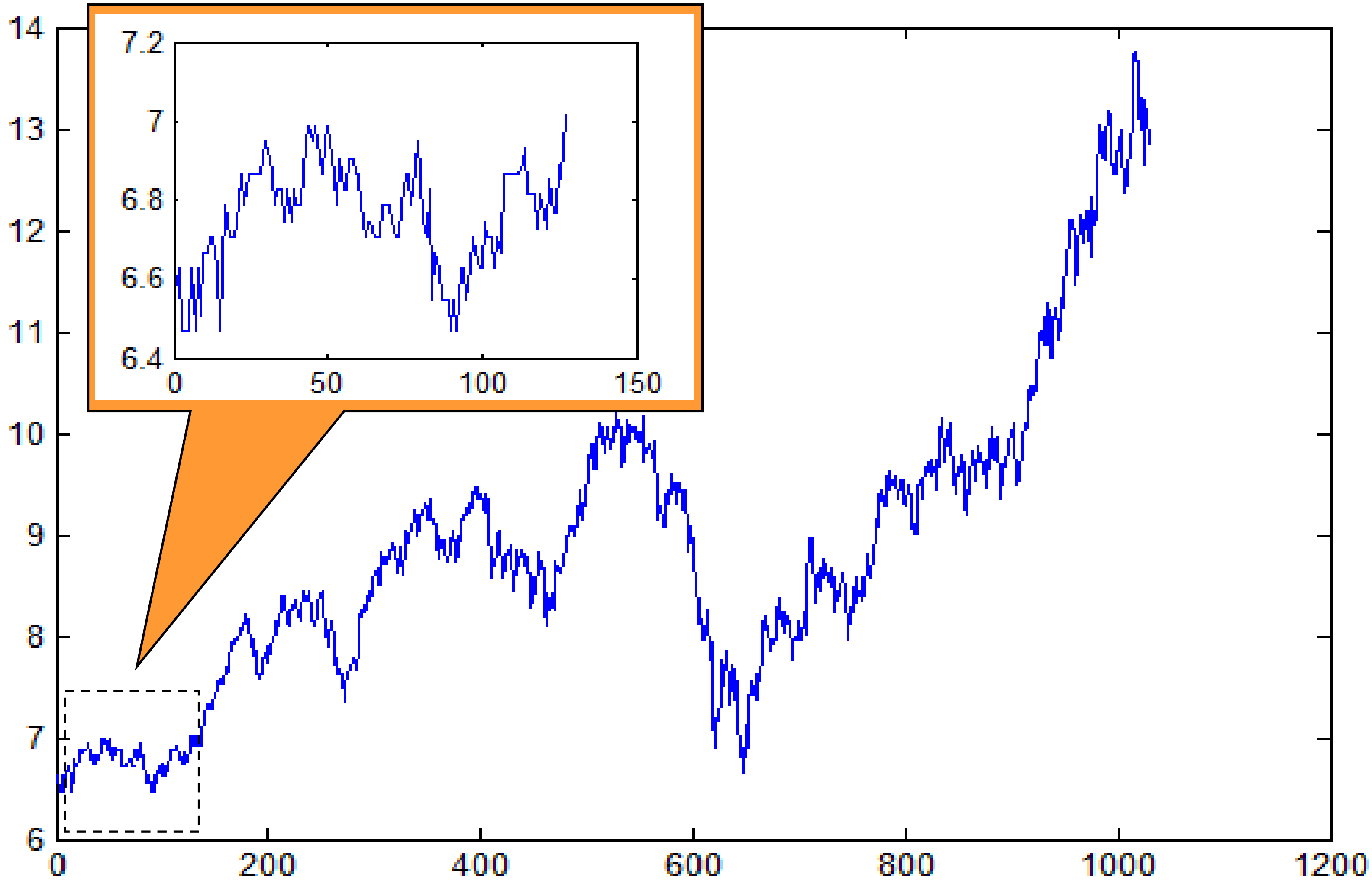
Multiscale Data Smoothing
April, 6th 2006

Orhan Karsligil

The original signal



The Multiscale Transformation

The unbalanced version

Transform:

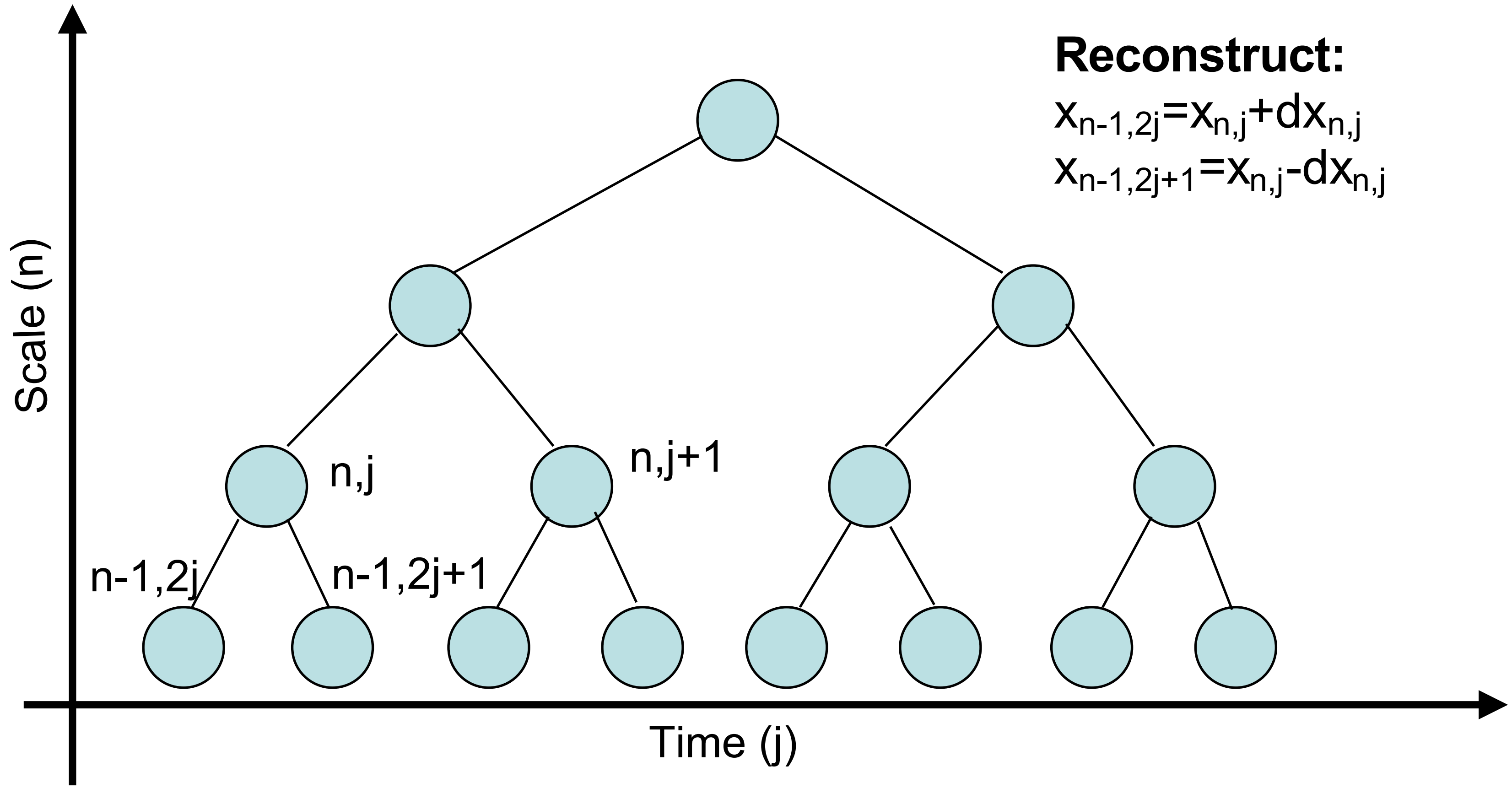
$x_{n,j}=1/2(x_{n-1,2j}+x_{n-12j+1})$

$dx_{n,j}=1/2(x_{n-1,2j}-x_{n-1,2j+1})$

Reconstruct:

