$$s_i = \sum_{j \in N} \begin{cases} \frac{1}{z_i} e^{-\frac{|Y_i - Y_j|^2}{2\beta\hat{\sigma}_n^2 |I|}} y_j & |Y_i - Y_j| < T \\ 0 & \text{otherwise} \end{cases}$$

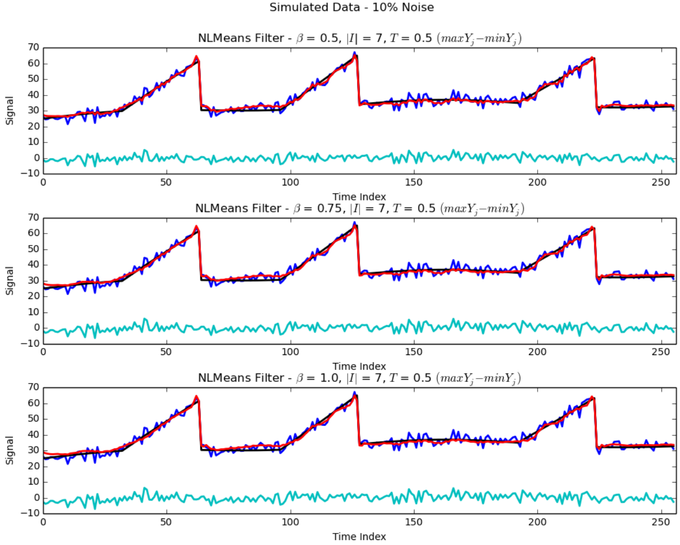
Y_j - vector of time series values around y_j

Parameters:

- β (Smoothing Level)
- $|I|$ (Window Size)
- $T = k \left(\max Y_j - \min Y_j \right) |I|$ (Pre-selection Threshold)

Complexity: High - $O\left(n^3\right)$

Non-Local Means Filter



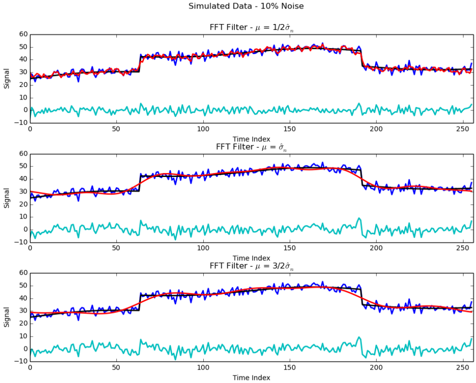
$\beta = 0.5$
$|I| = 7$
$T =$
$0.5\,(\max Y_j - \min Y_j)\;|I|$

$\beta = 0.75$
$|I| = 7$
$T =$
$0.5\,(\max Y_j - \min Y_j)\;|I|$

$\beta = 1.0$
$|I| = 7$
$T =$
$0.5\,(\max Y_j - \min Y_j)\;|I|$

Frequency Techniques

Fourier Transform



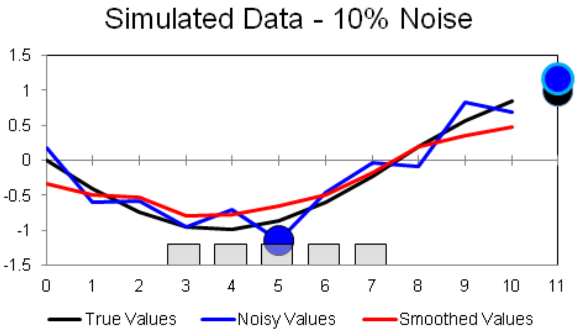
Wavelet Transform

Parameters/Options:

- Wavelet family
- Number of levels
- Threshold type
- Threshold cut-off

Fourier Transform and Wavelets - considered but not discussed

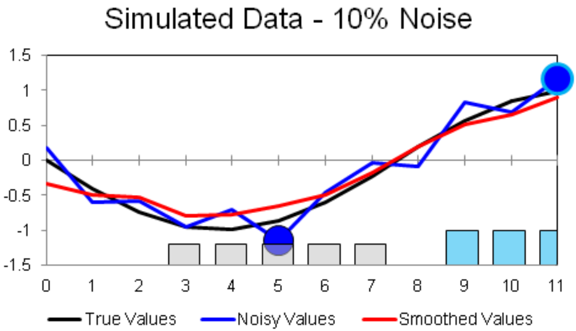
Practical Concerns



Typically denosing methods don't explicitly describe edge treatment...

