Computational Photography

Prof. Feng Liu

Spring 2022

http://www.cs.pdx.edu/~fliu/courses/cs510/

04/07/2022

Last Time

- Filters and its applications

Today

- De-noise
  - Median filter
  - Bilateral filter
  - Non-local mean filter
  - Video de-noising
- Quality metrics

Filter Re-cap

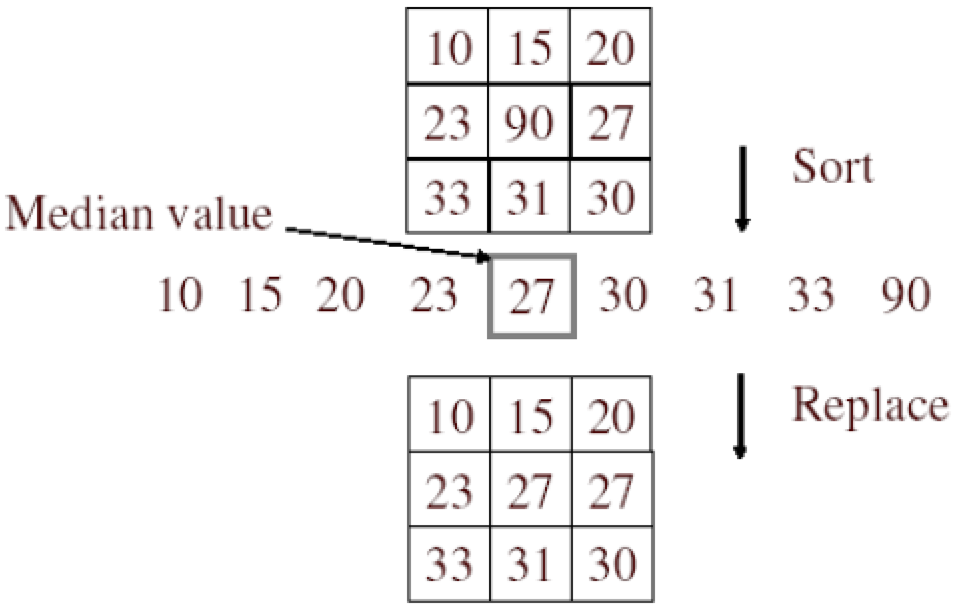

noisy image


naïve denoising
Gaussian blur


better denoising
edge-preserving filter

Slide credit: Sylvain Paris and Frédo Durand

Median Filter

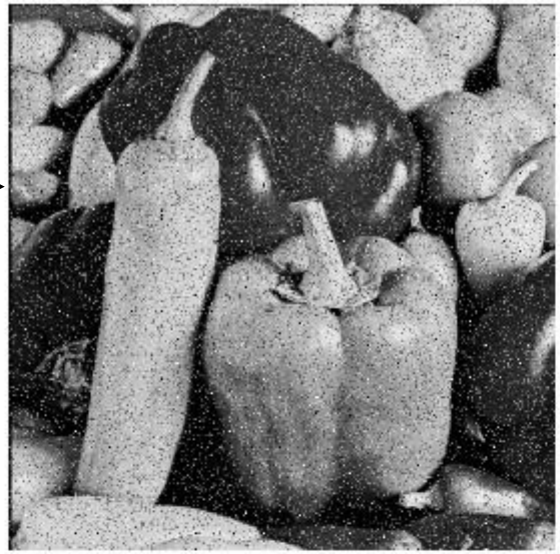


- Replace pixel by the median value of its neighbors
- No new pixel values introduced
- Removes spikes: good for impulse, salt & pepper noise

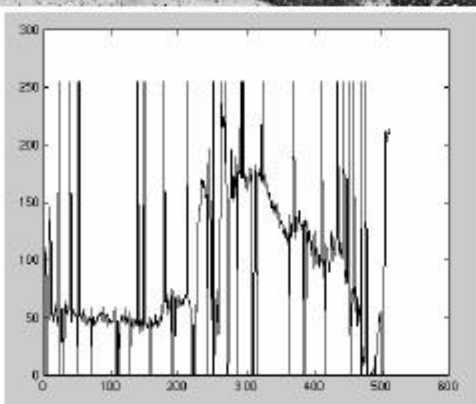
Slide credit: C. Dyer

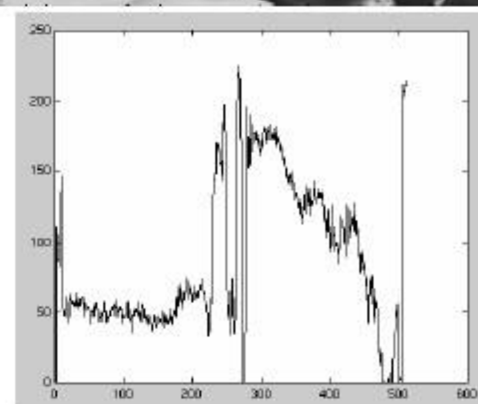
Median Filter

Salt and pepper noise



Median filtered





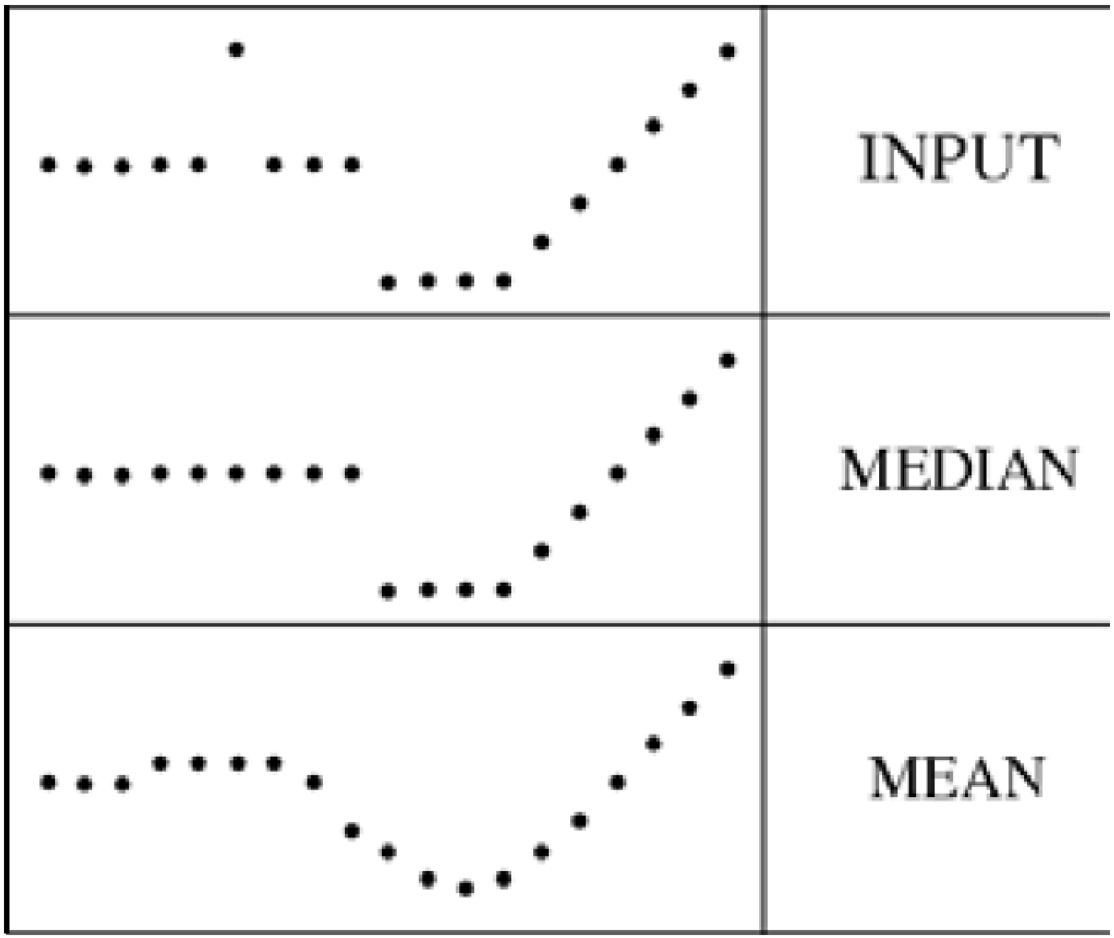
Plots of a row of the image

Matlab: output im = medfilt2(im, [h w])