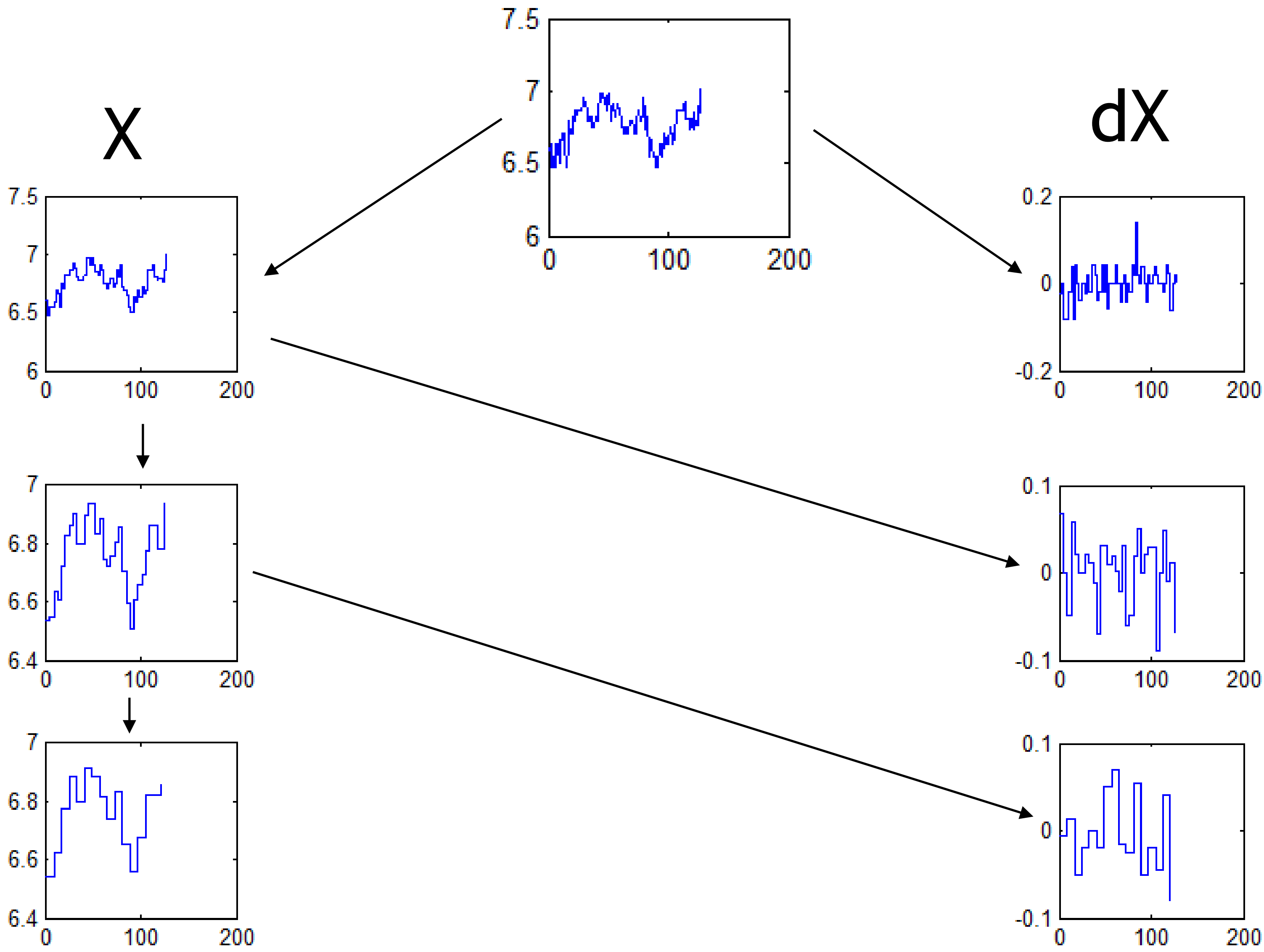
$x_{n-1,2j}=x_{n,j}+dx_{n,j}$

$x_{n-1,2j+1}=x_{n,j}-dx_{n,j}$

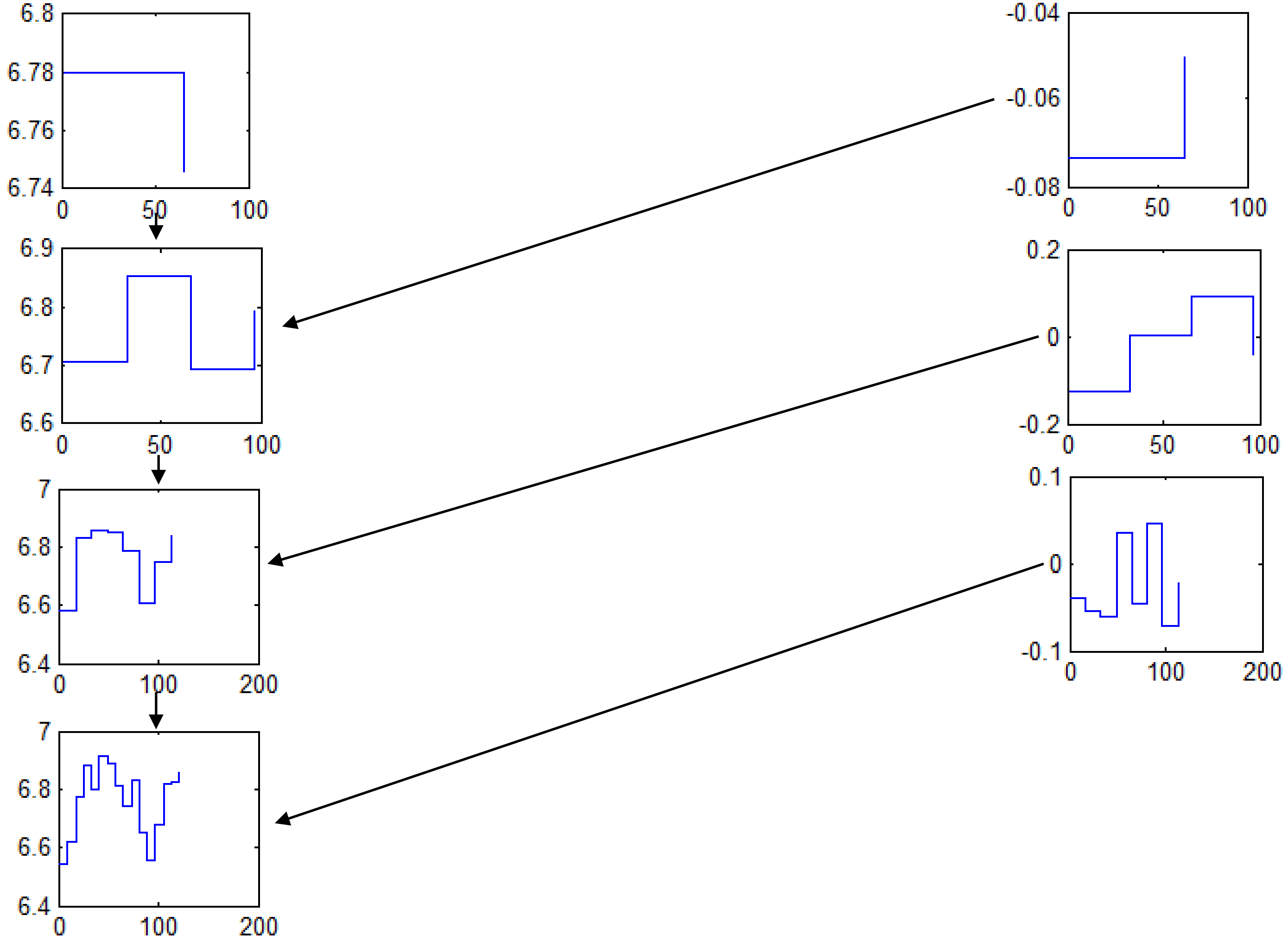


Multiscale Transformation of the Signal

The first 128 Points only



Multiscale Reconstruction



Some Important Points

- At all scales energy of the signal is preserved (the area under the xj steps is the same at every scale).