but the leading edge is most relevant portion of real-time time series

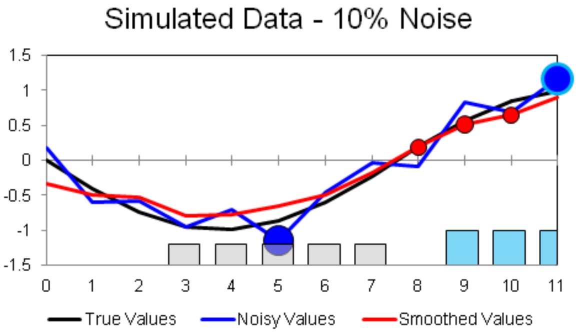
Spatial Filters



Problem: Incomplete windows to average over on edges

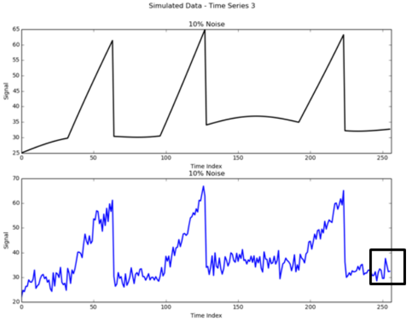
Solution: Average over the portion of the window that is available

Spatial Filters



Incremental Updating: when new data is received, update only the smoothed values of the previous edge values and add new smoothed value

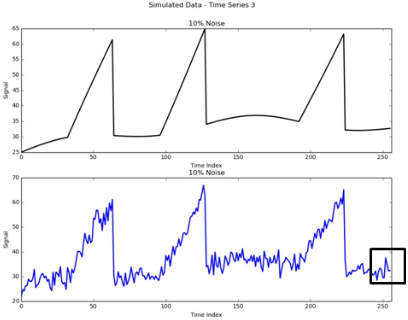
Non-Local Means Filter



Problem: Incomplete windows to compare on edges

Solution: Compare the portion of the window that is available

Non-Local Means Filter



Incremental Updating: when new data is received, update only the smoothed values of the previous edge values and add new smoothed value

Parameter Stability

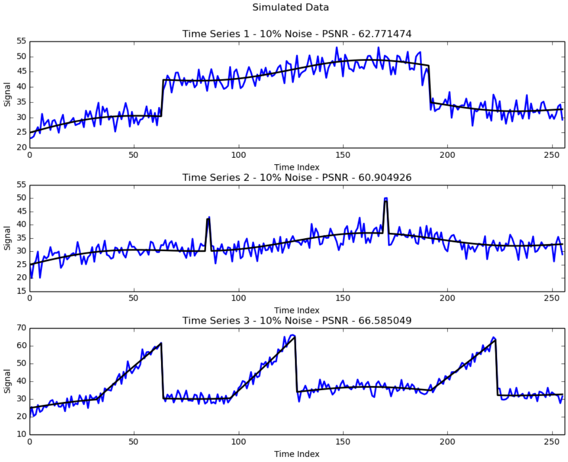
There are parameter optimization techniques, but

- Our data is dynamic
- Optimization takes time
- Bad parameters can destroy information
- Difficult to know if smoothed series is 'good'

Goal: Find methods/parameters that are stable across a variety of signals

Solution: DOE/grid search for optimal PSNR on known time series

PSNR



Time Series 1
$PSNR = 62.771$

Time Series 2
$PSNR = 60.905$

Time Series 3
$PSNR = 66.585$

PSNR - Peak Signal to Noise Ratio

$$PSNR = 10log_{10}\left(\frac{(\max y_i)^2}{MSE}\right)$$

$$MSE = \frac{1}{n}\sum_{i\in N}(\hat{y}_i - y_i)^2$$

DOE/Grid Search

Investigated smoothing performance at grid points and increased resolution in areas of interest

- Noise: 1%, 5%, 10%, 20%, and 30%
- $|I|$: 3, 5, 7, and 9
- σ_d: 0.1, 1.0, 2.0, 3.0, and 4.0
- σ_i: $0.1\hat{\sigma}_n$, $1.0\hat{\sigma}_n$, $2.0\hat{\sigma}_n$, $3.0\hat{\sigma}_n$, and $4.0\hat{\sigma}_n$
- β: 0.5, 0.75, and 1.0
- T: $0.25\left(\max Y_j - \min Y_j\right)|I|$, $0.5\left(\max Y_j - \min Y_j\right)|I|$, and $0.75\left(\max Y_j - \min Y_j\right)|I|$