Slide credit: M. Hebert, C. Dyer

Median Filter

- Median filter is edge preserving



Slide credit: C. Dyer

original image

1px median filter

3px median filter

10px median filter

Slide credit: C. Dyer

19x19 median filter

input output

Slide credit: C. Dyer

images by J. Plush

Bilateral filter

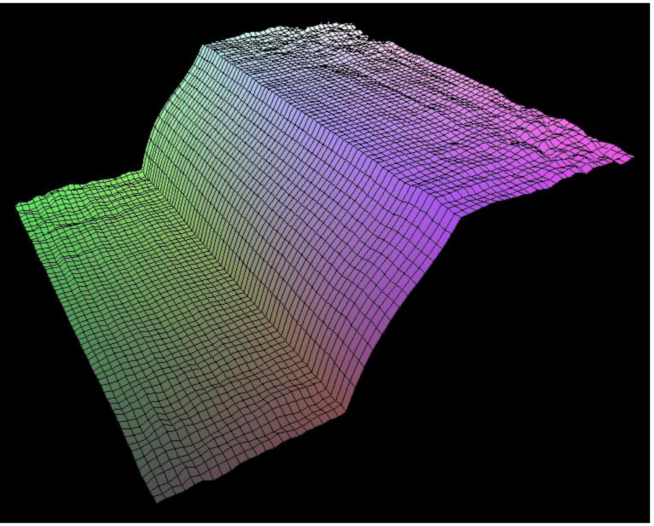
- Tomasi and Manduci 1998 http://www.cse.ucsc.edu/~manduchi/Papers/ICCV98.pdf
- Related to
  - SUSAN filter [Smith and Brady 95] http://citeseer.ist.psu.edu/smith95susan.html
  - Digital-TV [Chan, Osher and Chen 2001] http://citeseer.ist.psu.edu/chan01digital.html
  - sigma filter http://www.geogr.ku.dk/CHIPS/Manual/f187.htm

