Image Filtering & Its Applications

-- Junjie Cao

Introduction

Goal: Image Smoothing

Split an image into:

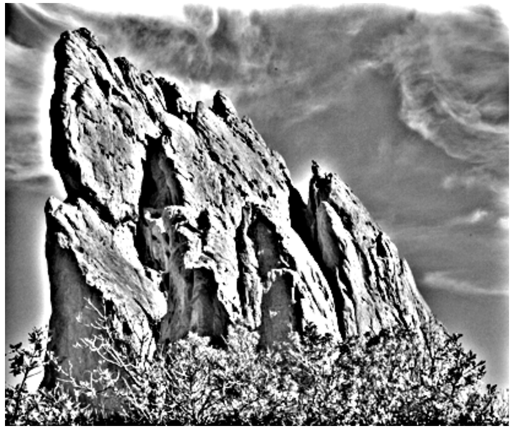
- large-scale features, structure
- small-scale features, texture

input

BLUR

smoothed
(structure, large scale)

HALOS



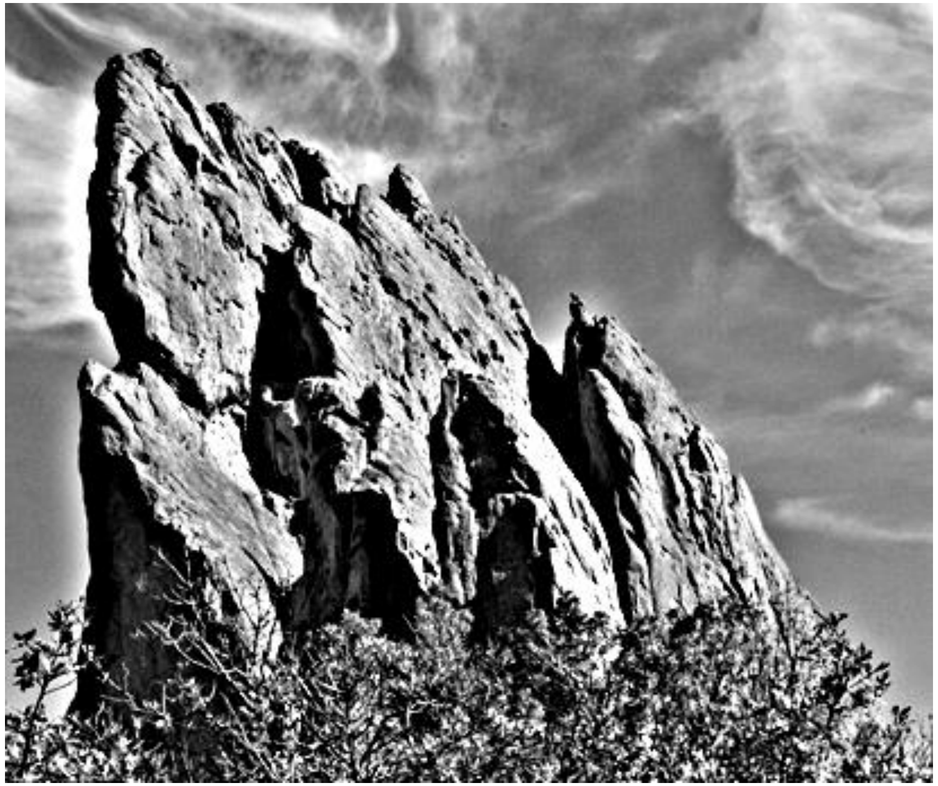
residual
(texture, small scale)

Gaussian Convolution

Impact of Blur and Halos

- If the decomposition introduces blur and halos, the final result is corrupted.

Sample manipulation: increasing texture (residual $\times$ 3)



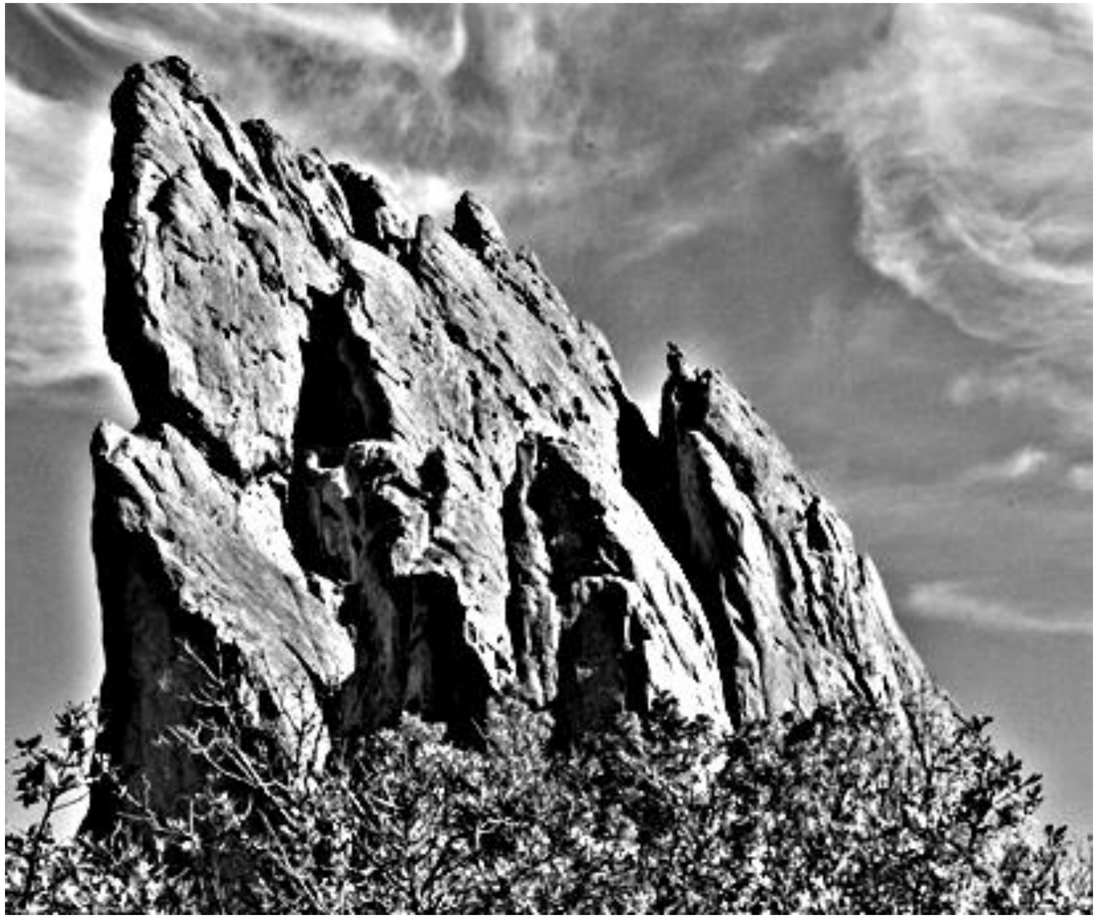
Bilateral Filter (no Blur, no Halos) vs Naïve Approach: Gaussian Blur (with blur & halos)