- The number of data points are halved at each scale
- Lower scales (small n) corresponds to high frequency changes and higher scales (large n) corresponds to lower frequencies.

Thresholding (I)

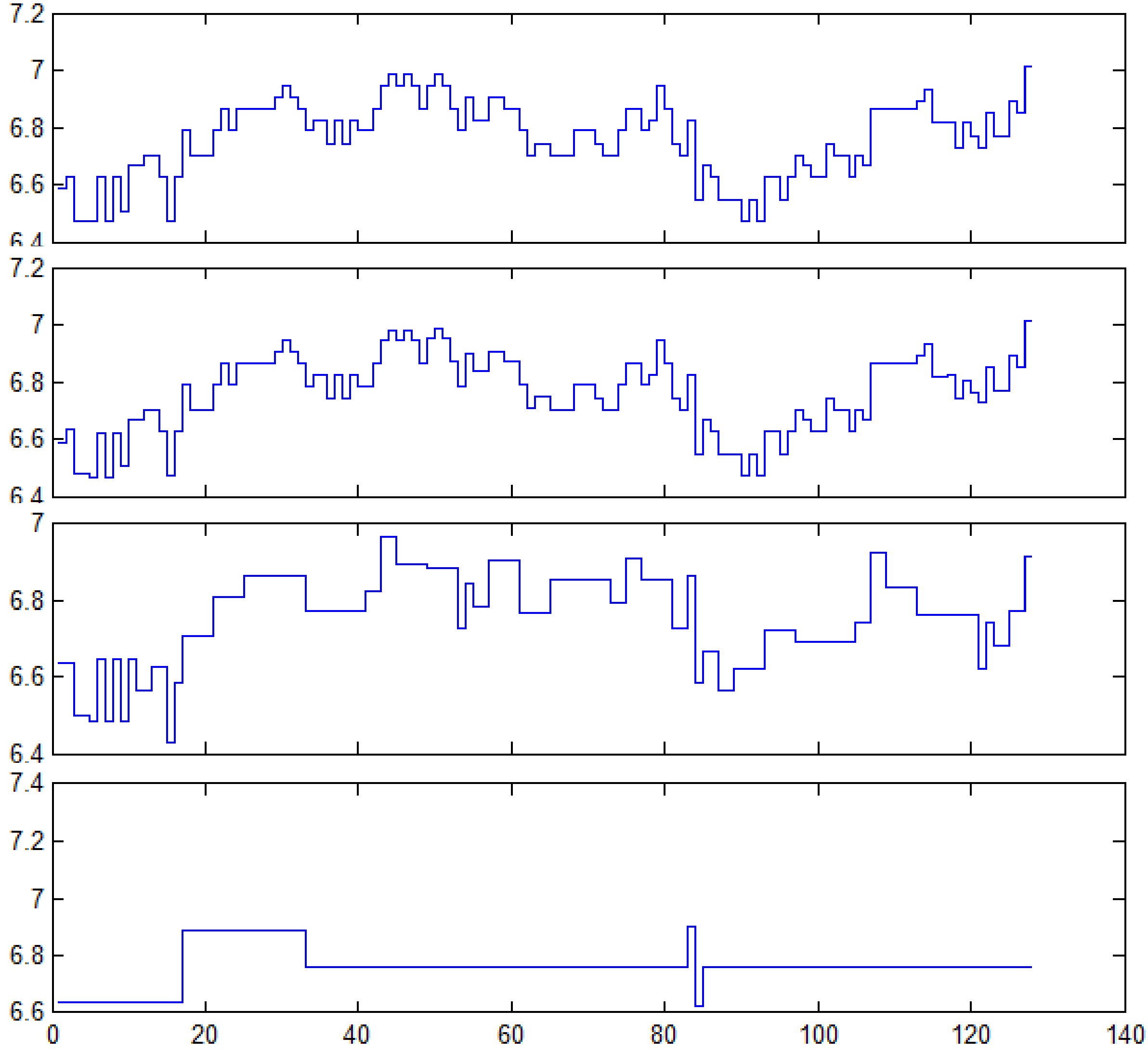
If abs(dxj) < ε then dxj=0

ε=0

ε=0.01

ε=0.05

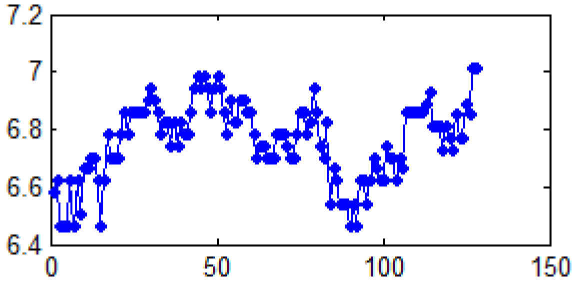
ε=0.1



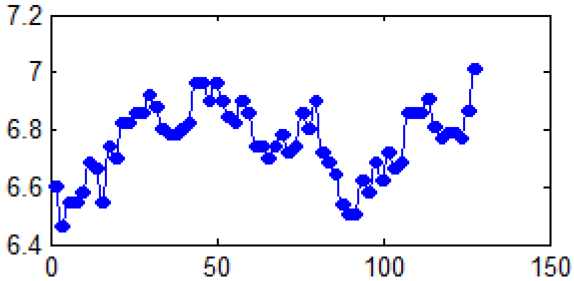
Thresholding (II)