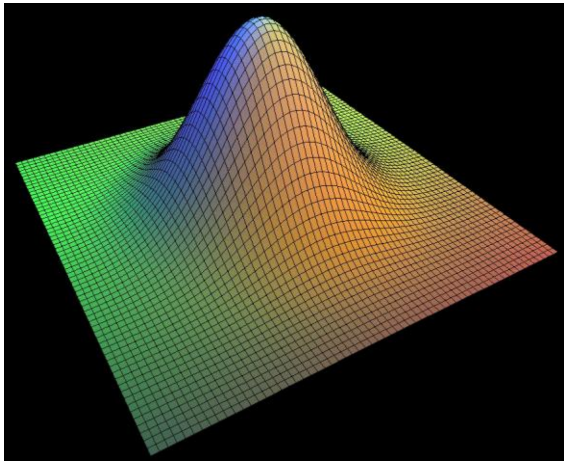
Slide credit: F. Durand

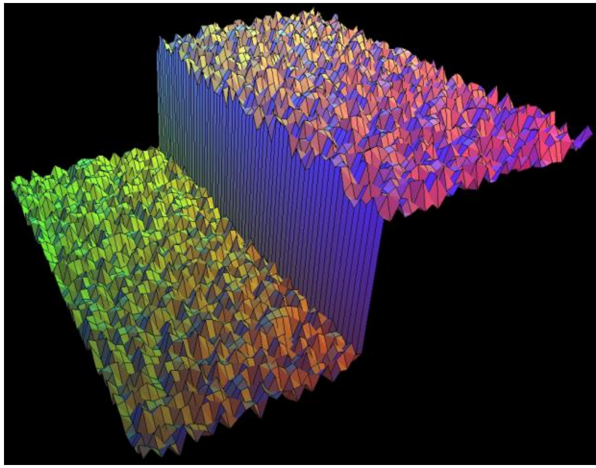
Start with Gaussian filtering

- Here, input is a step function + noise

$$J \quad = \quad f \quad \otimes \quad I$$







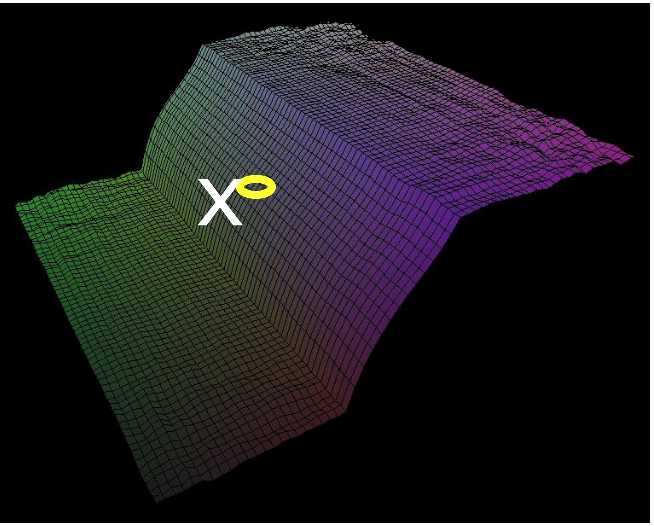
output ⟵ input

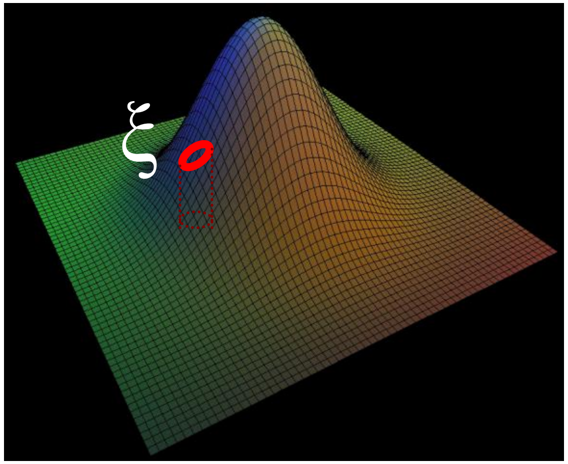
Slide credit: F. Durand

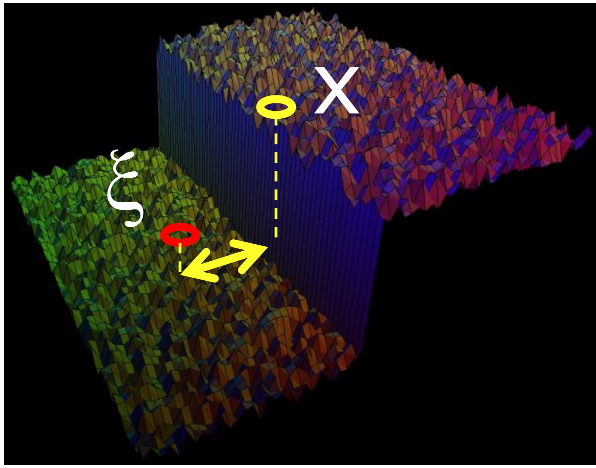
Gaussian filter as weighted average

- Weight of ξ depends on distance to x

$$J(x) = \sum_{\xi} \quad f(x,\xi) \quad I(\xi)$$







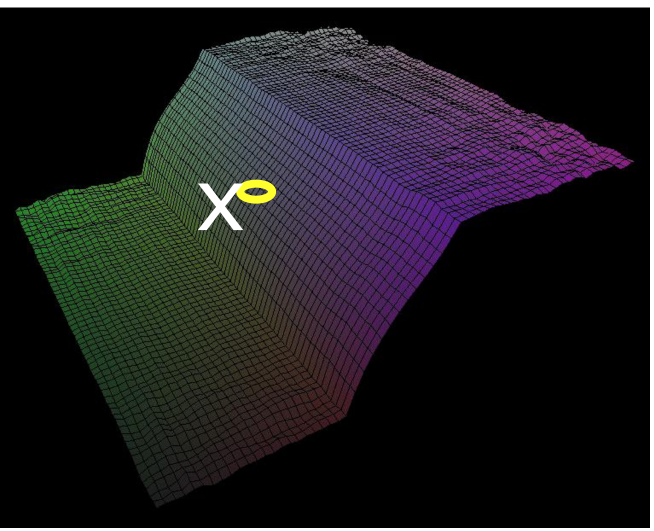
output ← input

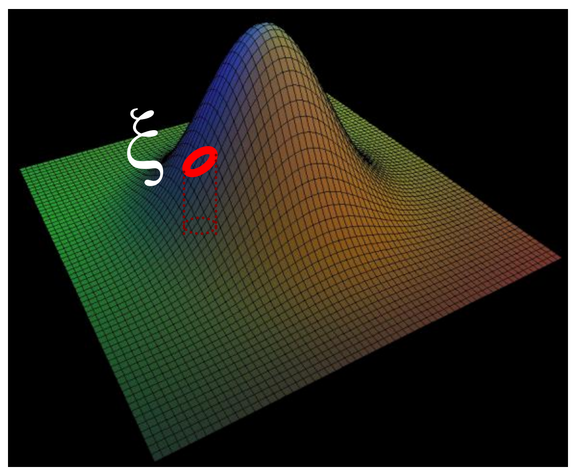
Slide credit: F. Durand

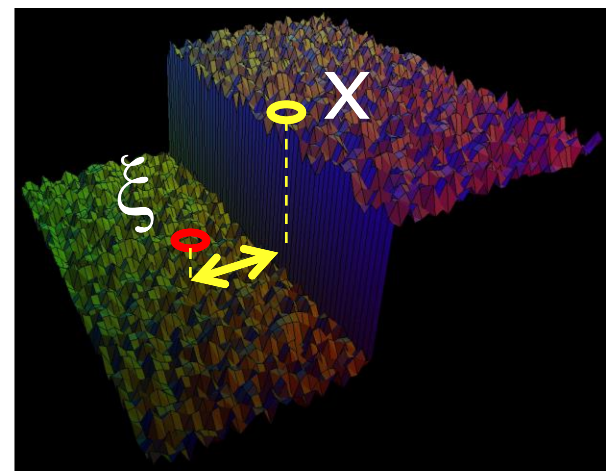
The problem of edges

- Here, $I(\xi)$ "pollutes" our estimate J(x)
- It is too different

$$J(x) = \sum_{\xi} f(x,\xi) \quad I(\xi)$$







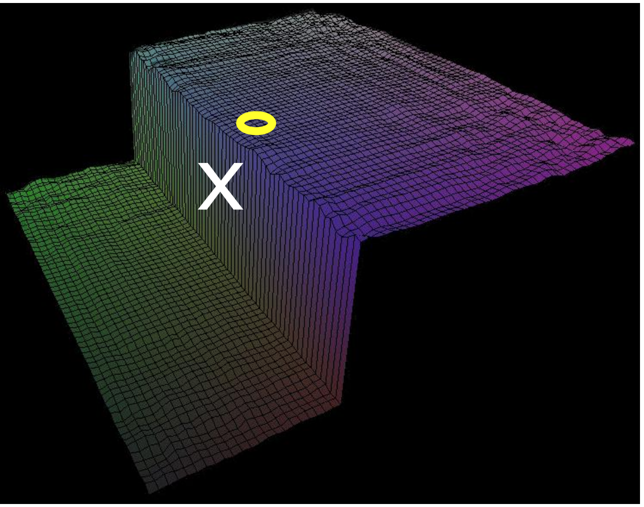
output $\longleftarrow$ input

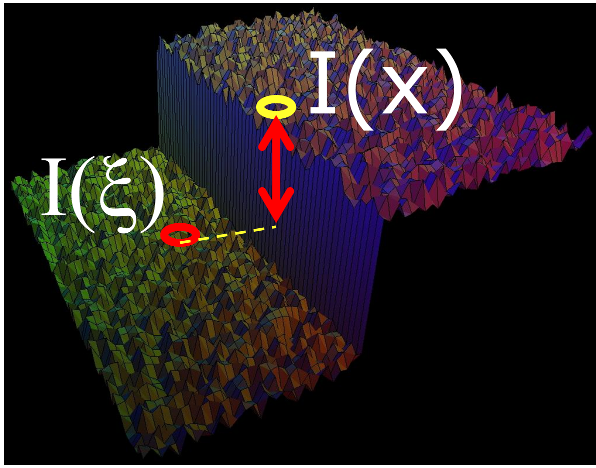
Slide credit: F. Durand

Principle of Bilateral filtering [Tomasi & Manduchi '98]

- Penalty g on the intensity difference

$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x,\xi) \quad g(I(\xi) - I(x)) \quad I(\xi)$$



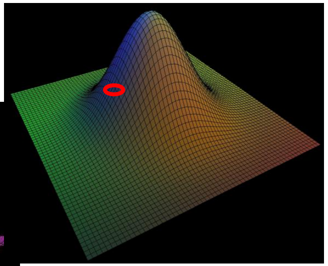


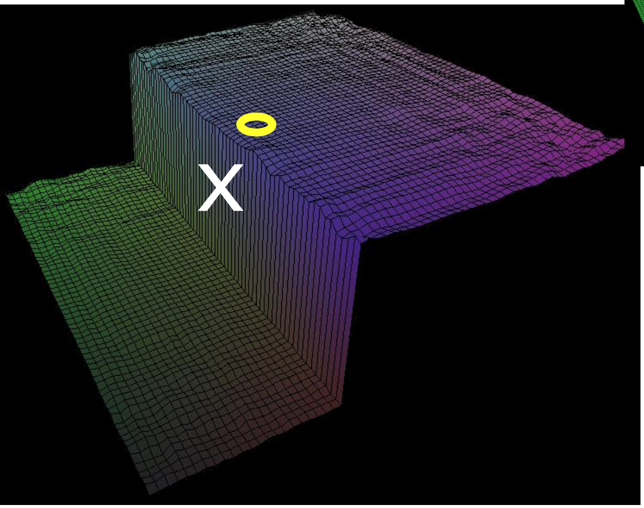
Slide credit: F. Durand

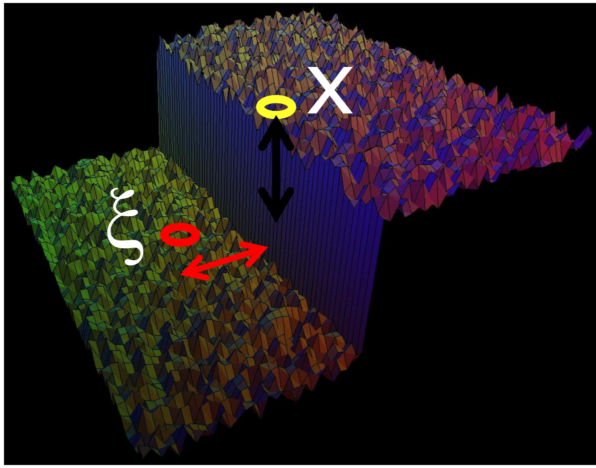
Bilateral filtering [Tomasi and Manduchi 1998]

- Spatial Gaussian f

$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x,\xi)\, g(I(\xi) - I(x)) \qquad I(\xi)$$