input

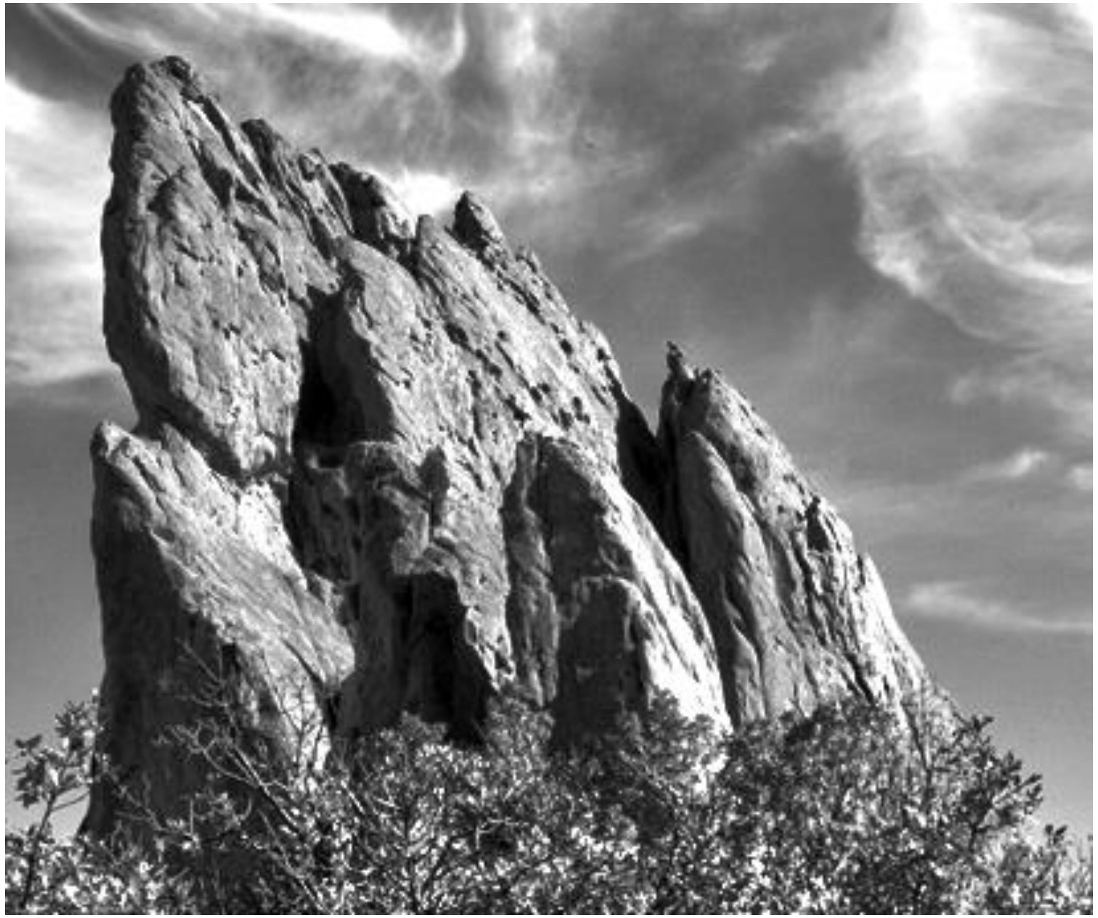
smoothed
(*structure, large scale*)

residual
(*texture, small scale*)

edge-preserving: Bilateral Filter

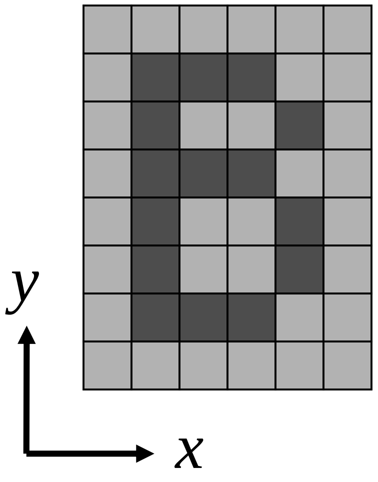


increasing texture
with Gaussian convolution
H A L O S



increasing texture
with bilateral filter
N O H A L O S

Notation and Definitions



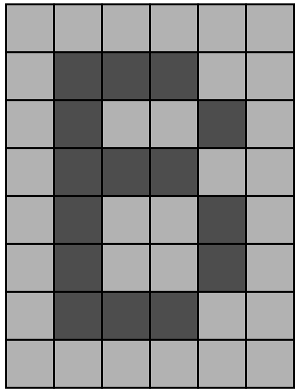
- Image = 2D array of pixels

- Pixel = intensity (scalar) or color (3D vector)

- $I_{\mathbf{p}}$ = value of image I at position: $\mathbf{p} = (\, p_x \,, p_y \,)$

- $F\,[\,I\,]$ = output of filter F applied to image I

What is filtering?



- Images are not smooth because adjacent pixels are different.
- Smoothing = making adjacent pixels look more similar.
- Smoothing strategy
  - pixel → average of its neighbors

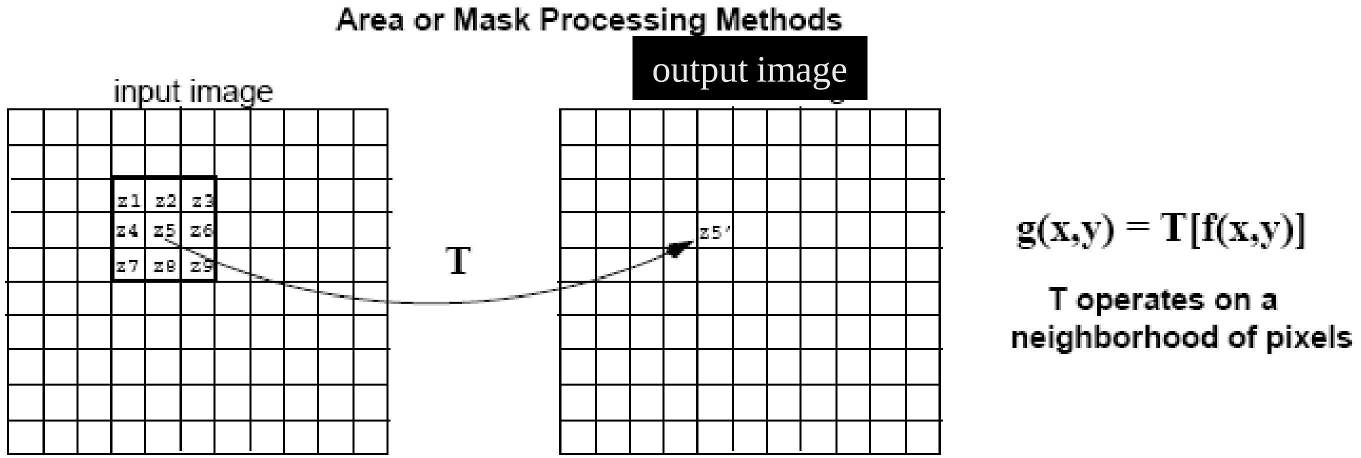
Spatial filtering/Mask processing - Operation

<u>Example</u>: weighted sum of input pixels.