When n < m set dXn=0

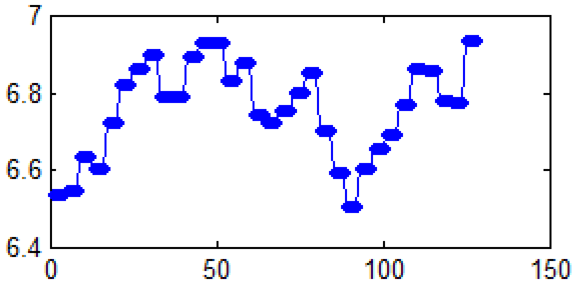
m=1



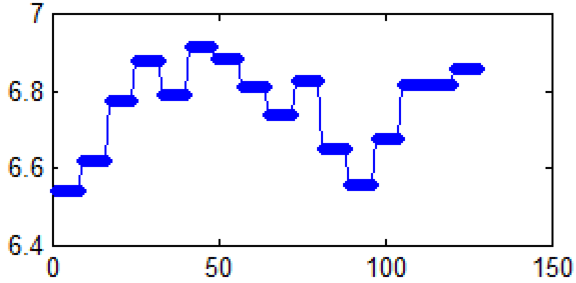
m=2



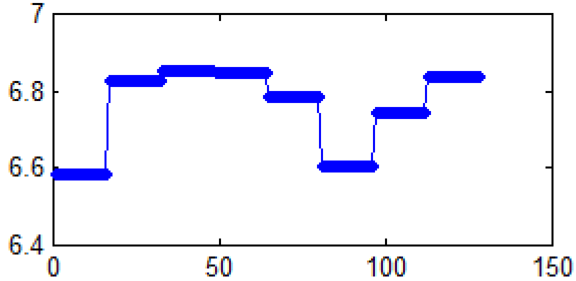
m=3



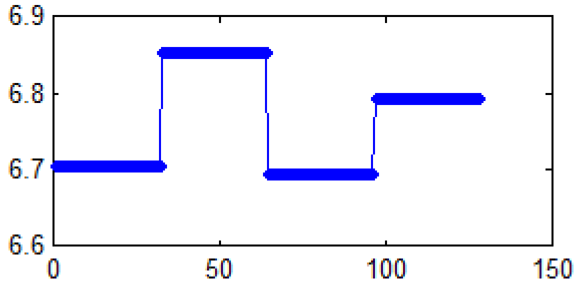
m=4



m=5



m=6



Conclusions

- Thresholding introduces artificial local jumps into the data.
- Local variations are accumulated at the edges of local zones.
- In many cases a smooth, continuous function with well behaved first and second derivatives is required.

Multiscale Smoothing

- Take any of the thresholding (II) results.
- Formulate the following optimization problem

$$\min x'Qx$$

subject to

$$Ax=b$$

where Q=R'R and

$$R=\begin{matrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & -1 \end{matrix}$$

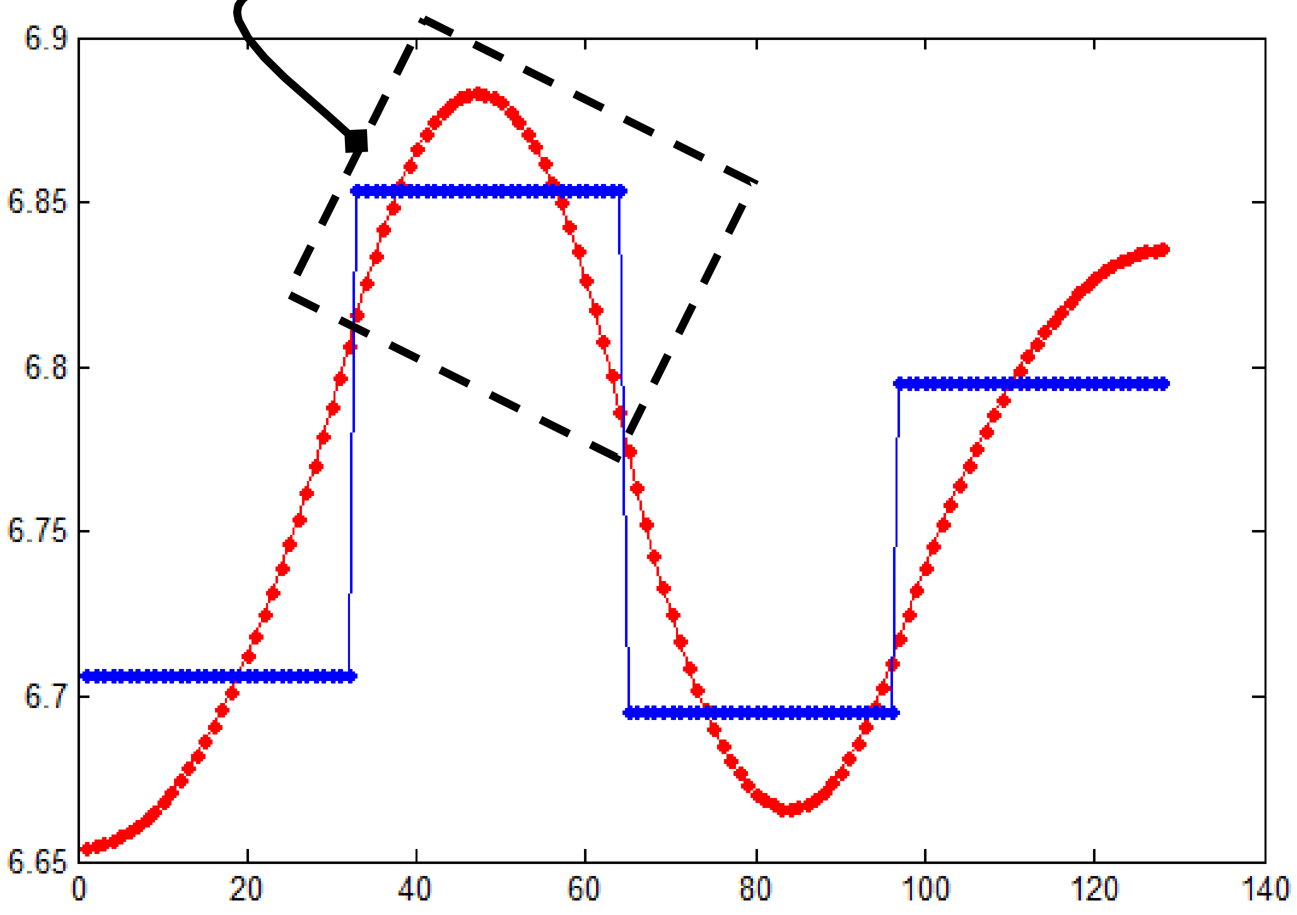
A is the local average matrix and b is the local average values