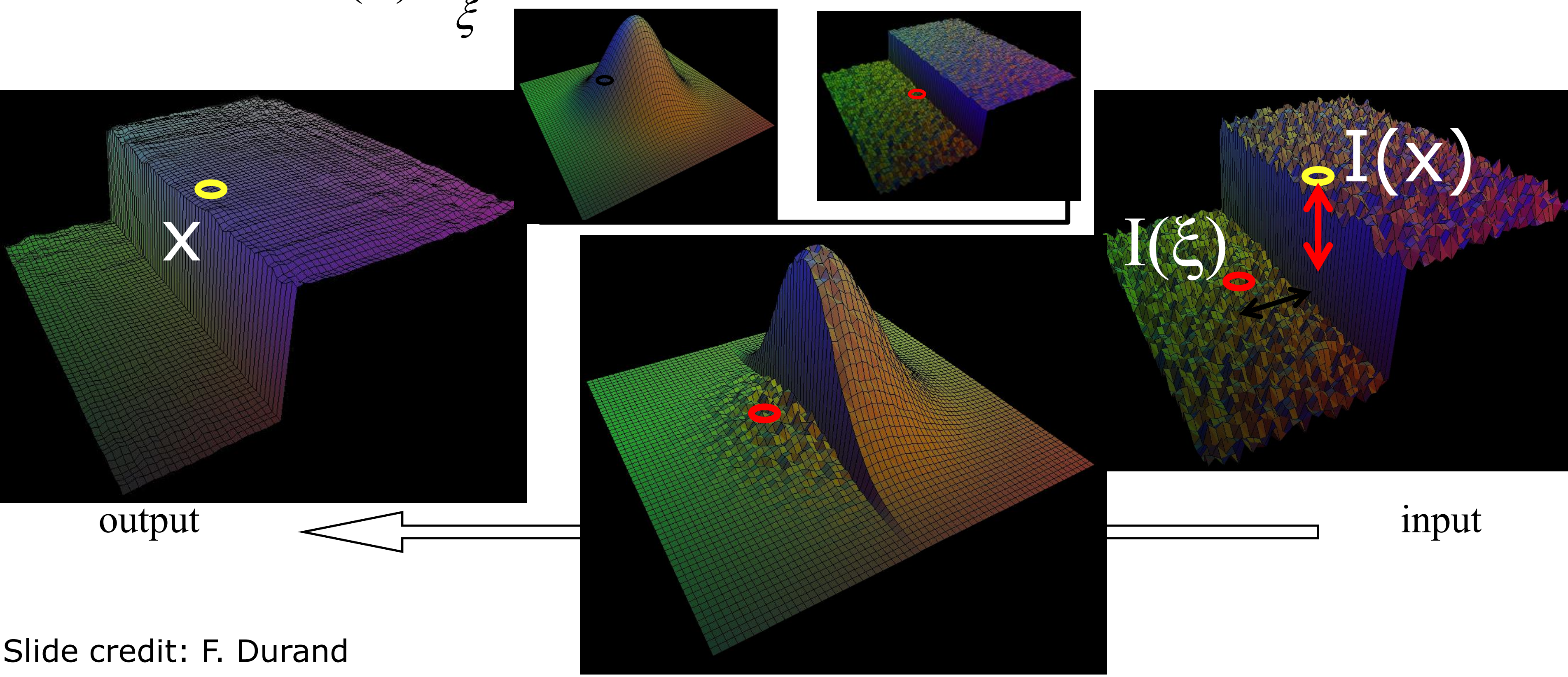
output input

Slide credit: F. Durand

Bilateral filtering [Tomasi and Manduchi 1998]

- Spatial Gaussian f
- Gaussian g on the intensity difference

$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x,\xi)\, g(I(\xi) - I(x))\, I(\xi)$$

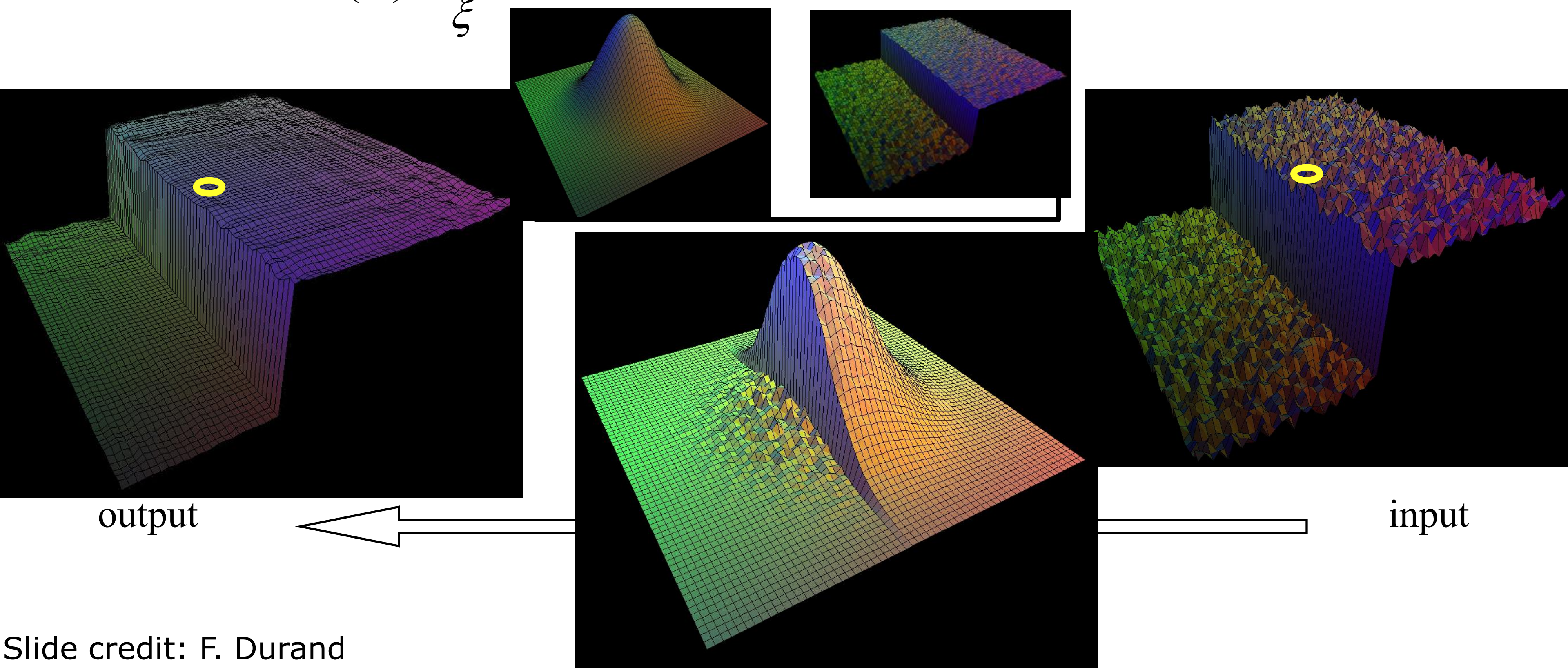


Slide credit: F. Durand

Normalization factor [Tomasi and Manduchi 1998]

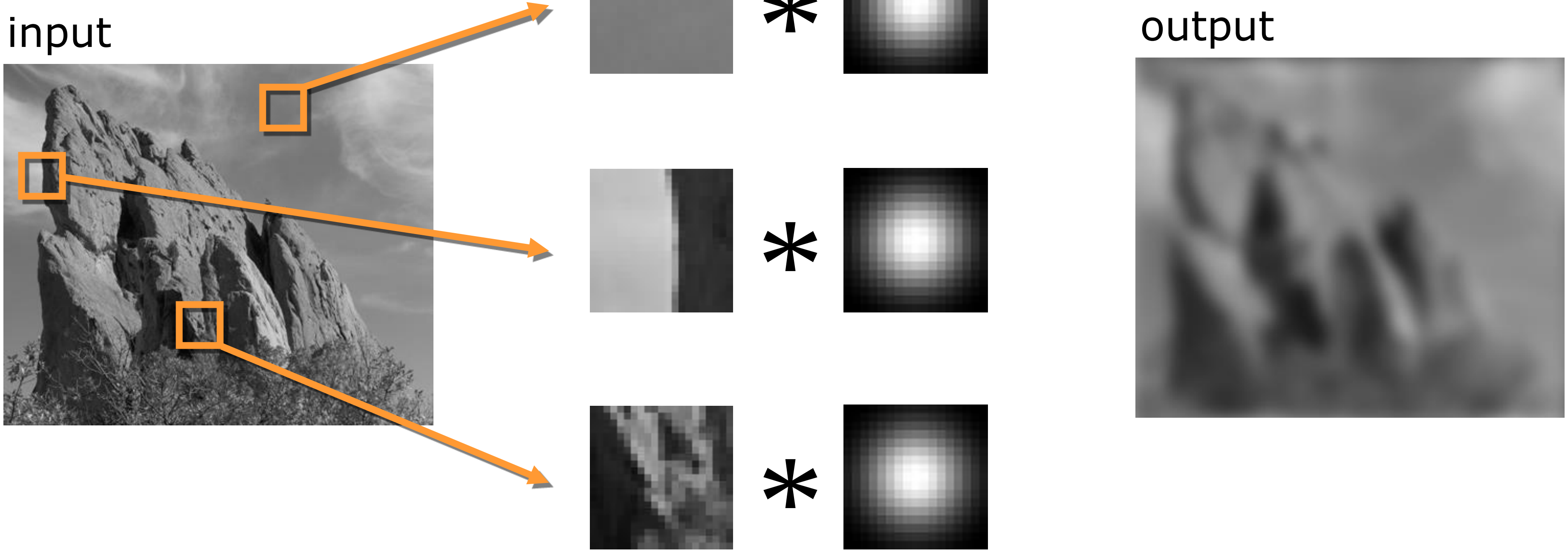
- k(x)= $\sum_{\xi} f(x,\xi) \quad g(I(\xi)-I(x))$