$$z5' = R = w1z1 + w2z2 + \ldots + z9w9$$

mask weights:

w1	w2	w3
w4	w5	w6
w7	w8	w9

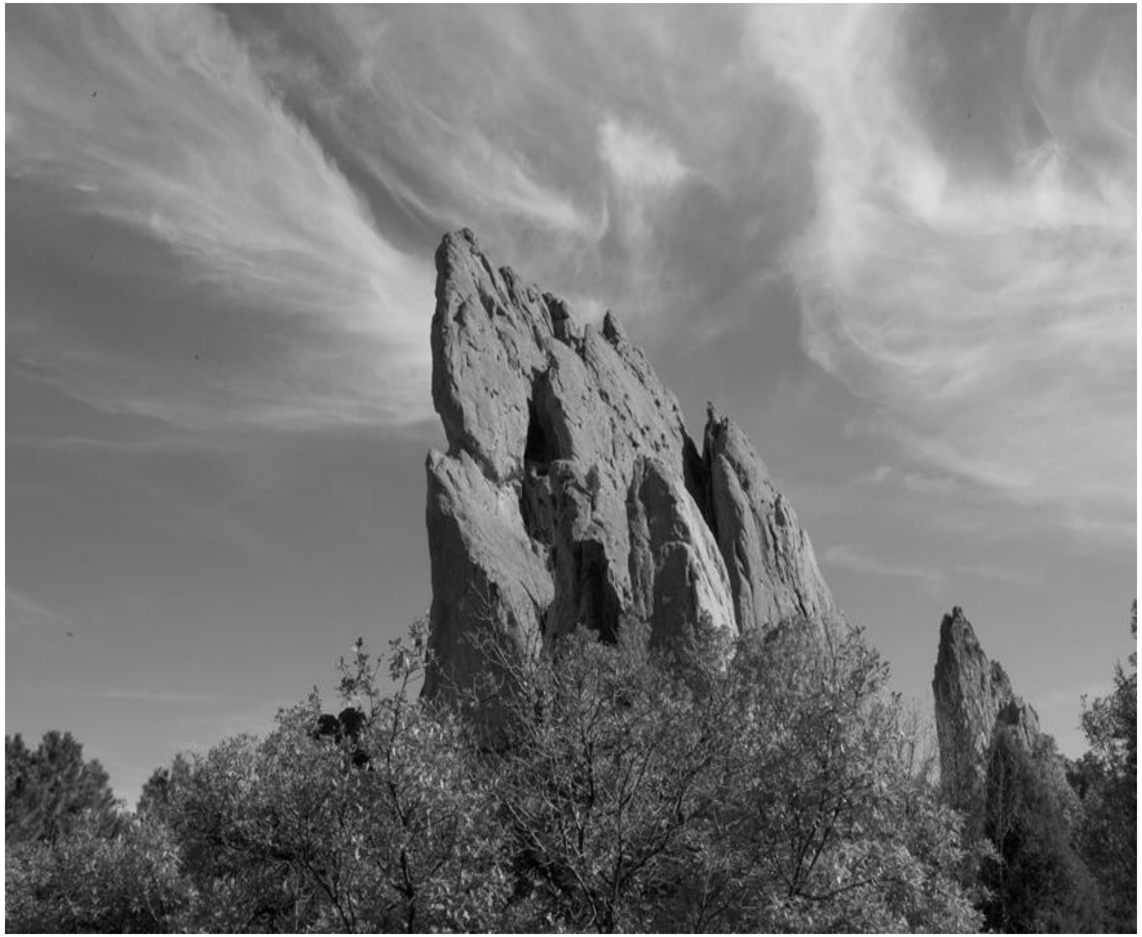


A filtered image is generated as the center of the mask moves to every pixel in the input image.

Why is it important?

- Many applications with high quality results.
  - Computational photography
  - Computer graphics
- Papers in top conferences
  - Siggraph 15: An L1 Image Transform for Edge-Preserving Smoothing and Scene-Level Intrinsic Decomposition
  - Siggraph asia 15: Rolling Guidance Normal Filter for Geometric Processing
  - Siggraph 14: Bilateral texture filtering
  - Siggraph 14: Fast Local Laplacian Filters: Theory and Applications
  - Siggraph asia 14: Depth of field rendering via adaptive recursive filtering
  - …

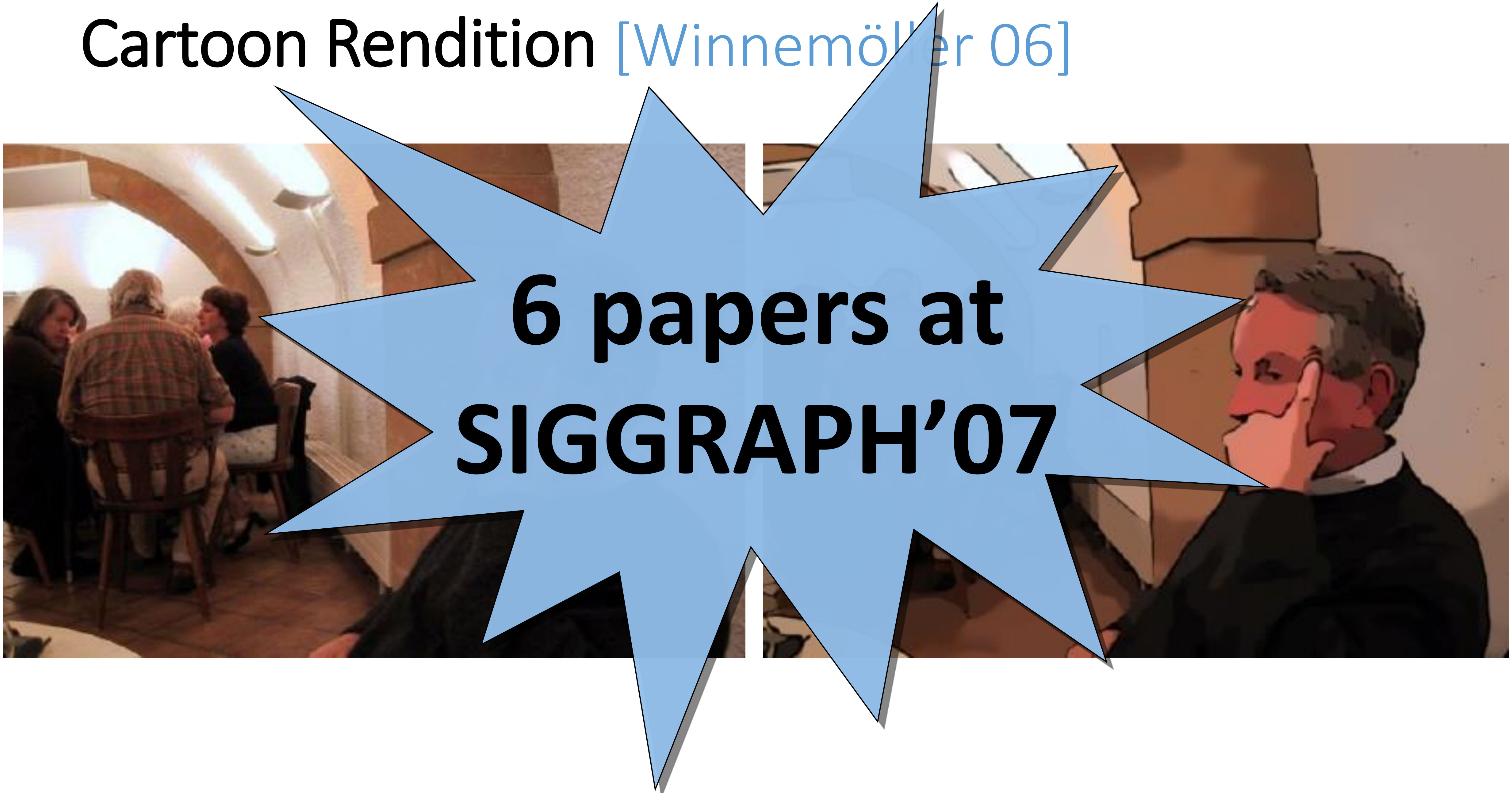
Photographic Style Transfer [Bae 06]