Multiscale Smoothing

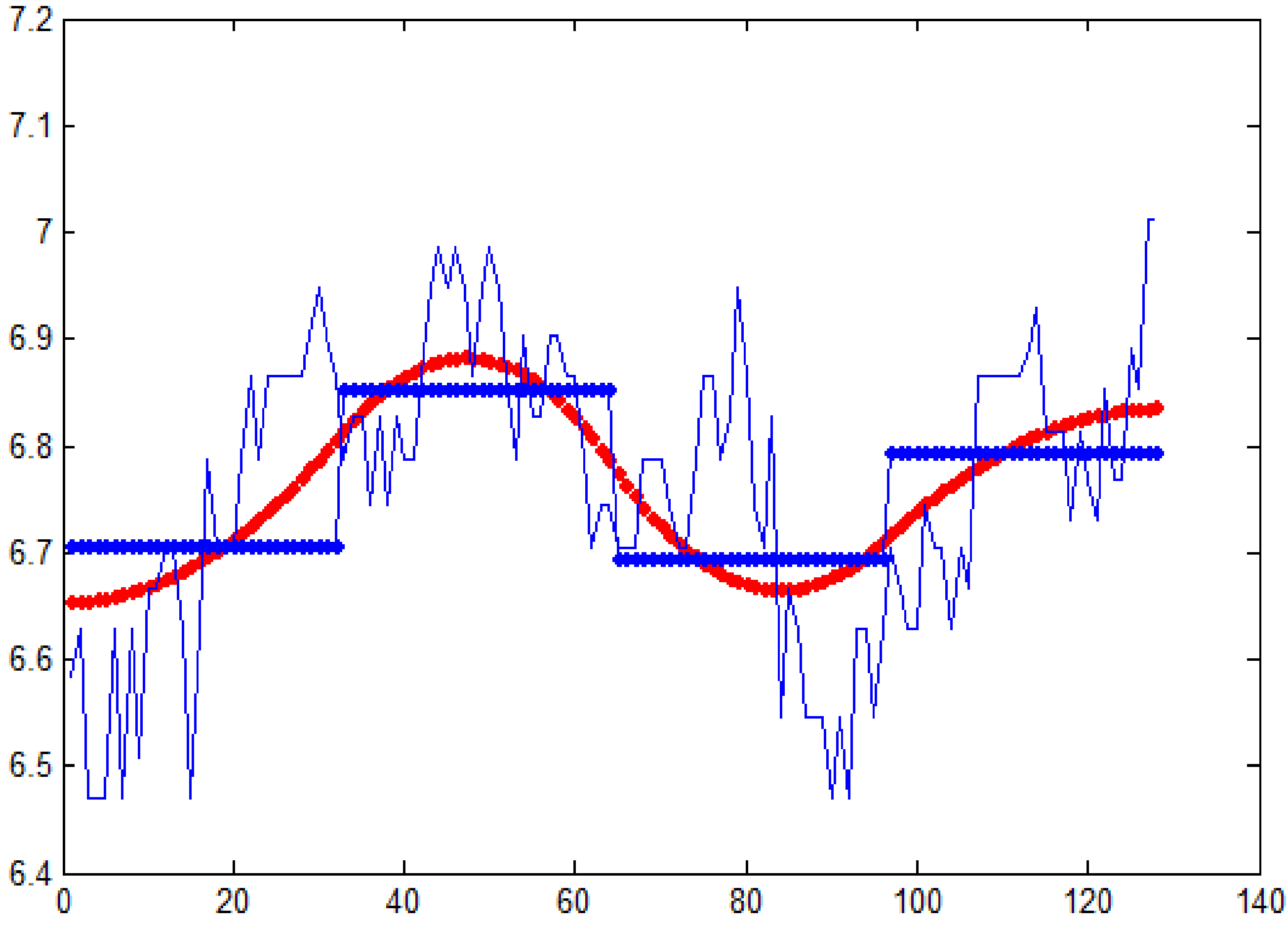
- The objective is to minimize the distance between neighbouring points while preserving the local averages.

Results

Sum of the values of blue dots
is equal to the red dots in this box

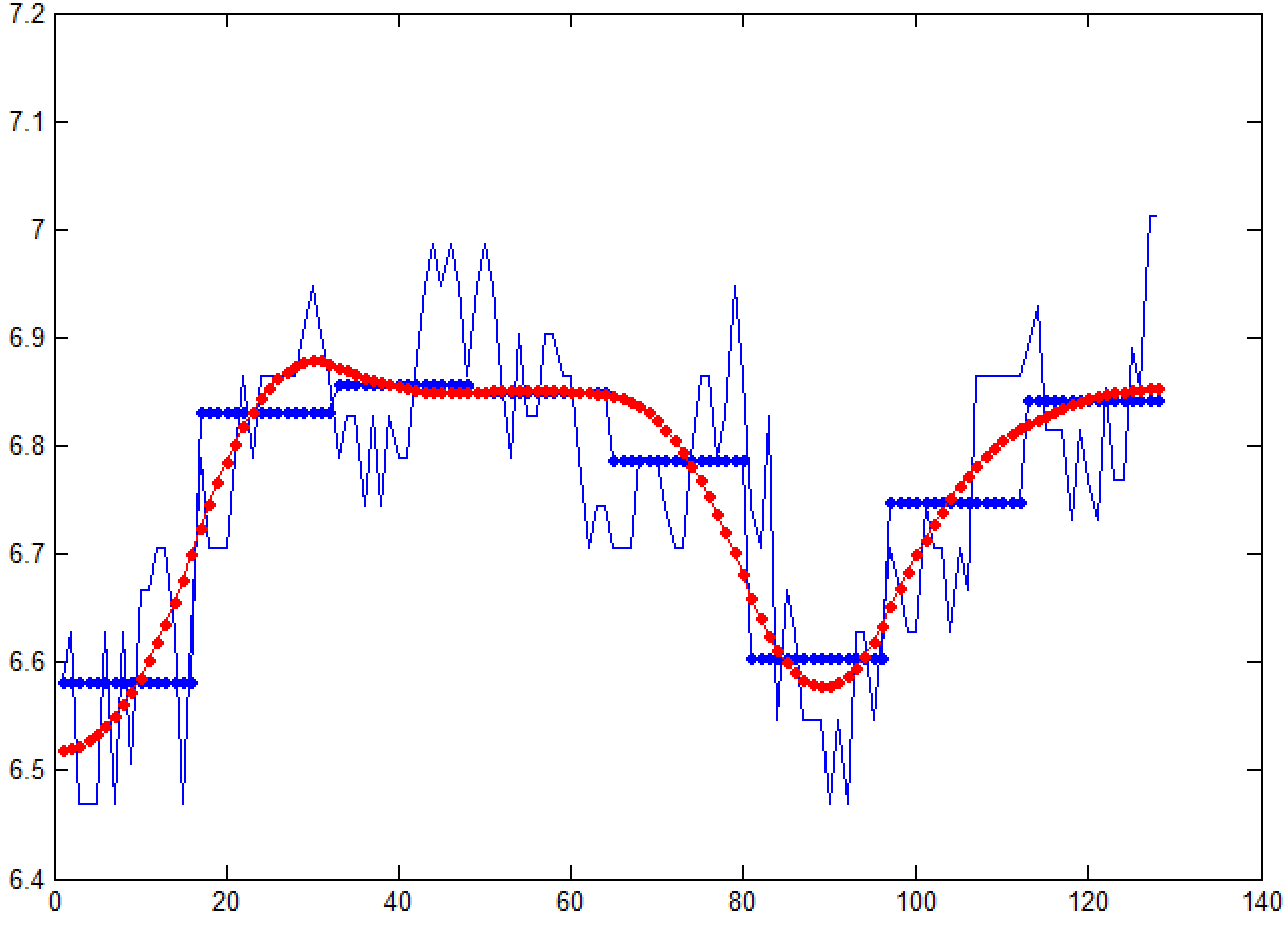


Results (with the original signal)