$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x,\xi) \quad g(I(\xi)-I(x)) \quad I(\xi)$$



Slide credit: F. Durand

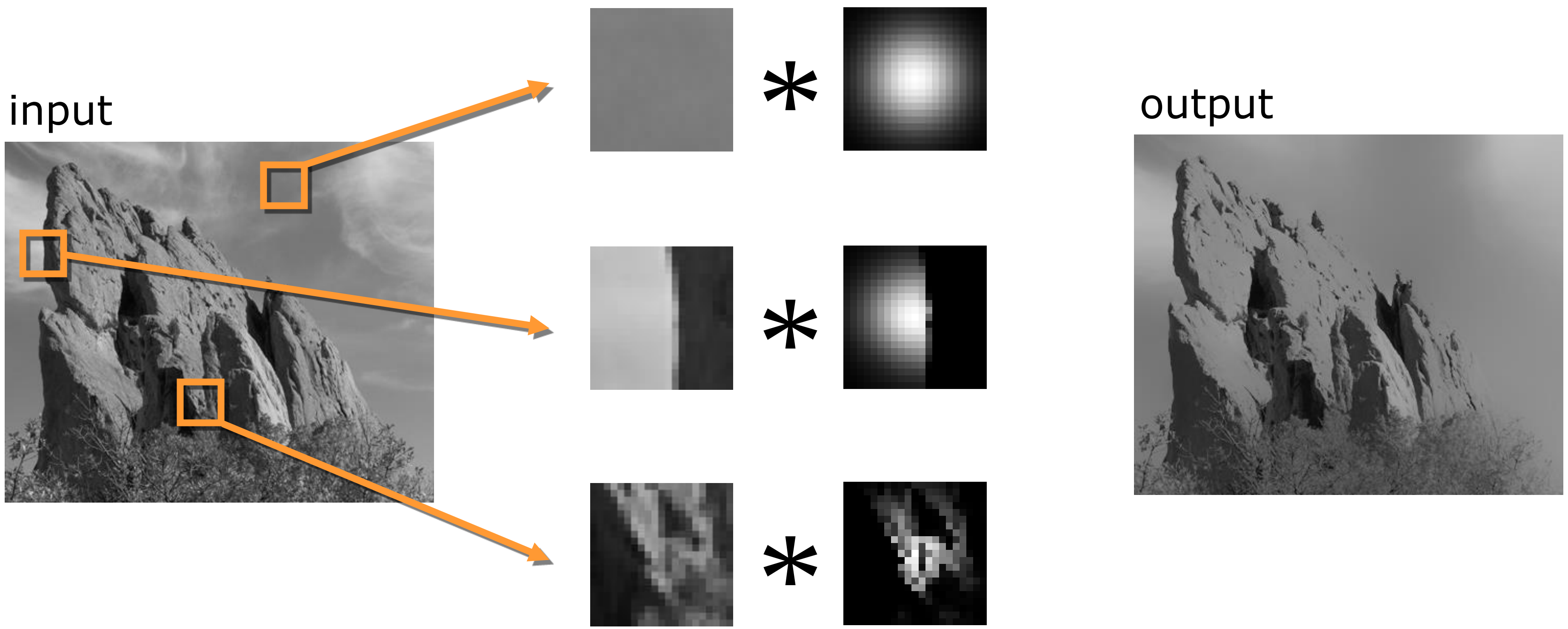
Blur from averaging across edges



Same Gaussian kernel everywhere.

Slide credit: P. Sylvain

Bilateral filter: no averaging across edges



The kernel shape depends on the image content.

Slide credit: P. Sylvain

input

Parameter for intensity difference Gaussian g

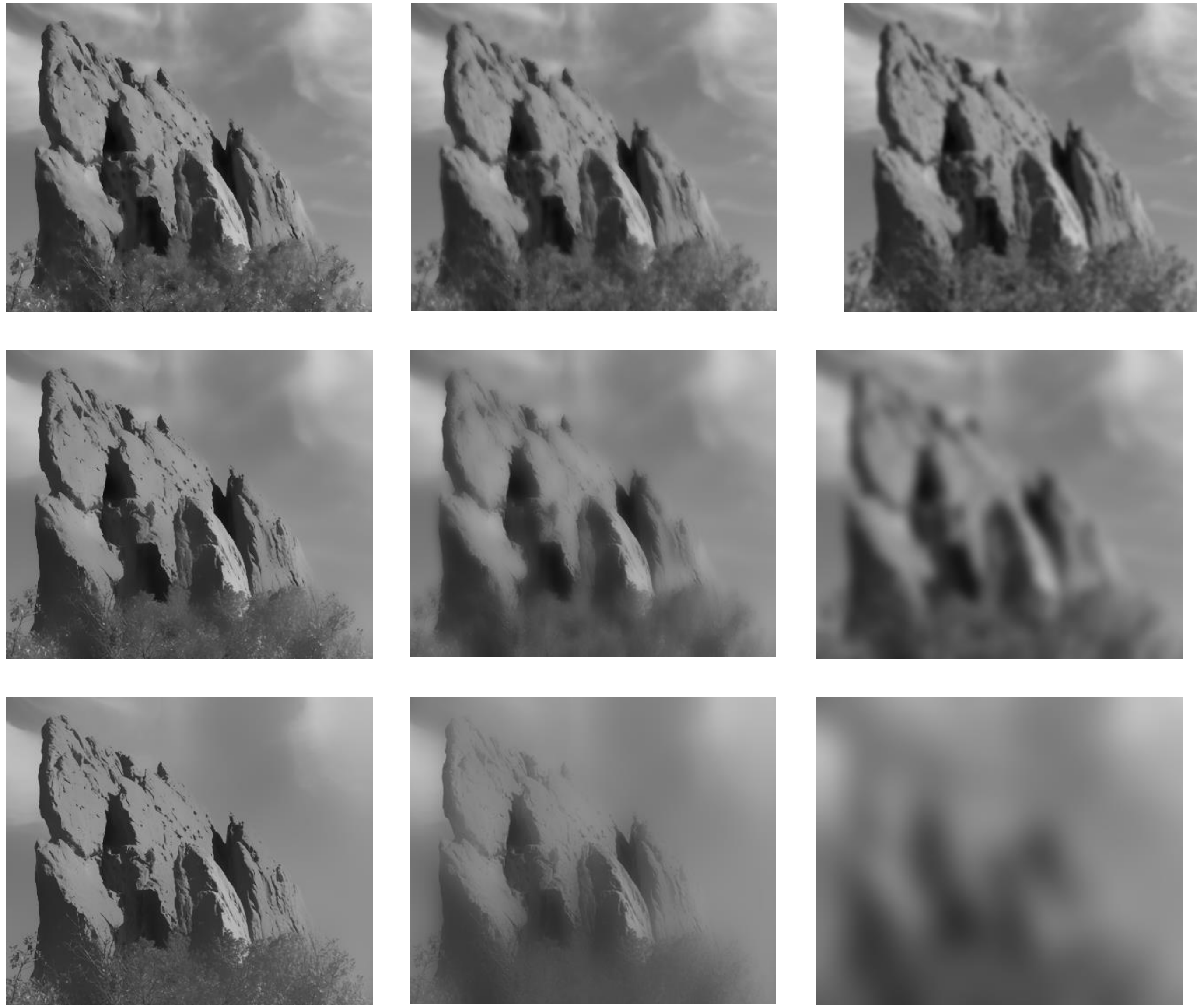
$\sigma_r = 0.1$ | $\sigma_r = 0.25$ | $\sigma_r = \infty$ (Gaussian blur)