Tone Mapping [Durand 02]

HDR input

Cartoon Rendition [Winnemöller 06]



Keywords & terms

- Smoothing, denoising, blur, decomposition

- Feature, Structure, texture
- Blur, halo

- Convolution, kernel

- Spatial Filtering,

Many Other Options

- Bilateral filtering is not the only image smoothing filter
  - Diffusion, wavelets, Bayesian…
- We focus on bilateral filtering
  - Suitable for strong smoothing used in computational photography
  - Conceptually simple

Content of the Course

All you need to know about bilateral filtering:

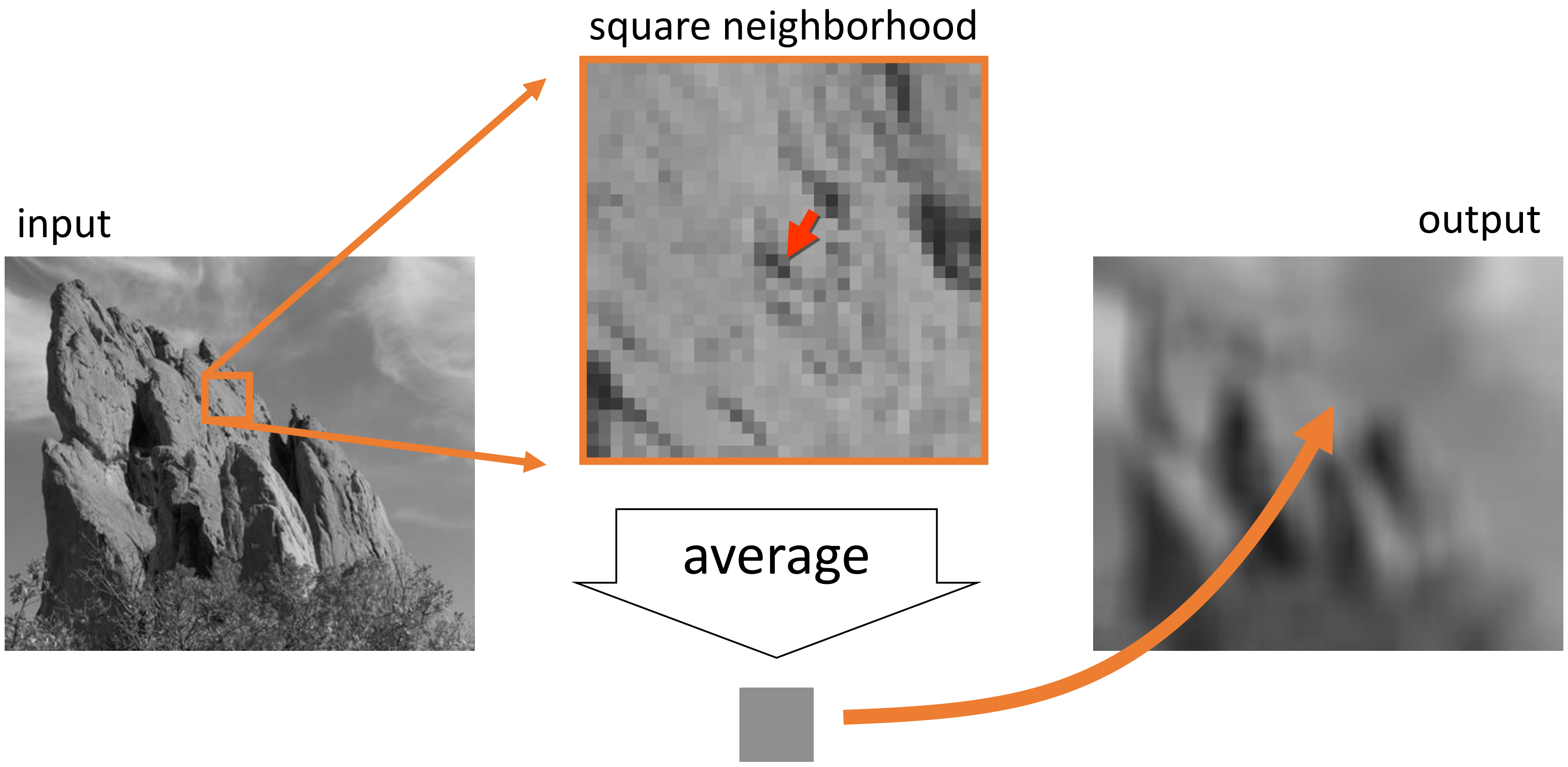
- Definition of the bilateral filter
- Parameter influence and settings
- Applications
- Relationship to other filters
- Theoretical properties
- Efficient implementation

Course Material

- Course webpage (google “bilateral filter course”): http://people.csail.mit.edu/sparis/bf_course/)
- Detailed course notes
  - C++ and Matlab code

Naïve Image Smoothing: Gaussian Blur

Box Average



Equation of Box Average

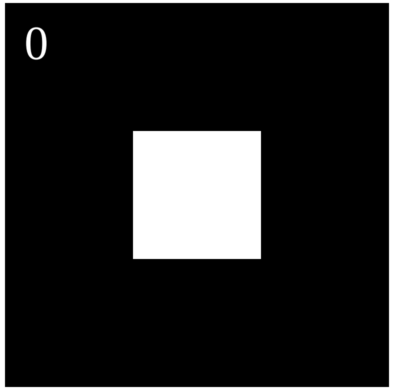
$$BA[I]_{\mathbf{p}} \qquad \sum_{\mathbf{q}\in S} B_\sigma(\mathbf{p}-\mathbf{q})\, I_{\mathbf{q}}$$