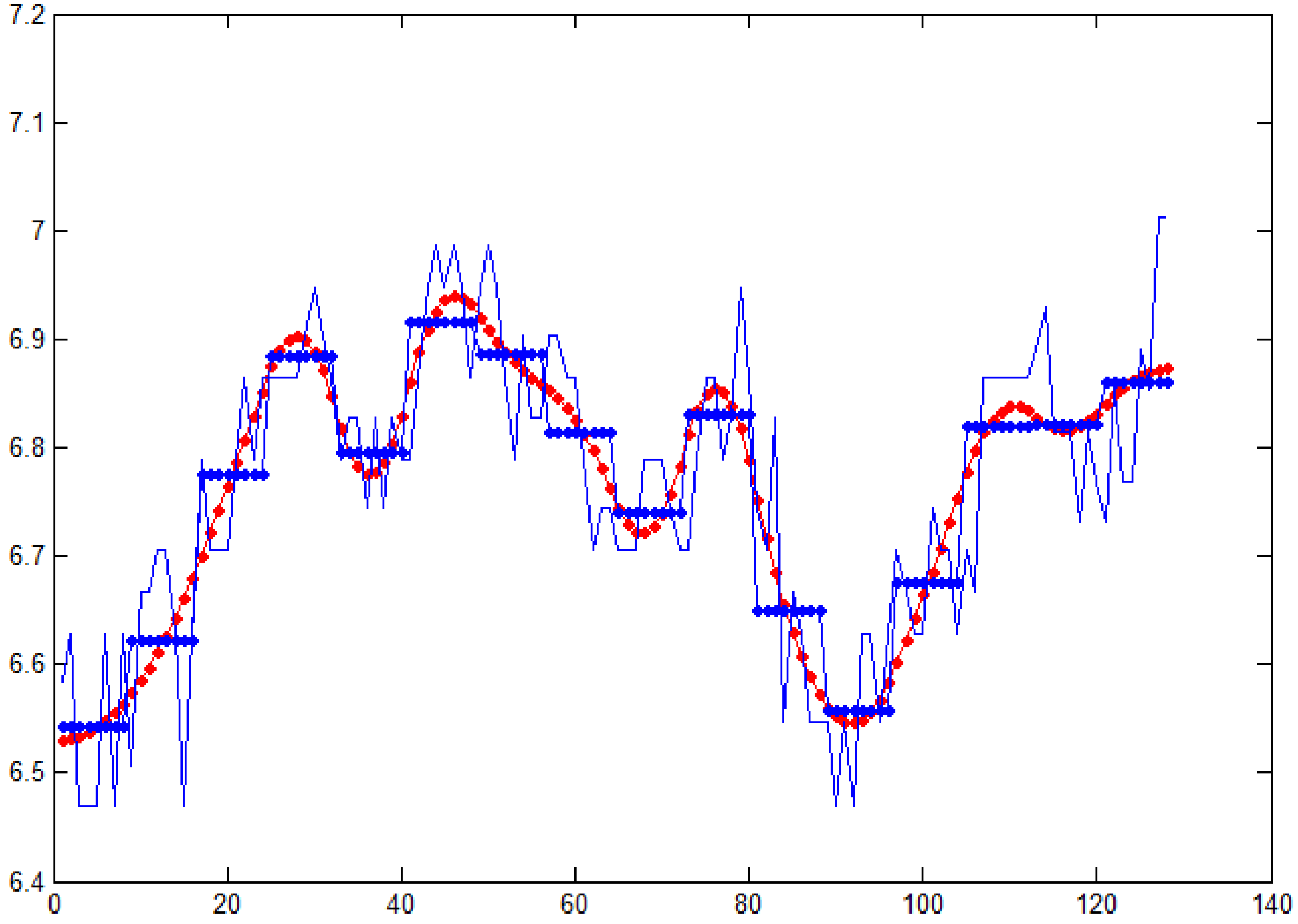


Keep in mind that this is a very low resolution (scale 5) approximation

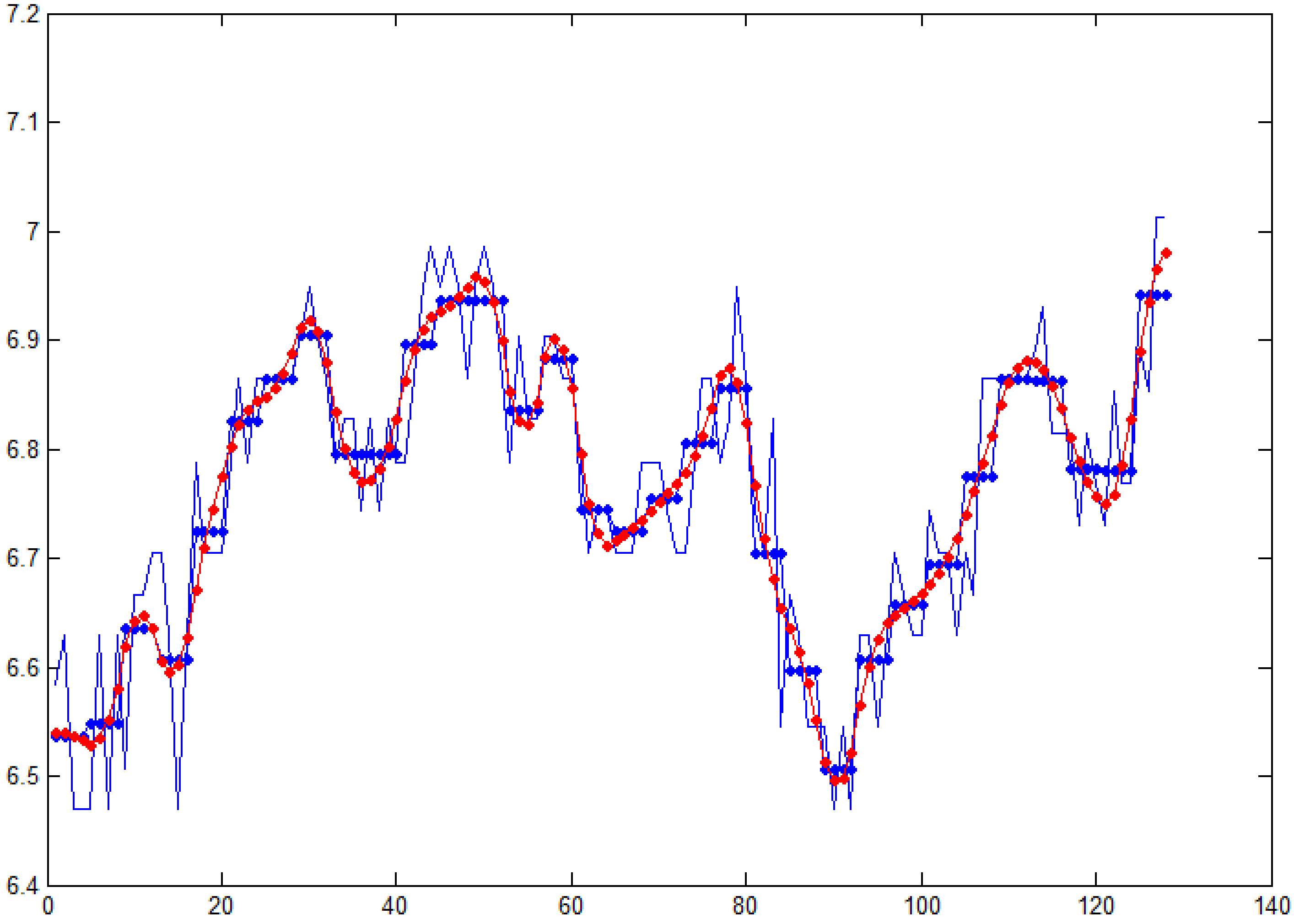
Results (scale 4)