Box Filter

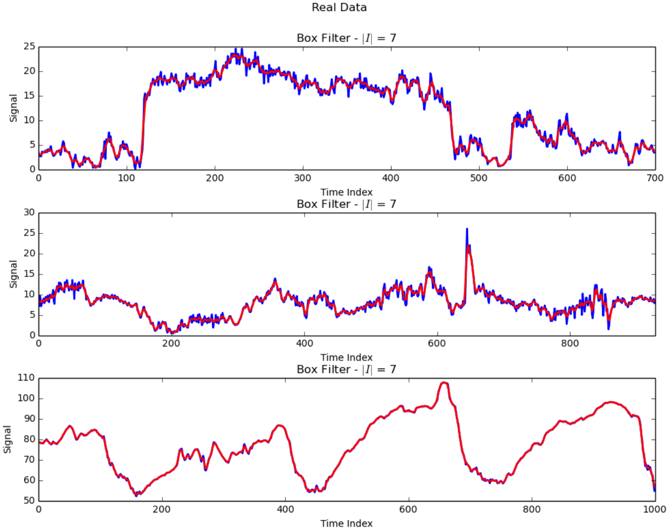
PSNR - Optimal Settings

	Time Series 1	Time Series 2	Time Series 3
1% Noise	85.594	72.175	67.346
5% Noise	81.206	70.377	66.771
10% Noise	74.047	66.201	65.354
20% Noise	64.323	58.005	60.920
30% Noise	56.888	52.199	57.145

PSNR - $|I| = 7$

	Time Series 1	Time Series 2	Time Series 3
1% Noise	85.549/99.9%	72.175/100.0%	67.346/100.0%
5% Noise	81.059/99.8%	70.181/99.7%	66.747/100.0%
10% Noise	73.249/98.9%	66.040/99.8%	64.923/99.3%
20% Noise	62.786/97.6%	57.932/99.9%	60.631/99.5%
30% Noise	56.165/98.7%	51.829/99.3%	56.082/98.1%

Box Filter


Real Data
Box Filter - |I| = 7
Signal
Time Index
0
5
10
15
20
25
100
200
300
400
500
600
700
Box Filter - |I| = 7
Signal
Time Index
30
200
400
600
800
Box Filter - |I| = 7
Signal
Time Index
50
60
70
80
90
100
110
1000

Gaussian Filter

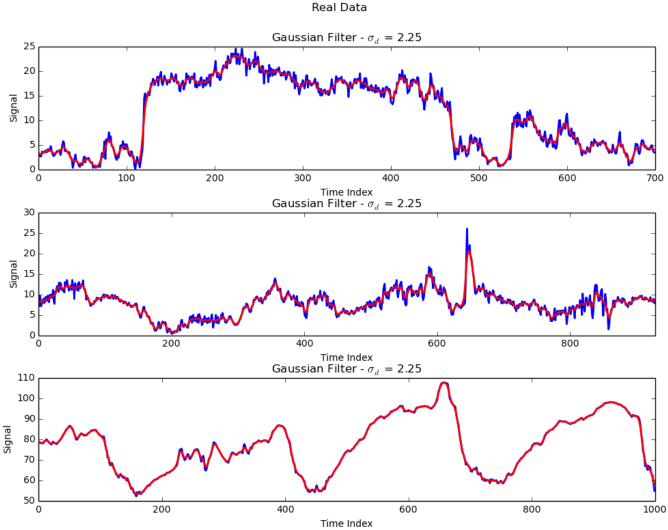
PSNR - Optimal Settings

	Time Series 1	Time Series 2	Time Series 3
1% Noise	83.257	71.659	74.807
5% Noise	80.684	70.459	79.871
10% Noise	75.844	68.646	69.177
20% Noise	69.529	60.971	62.031
30% Noise	61.243	56.495	58.172

PSNR - $\sigma_d = 2.25$

	Time Series 1	Time Series 2	Time Series 3
1% Noise	83.257/100.0%	71.659/100.0%	63.953/85.5%
5% Noise	80.684/100.0%	70.459/100.0%	63.685/79.7%
10% Noise	75.844/100.0%	67.568/98.4%	62.980/91.0%
20% Noise	67.086/96.5%	60.005/98.4%	60.437/97.4%
30% Noise	59.973/97.9%	54.554/96.6%	56.797/97.6%