result at pixel $\mathbf{p}$

sum over all pixels $\mathbf{q}$

intensity at pixel $\mathbf{q}$

normalized box function



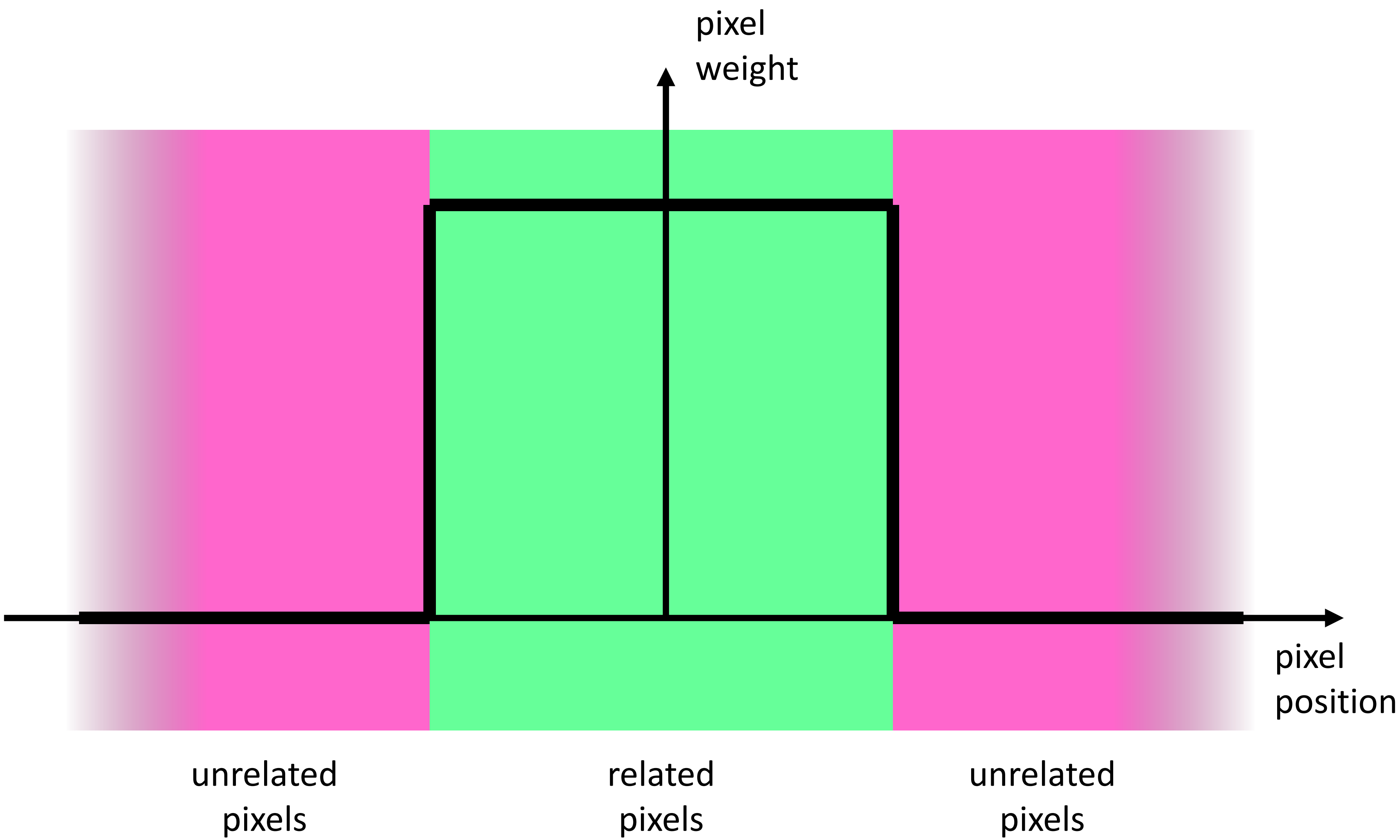
Square Box Generates Defects

- Axis-aligned streaks
- Blocky results

input

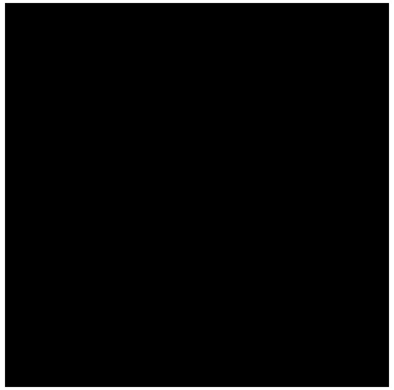
output

Box Profile

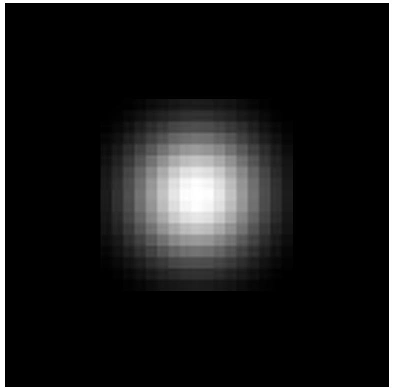

pixel
weight
pixel
position
unrelated
pixels
related
pixels
unrelated
pixels

Strategy to Solve these Problems

- Use an isotropic (*i.e.* circular) window.
- Use a window with a smooth falloff.