Parameter for spatial distance Gaussian f

$\sigma_s = 2$

$\sigma_s = 6$

$\sigma_s = 18$



Slide credit: P. Sylvain

input

Parameter for intensity difference Gaussian *g*

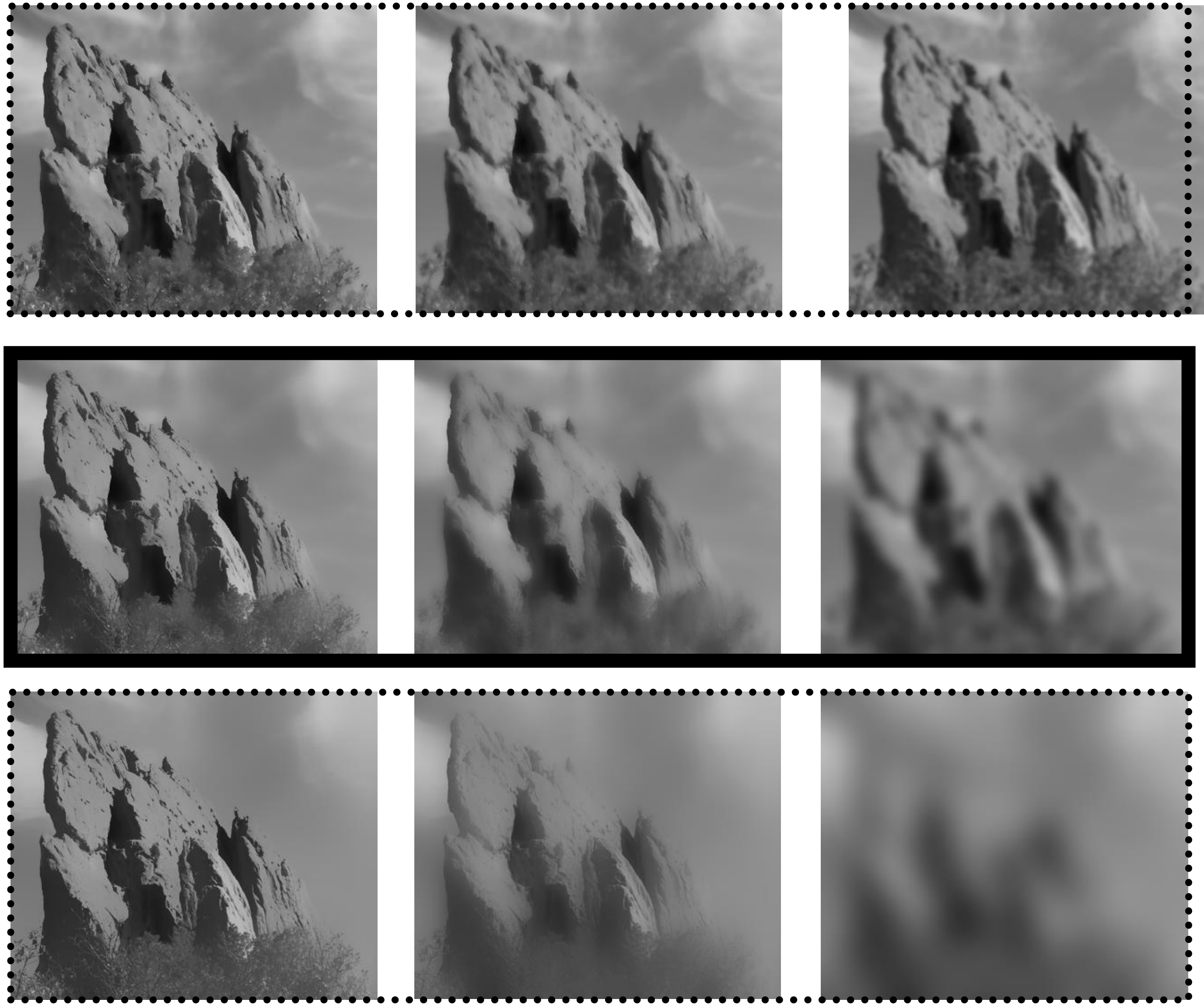
$\sigma_r = 0.1$ | $\sigma_r = 0.25$ | $\sigma_r = \infty$ (Gaussian blur)

Parameter for spatial distance Gaussian *f*

$\sigma_s = 2$

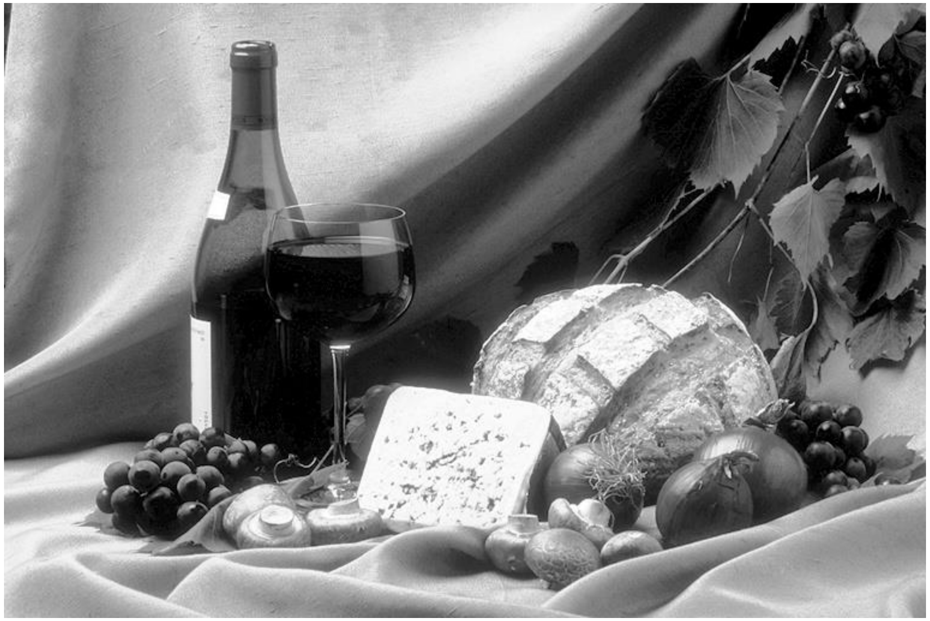
$\sigma_s = 6$

$\sigma_s = 18$



Slide credit: P. Sylvain

Result



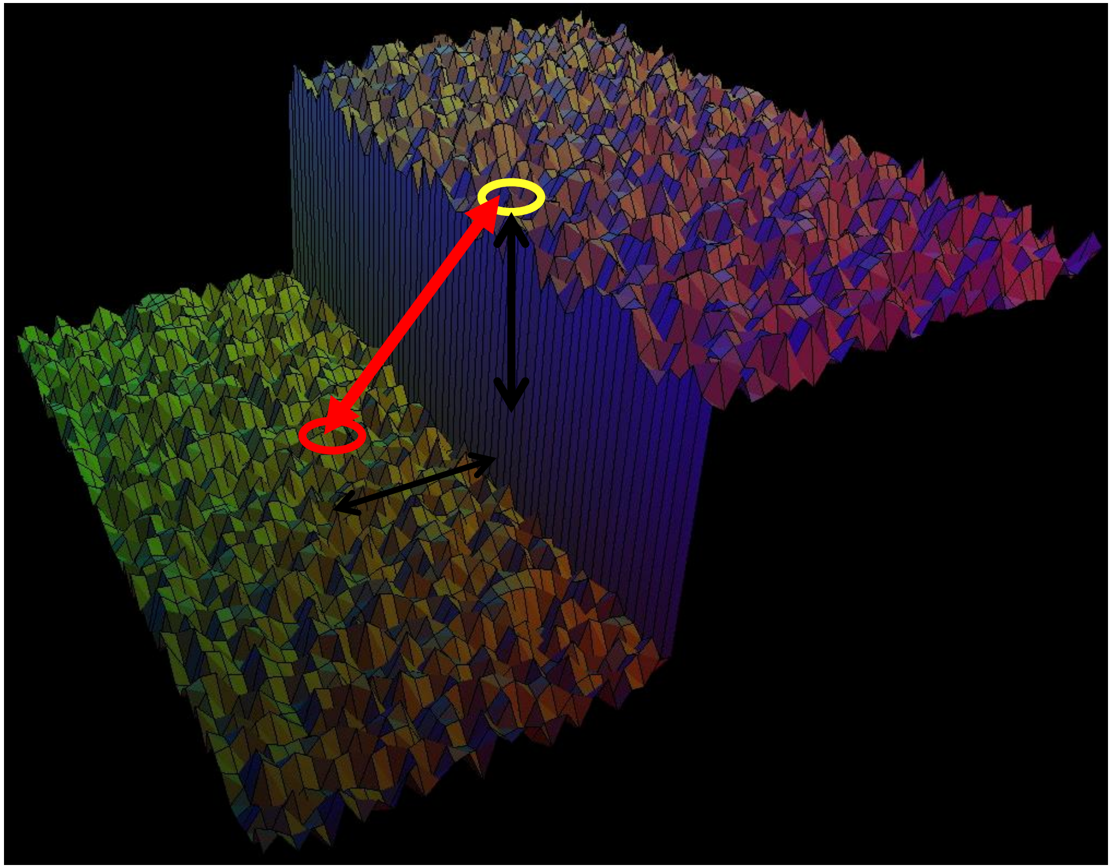
Input

Output

Tomasi and Manduchi 1998

Other view

- The bilateral filter uses the 3D distance



Slide credit: F. Durand

Speed

- Direct bilateral filtering is slow (minutes)