Gaussian Filter



Bilateral Filter

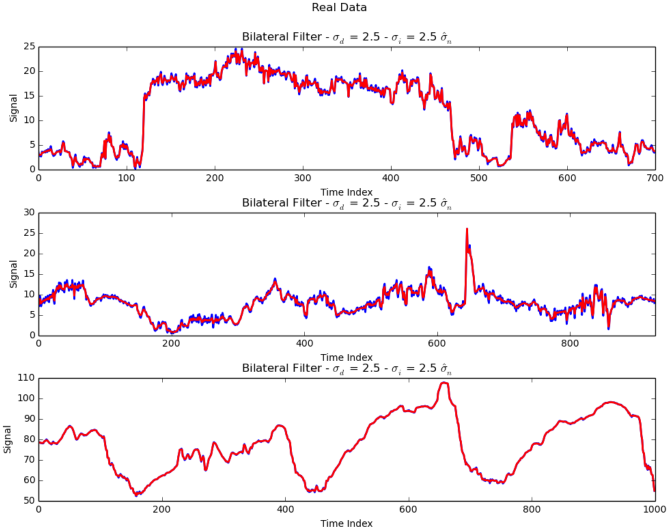
PSNR - Optimal Settings

	Time Series 1	Time Series 2	Time Series 3
1% Noise	127.251	128.396	121.790
5% Noise	97.303	96.758	96.698
10% Noise	81.774	78.128	84.987
20% Noise	69.794	63.453	70.033
30% Noise	62.657	59.049	61.736

PSNR - $\sigma_d = 2.5$, $\sigma_i = 2.5\hat{\sigma}_n$

1% Noise	124.947/98.2%	126.752/98.7%	110.432/90.7%
5% Noise	93.220/95.8%	92.841/96.0%	96.698/100.0%
10% Noise	79.580/97.3%	75.578/96.7%	84.480/99.4%
20% Noise	64.659/92.6%	60.585/95.5%	68.456/97.7%
30% Noise	57.875/92.4%	53.774/91.1%	60.891/98.6%

Bilateral Filter



Non-Local Means Filter

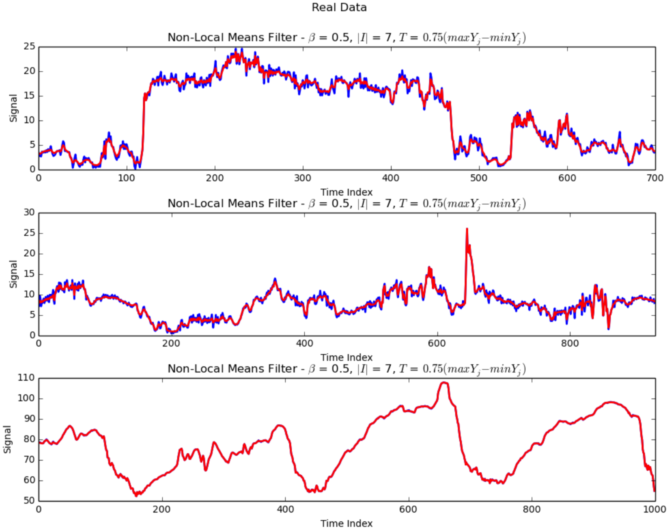
PSNR - Optimal Settings

	Time Series 1	Time Series 2	Time Series 3
1% Noise	114.852	112.238	101.050
5% Noise	89.317	89.332	92.007
10% Noise	77.760	75.828	79.996
20% Noise	65.997	60.698	68.439
30% Noise	58.928	54.630	62.217

PSNR - $\beta = 0.5,\ |I| = 7,\ T = 0.75\left(\max Y_j - \min Y_j\right)$

1% Noise	114.852/98.0%	112.238/98.1%	101.050/98.0%
5% Noise	89.317/98.6%	89.332/97.4%	92.007/98.1%
10% Noise	77.760/95.7%	75.828/95.1%	79.996/98.8%
20% Noise	65.997/95.5%	60.698/95.0%	68.439/96.6%
30% Noise	58.928/96.5%	54.630/99.6%	62.217/96.1%