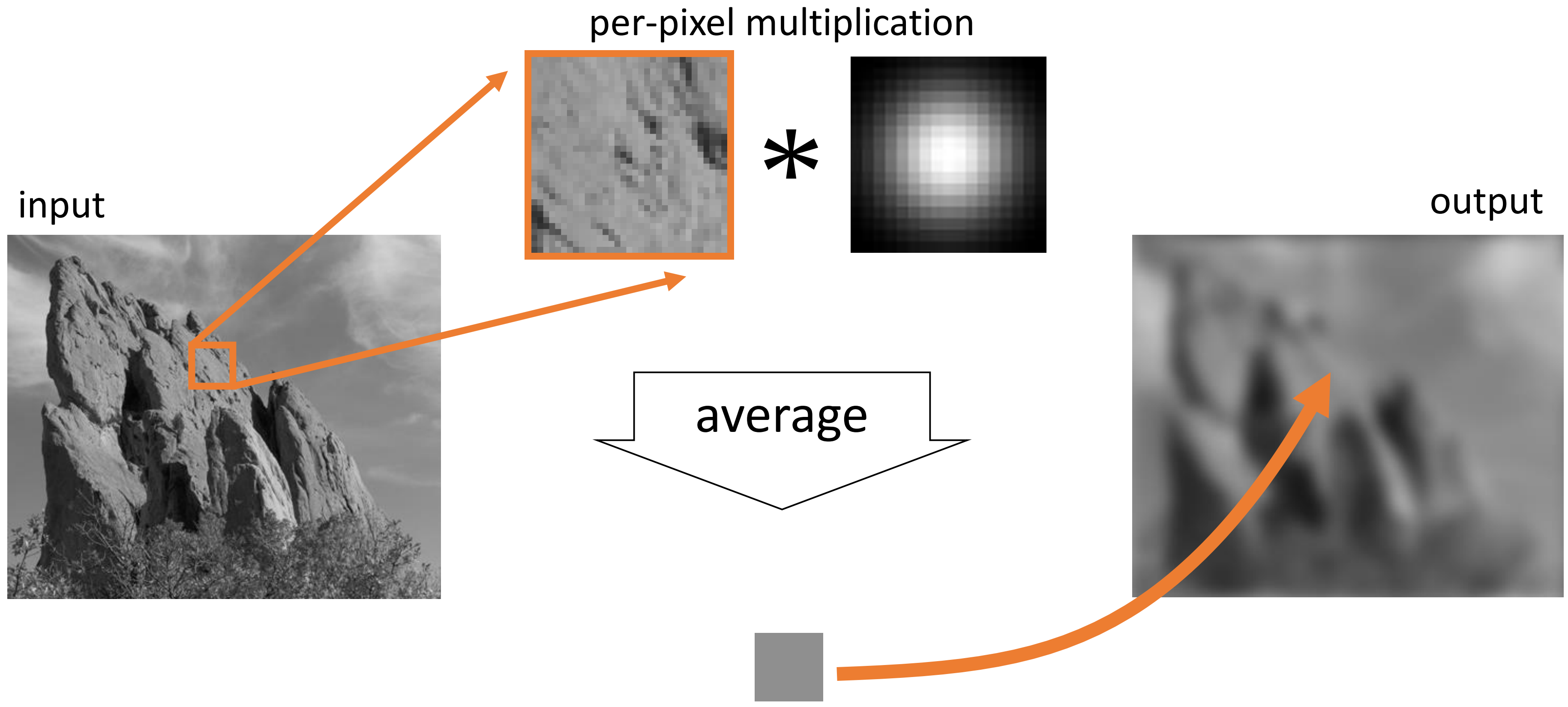


box window



Gaussian window

Gaussian Blur

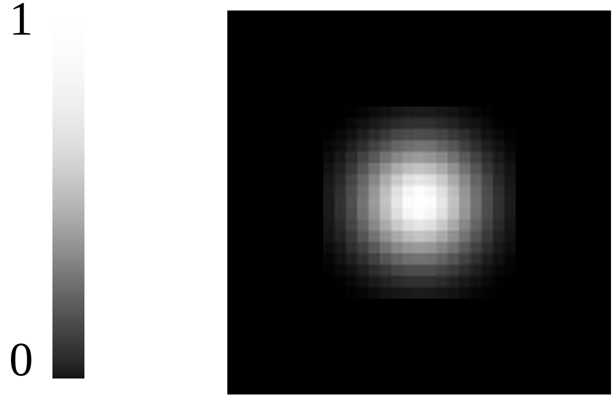


Equation of Gaussian Blur

Same idea: **weighted average of pixels**.

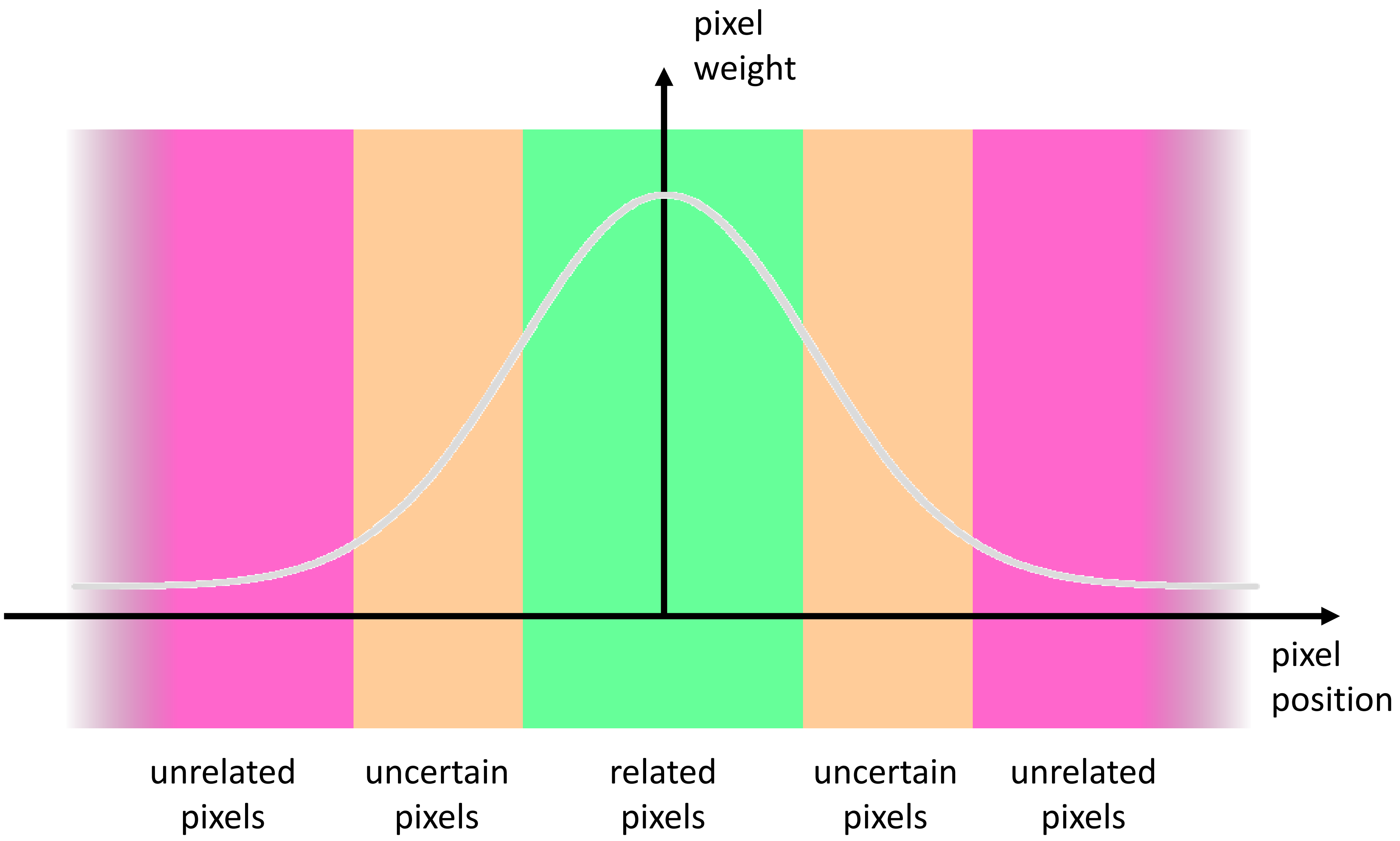
$$GB[I]_{\mathbf{p}} = \sum_{\mathbf{q} \in S} \boxed{G_\sigma(\|\mathbf{p} - \mathbf{q}\|)}\, I_{\mathbf{q}}$$

normalized
Gaussian function



Gaussian Profile

$$G_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

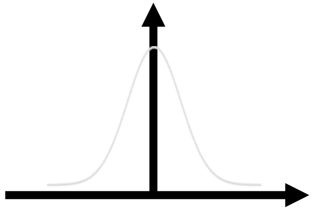


Spatial Parameter

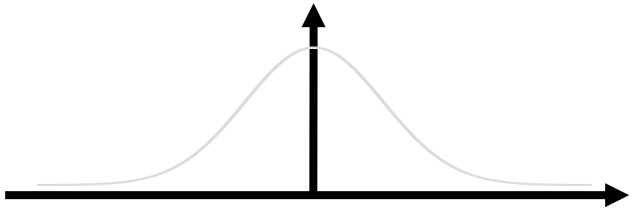
input

$$GB[I]_{\mathbf{p}} = \sum_{\mathbf{q}\in S} G_{\boxed{\sigma}}(\|\mathbf{p}-\mathbf{q}\|)\, I_{\mathbf{q}}$$

size of the window



small σ



large σ

limited smoothing

strong smoothing

How to set σ

- Depends on the application.