Results (scale 3)

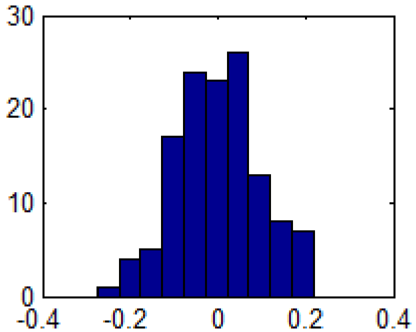


Results (scale 2)

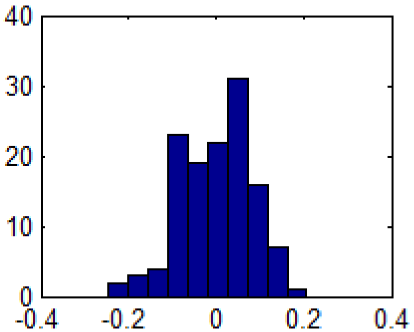


Some statistics

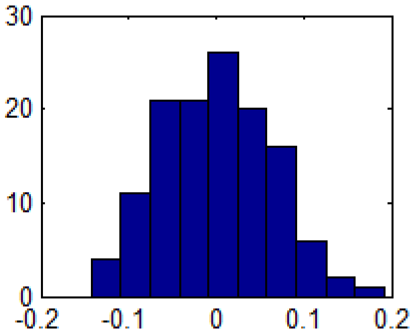
Distribution of the approximation error of smoothing process



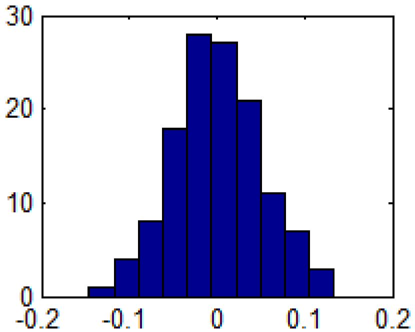
scale 5



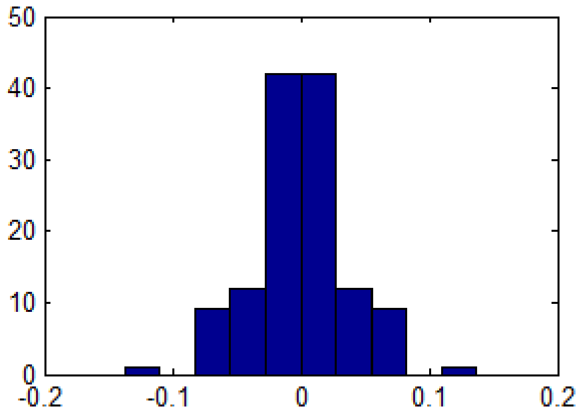
scale 4



scale 3

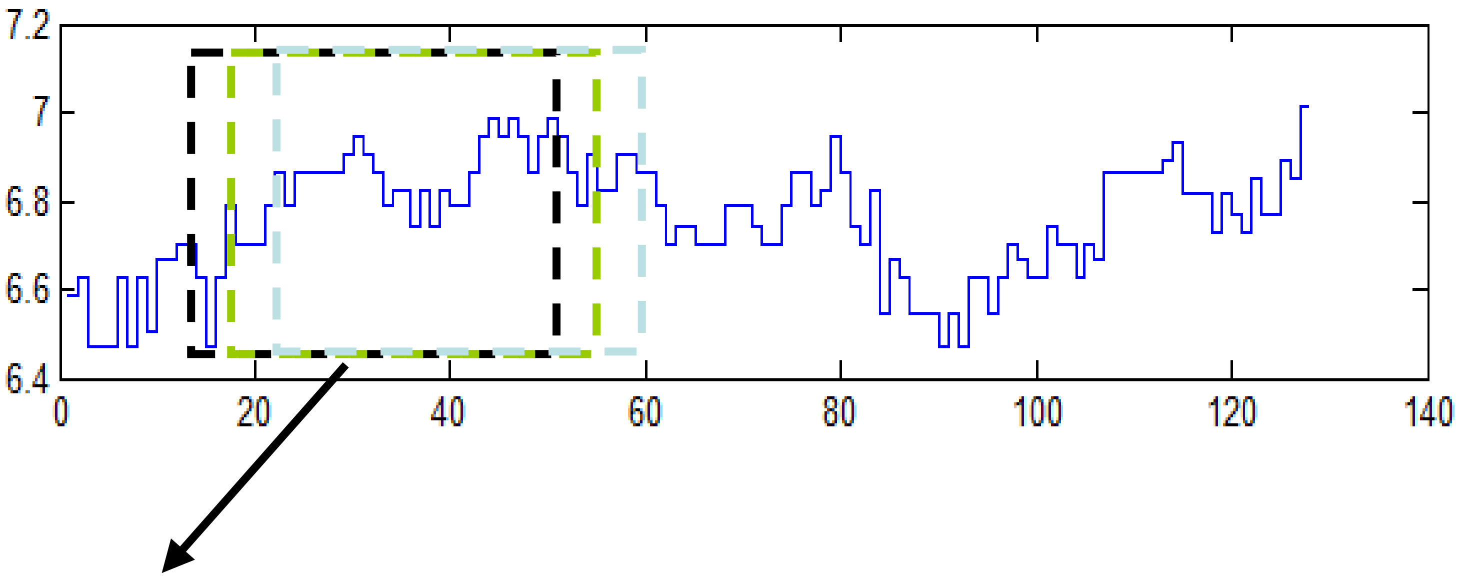


scale 2



scale 1

Moving window smoothing



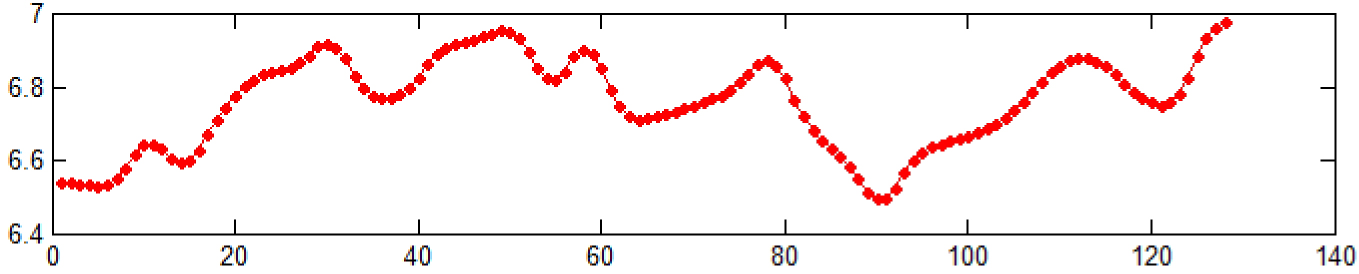
For each of these windows calculate the smoothing results and see if they overlap and how previous points are affected by the new points