Non-Local Means Filter



Bilateral and Non-Local Means Filters Combination

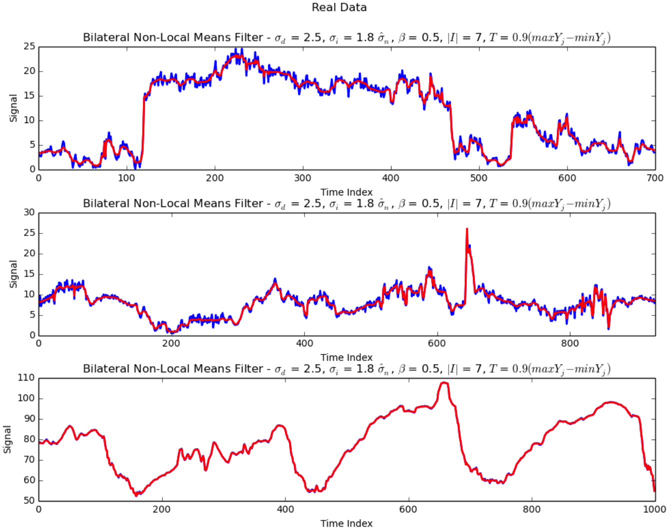
PSNR - Optimal Settings

	Time Series 1	Time Series 2	Time Series 3
1% Noise	116.272	114.415	110.231
5% Noise	100.498	97.932	97.620
10% Noise	87.153	82.641	86.879
20% Noise	73.244	66.868	75.505
30% Noise	67.219	60.023	84.472

PSNR - $\sigma_d = 2.5$, $\sigma_i = 1.8\hat{\sigma}_n$, $\beta = 0.5$, $|I| = 7$, $T = 0.9\,(\max Y_j - \min Y_j)$

1% Noise	114.572/98.5%	111.731/97.7%	94.137/85.4%
5% Noise	94.730/94.3%	94.948/97.0%	89.549/91.7%
10% Noise	79.659/91.4%	78.388/94.9%	77.953/89.7%
20% Noise	66.427/90.7%	61.955/92.7%	66.799/88.5%
30% Noise	60.274/89.7%	57.825/96.3%	84.472/100.0%

Bilateral and Non-Local Means Filters Combination



Comparison

Selected Performance at 10% Noise

	Time Series 1	Time Series 2	Time Series 3
Noisy	62.771	60.905	66.585
Box Filter	73.249/98.9%	66.040/99.8%	64.923/99.3%
Gaussian Filter	75.844/100.0%	67.568/98.4%	62.980/91.0%
Bilateral Filter	79.580/97.3%	75.578/96.7%	84.480/99.4%
Non-Local Means	77.760/95.7%	75.828/95.1%	79.996/98.8%
Bilateral NL-Means	79.659/91.4%	78.388/94.9%	77.953/89.7%

Summary

The Bilateral and Bilateral/Non-Local Means filters do a good job smoothing and appear to have stable optimal parameters

Acknowledgements

Collaborators:

- Dr Ya Ju Fan
- Dr Chandrika Kamath

SensorStreams project funded by ASCR Applied Math Program

LLNL IM Release: LLNL-PRES-657747

Future Research

1. Different noise models
2. Longer simulated time series
3. Different simulated time series
4. Improved maximum intensity windows difference estimate

An Empirical Evaluation of Denoising Techniques for Streaming Data

Jeremy L Thompson

United States Air Force Academy
Lawrence Livermore National Laboratory

jeremy.thompson@usafa.edu

July 28, 2014

Extra Slides

Extra Slides

Frequency Transform Coefficient Hard Thresholding

$v(\alpha)$ represents the transformed data, the frequency coefficients

$$v(\alpha) = \begin{cases} v(\alpha) & |v(\alpha)| > \mu \\ 0 & |v(\alpha)| < \mu \end{cases}$$

Parameter: μ

Frequency Transform Coefficient Soft Thresholding

$v(\alpha)$ represents the transformed data, the frequency coefficients

$$v(\alpha) = \begin{cases} v(\alpha) - \mu & |v(\alpha)| > \mu \\ 0 & |v(\alpha)| < \mu \end{cases}$$

Parameter: μ

An Empirical Evaluation of Denoising Techniques for Streaming Data

Jeremy L Thompson

United States Air Force Academy
Lawrence Livermore National Laboratory

jeremy.thompson@usafa.edu

July 28, 2014