- Common strategy: proportional to image size
  - e.g. 2% of the image diagonal
  - property: independent of image resolution

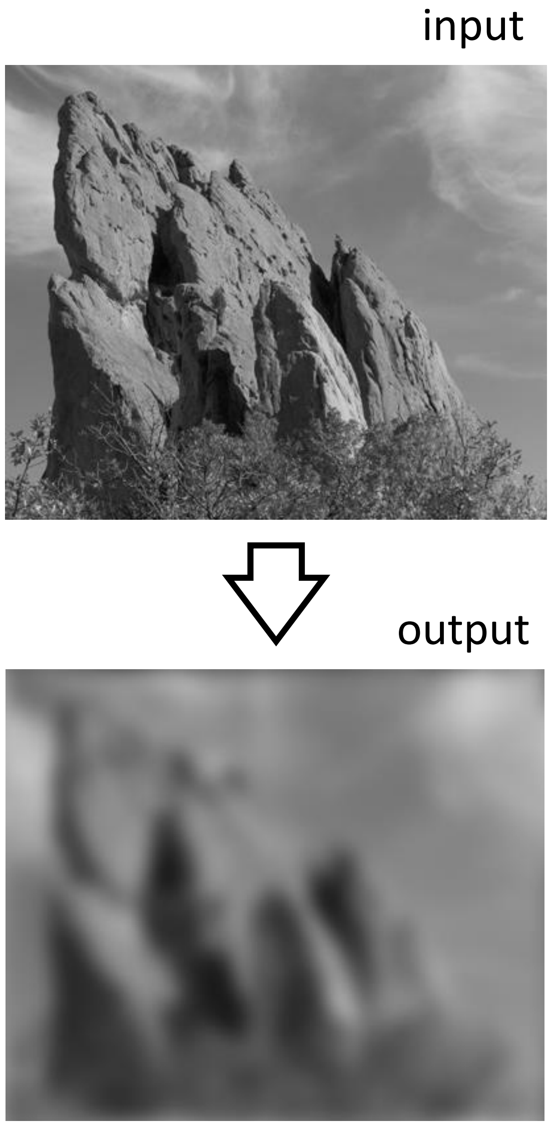
Properties of Gaussian Blur

- Weights independent of spatial location
    - linear convolution
    - well-known operation
    - efficient computation (recursive algorithm, FFT…)

Properties of Gaussian Blur

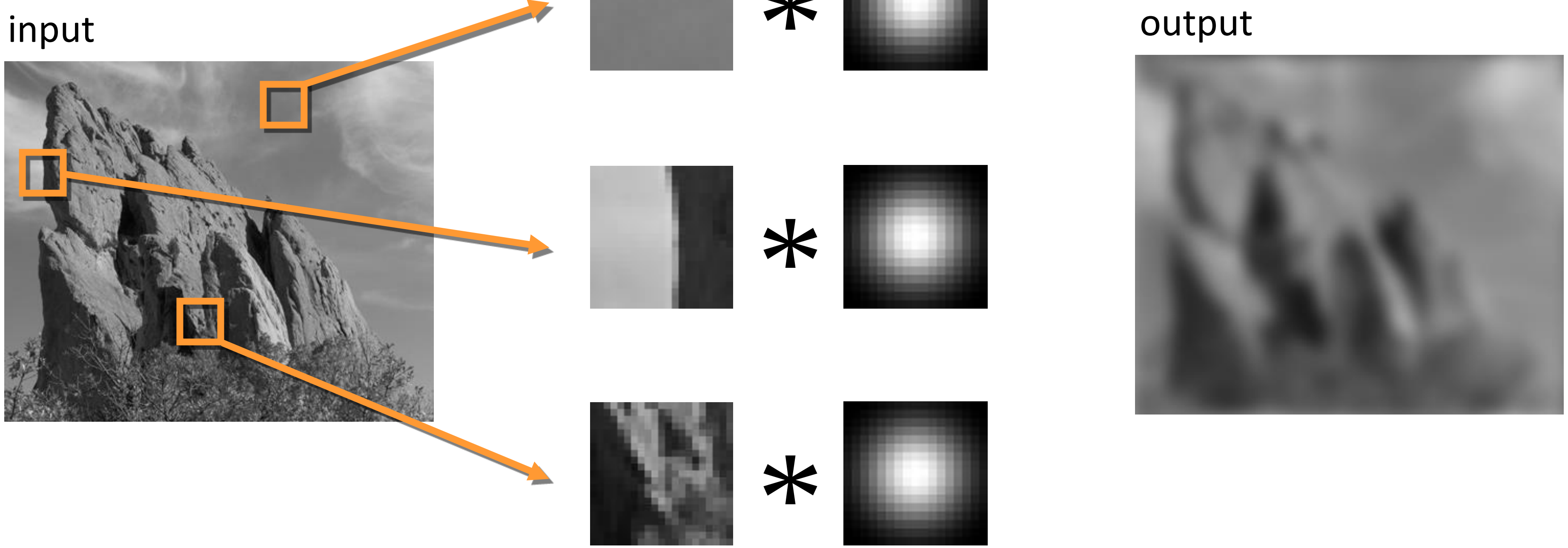
- Does smooth images
- But smoothes too much: **edges are blurred**.
  - Only spatial distance matters
  - No edge term

$$GB[I]_{\mathbf{p}} = \sum_{\mathbf{q}\in S} \underbrace{G_\sigma(\|\mathbf{p}-\mathbf{q}\|)}_{\text{space}} I_{\mathbf{q}}$$