Moving window smoothing

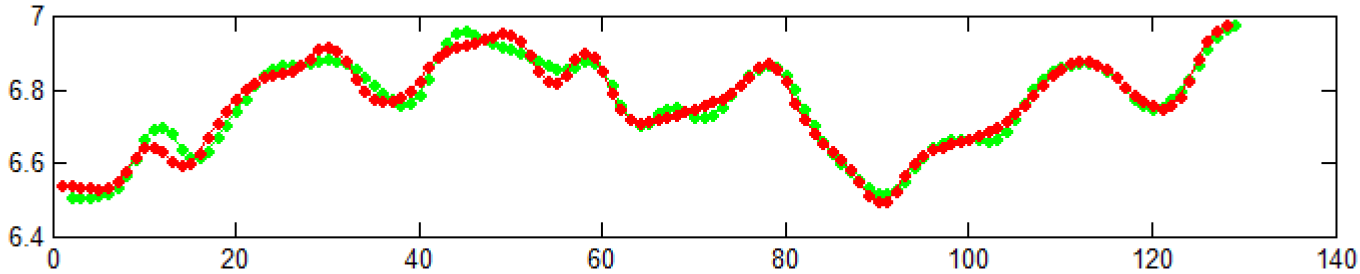
- Use scale 2 smoothing for most detail.
- Use [1:128], [2:129], [3:130], [4:131] etc
- Then compare all m windows in the [2+m:128+m] range

Mowing Window Smoothing

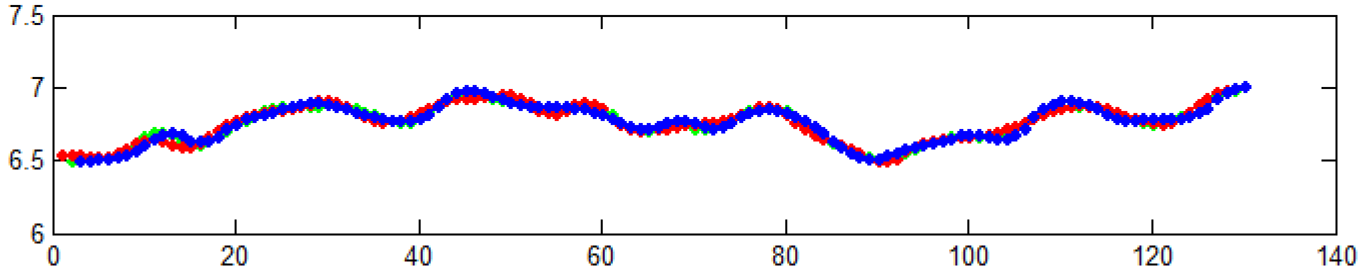
1:128



1:128 and
2:129



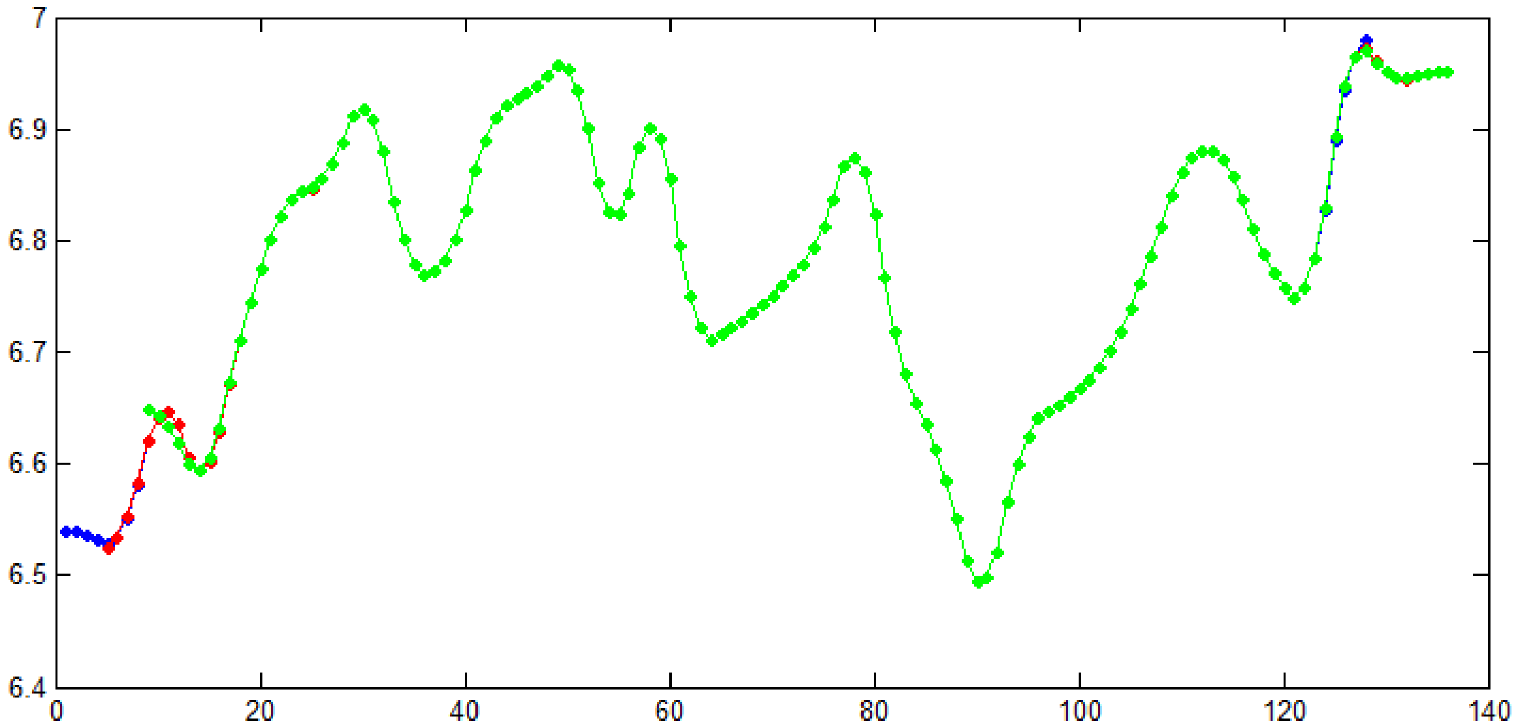
1:128 and
2:129 and
3:130



Mowing Window Smoothing

- As one can see new points change the smoothing results of previous points drastically in some cases.
- This happens since the averages change.
- If this is done in blocks this would not be an issue: [1:128],[5:132],[9:136] etc..

Mowing Windows Smoothing (blocks)



Blue: 1:128, Red 5:132, Green 9:136

Full overlap within the 9:128 range and some edge effects

Mowing Window Smoothing Results

- New points affect past points because of the global effect of the optimization.
- When blocks corresponding to the scale is used (scale 2=4 points, scale 3=8 points, scale m=2^m points) then this effect disappears because local averages remain the same in each window.