- Accelerations exist:
  - Subsampling in space & range
    - Durand & Dorsey 2002
    - Paris & Durand 2006
  - Limit to box kernel & intelligent maintenance of histogram
    - Weiss 2006

Slide credit: F. Durand

Local filters

- Compute a new value at each pixel using its neighboring pixels
  - Box filter
  - Gaussian filter
  - Median filter
  - Bilateral filter

Non-local means filter

- Compute a new value at each pixel from the whole image

$$NL[v](i) = \sum_{j \in I} w(i,j)\,v(j)$$

final value at pixel *i*

weight of pixel *j*

value at pixel *j*

Buades, A., Coll, B., Morel, J.-M. A non-local algorithm for image denoising. *CVPR 2005*

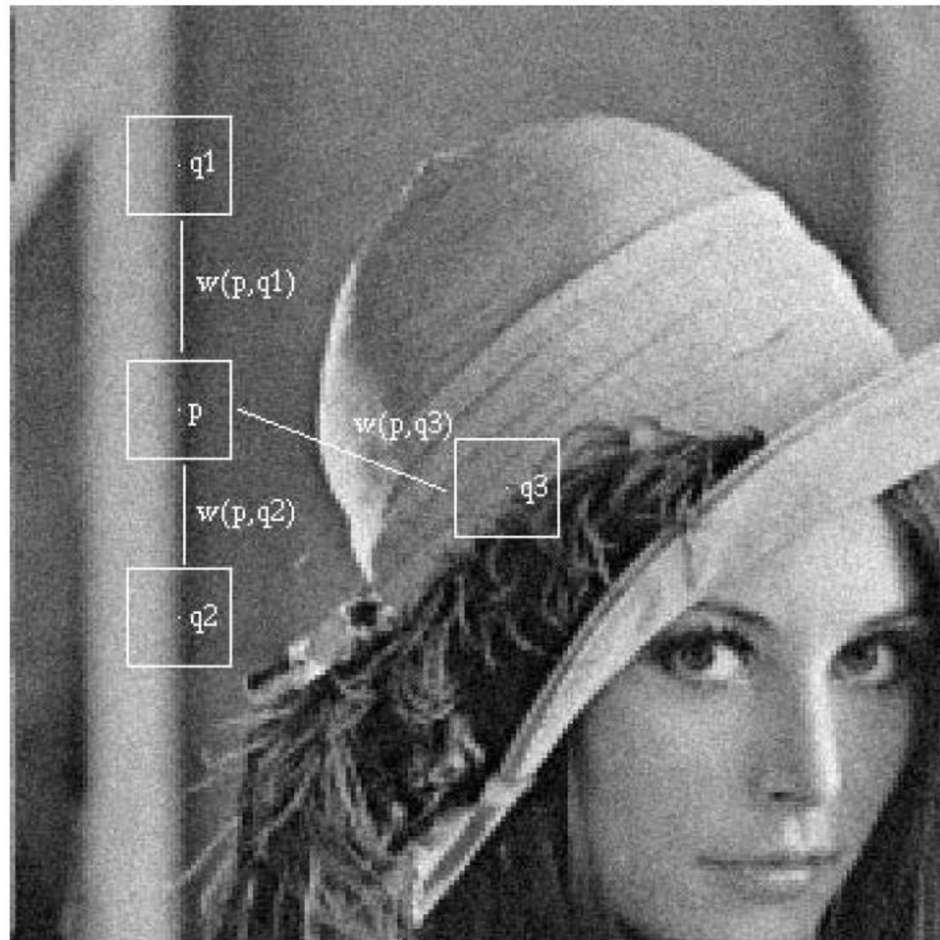
Weight

$$w(i,j) = \frac{1}{Z(i)} e^{-\frac{||v(\mathcal{N}_i) - v(\mathcal{N}_j)||_{2,a}^2}{h^2}}$$

$v(\mathcal{N}_i)$: patch centered at pixel *i*

$v(\mathcal{N}_j)$: patch centered at pixel *j*

Similar pixel neighborhoods give a large weight



Reprint from *Buades et al.* 2005

Reprint from Buades *et al*. 2005

Non-local means filter

- High-quality
- Slow
  - Fast non-local means algorithms available

Video de-noise

- We know how to de-noise an image
- How about video?

E. P. Bennett and L. McMillan. Video Enhancement using Per-pixel Virtual Exposures, SIGGRAPH 2005

Gaussian filter in video cube

- Blurring artifacts
  - Not edge-preserving
  - Motion blur

Bilateral filter in video cube

- Cannot remove shot noise

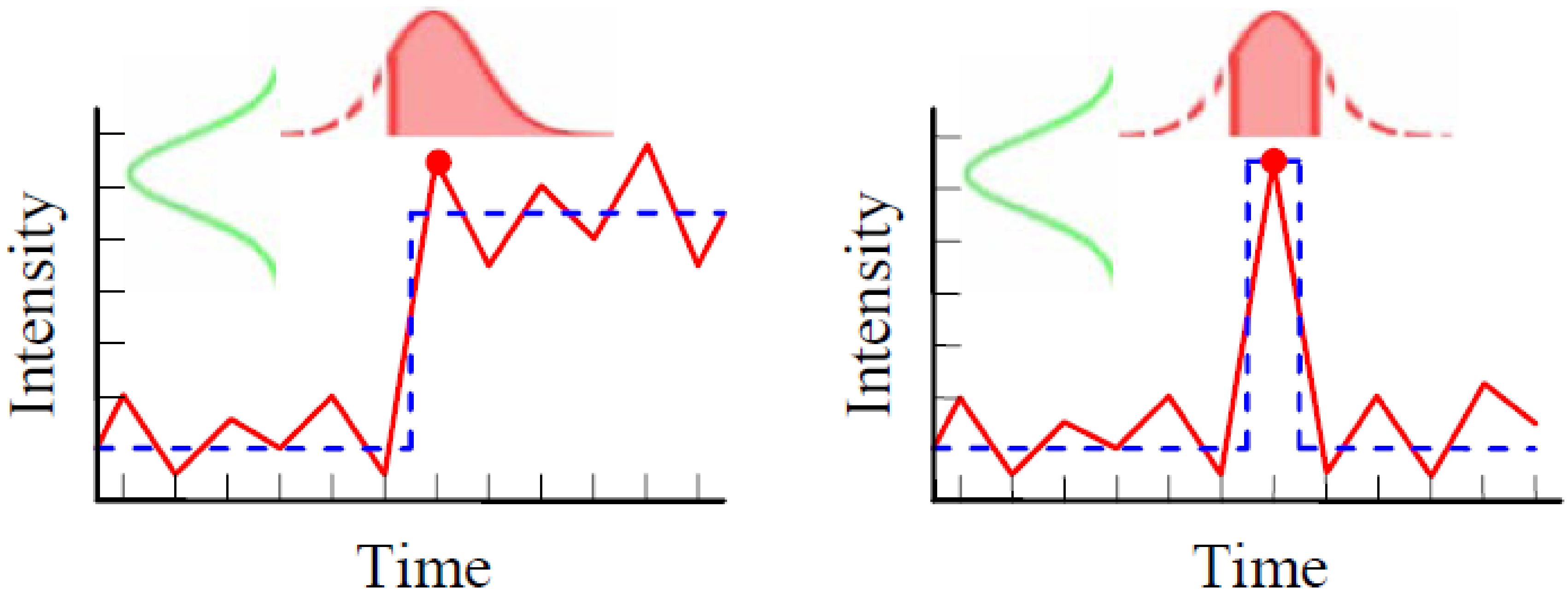


Figure 3: Left: The bilateral filter recovers the signal (blue) from the noisy input (red). Right: The bilateral filter is unable to attenuate the shot noise because no other pixels fall within the intensity dissimilarity Gaussian.

Reprint from [Bennett and McMillan 2005]

ASTA Filter [Bennett and McMillan ‘05]

- Build upon bilateral filter
- Find similar pixels in a video cube for filtering
  - Patch-based similarity measurement
- Adaptive Spatial-temporal Accumulation Filter
  - Prefer temporal neighbors

Patch-based similarity measurement

$$D\left(p_{xyt}, s_{xyt}\right) = \frac{\sum_{x=sx-n}^{sx+n} \sum_{y=sy-n}^{sy+n} g\left(\left\| x-p_x, y-p_y \right\|, \sigma_e\right) \left| I_{x,y,pt} - I_{x,y,st} \right|}{\sum_{x=sx-n}^{sx+n} \sum_{y=sy-n}^{sy+n} g\left(\left\| x-p_x, y-p_y \right\|, \sigma_e\right)}$$

Similarity measurement

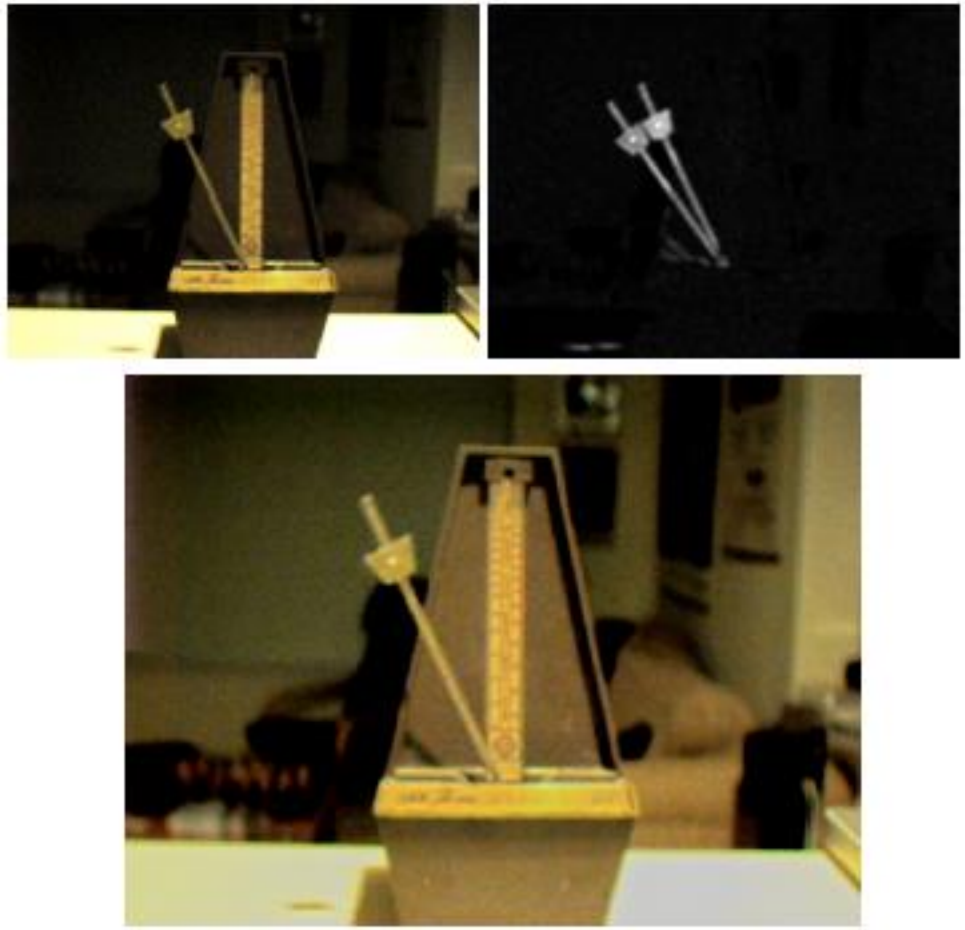


Figure 4: Illustration of our spatial neighborhood similarity distance used in temporal filtering. The original frame is shown in the upper left. Each *(x,y)* for a pair of nearby frames are shown in the upper right. Two metronome arms are seen because the similarity distance is based on absolute value. The bottom image is the same frame processed using ASTA and our tone mapper.

Reprint from
[Bennett and McMillan 2005]

Adaptive Filtering

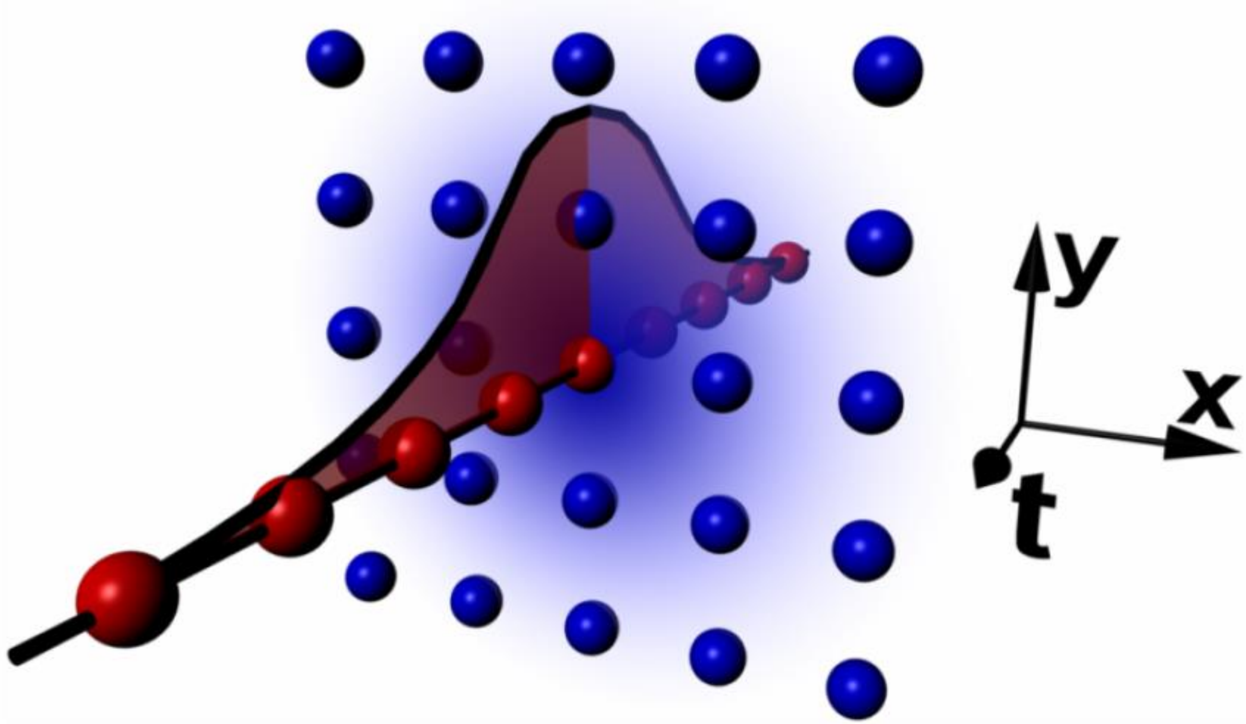


Figure 5: Illustration of the temporal-only and spatial-only nature of ASTA. The temporally filtered red pixels are preferred to be integrated into the filter, but if not enough are similar to the center of the kernel, the blue spatial pixels begin to be integrated.

Reprint from [Bennett and McMillan 2005]

Results (filtering + tone mapping)

Input

Naïve method

ASTA

Reprint from [Bennett and McMillan 2005]

Denoising Summary

- Find relevant pixels for denoising
  - Spatial neighbors
    - Gaussian filter
  - Spatial neighbors with similar color
    - Bilateral filter
  - Pixels with similar patches
    - No-local mean
  - Pixels in spatial-temporal neighborhood with similar patches
    - Video denoising

Student paper presentation

Accelerating Spatially Varying Gaussian Filters

Baek, J., and Jacobs, D. E.
SIGGRAPH Asia 2010

Presenter: Dave Howell

Next Time

- Color
- Lighting
- Student paper presentation
  - Joint bilateral upsampling
    J. Kopf, M. Cohen, D. Lischinski, and M. Uyttendaele
    SIGGRAPH 2007
  - By Singh, Harmandeep