“Fixing the Gaussian Blur”: the Bilateral Filter

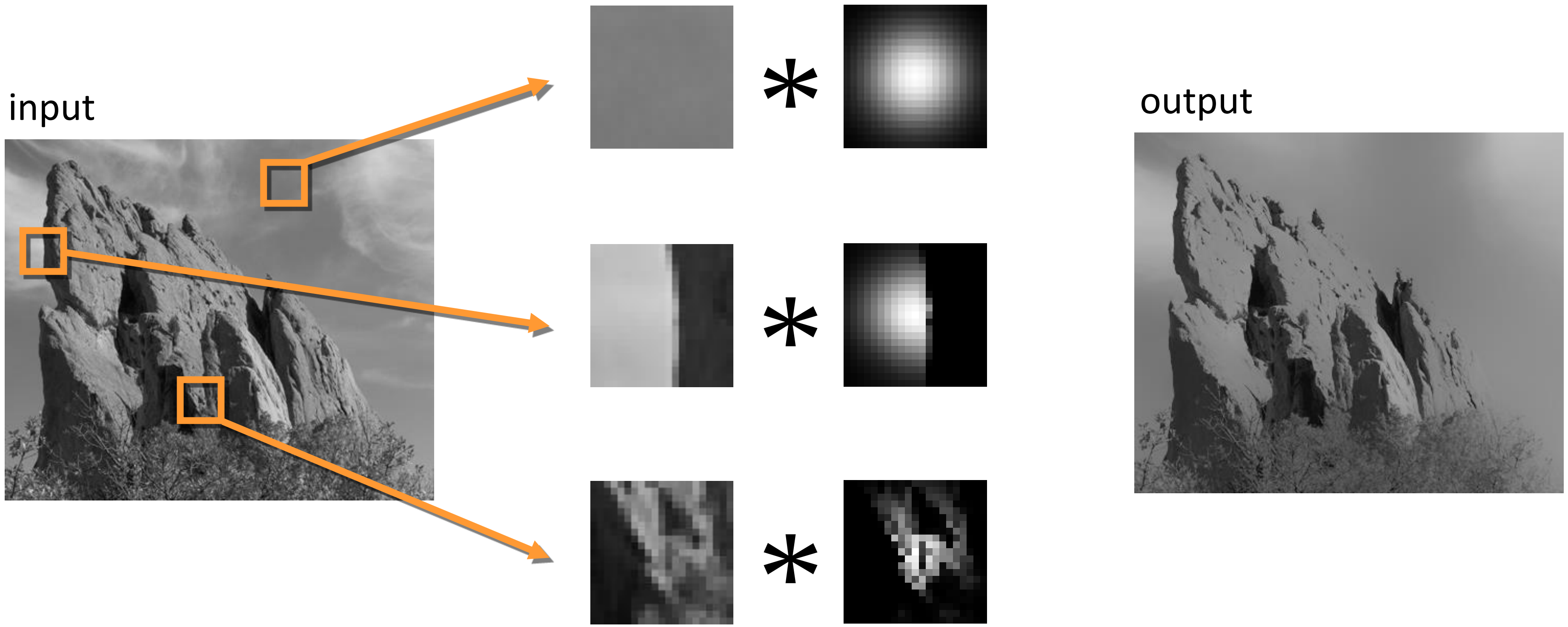
Blur Comes from Averaging across Edges



Same Gaussian kernel everywhere.

Bilateral Filter
No Averaging across Edges

[Aurich 95, Smith 97, Tomasi 98]

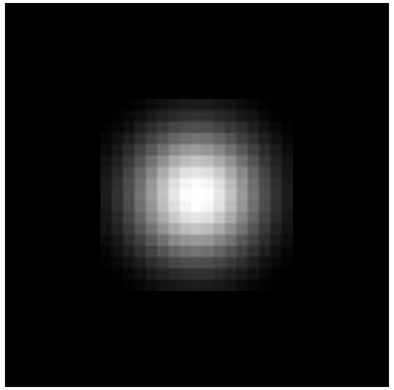


The kernel shape depends on the image content.

Bilateral Filter Definition: an Additional Edge Term

Same idea: **weighted average of pixels.**

$$BF[I]_{\mathbf{p}} = \overset{\text{new}}{\underbrace{\frac{1}{W_{\mathbf{p}}}}_{\text{normalization factor}}} \sum_{\mathbf{q} \in S} \overset{\text{not new}}{\underbrace{G_{\sigma_s}(\|\mathbf{p}-\mathbf{q}\|)}_{\textit{space}\text{ weight}}} \; \overset{\text{new}}{\underbrace{G_{\sigma_r}(|I_{\mathbf{p}} - I_{\mathbf{q}}|)}_{\textit{range}\text{ weight}}} \; I_{\mathbf{q}}$$



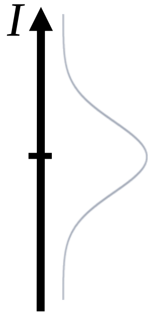
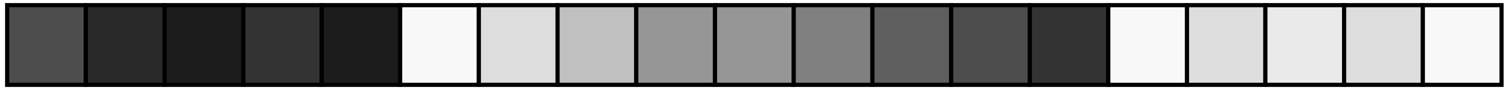
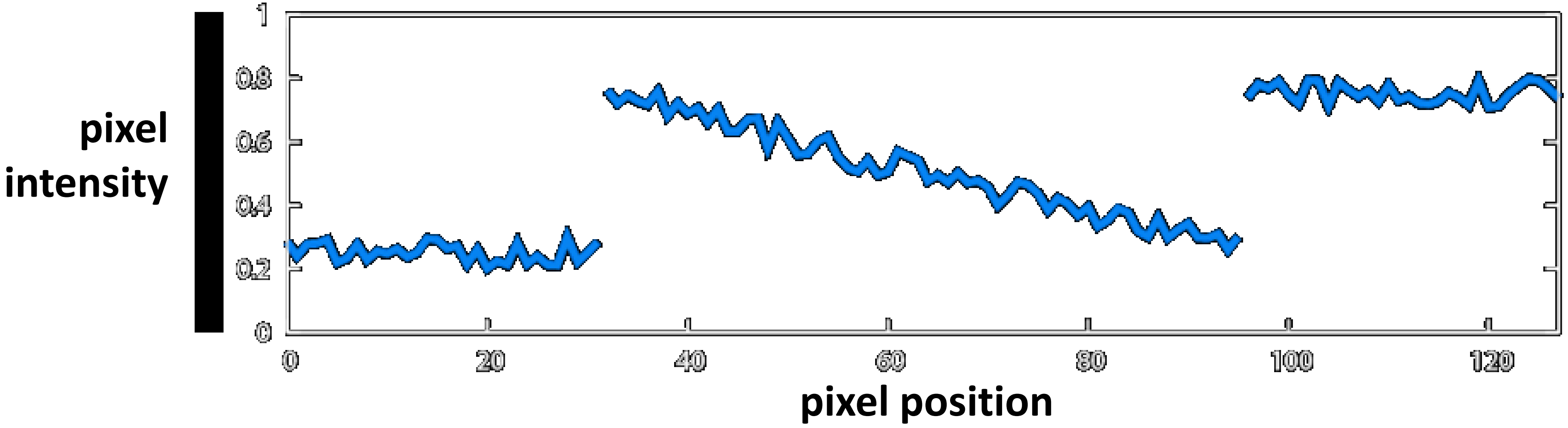


Illustration a 1D Image

- 1D image = line of pixels

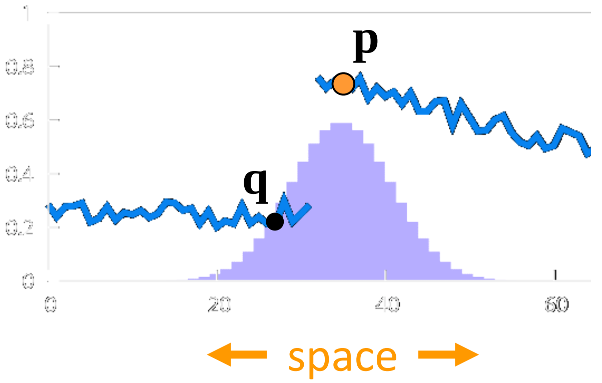


- Better visualized as a plot



Gaussian Blur and Bilateral Filter

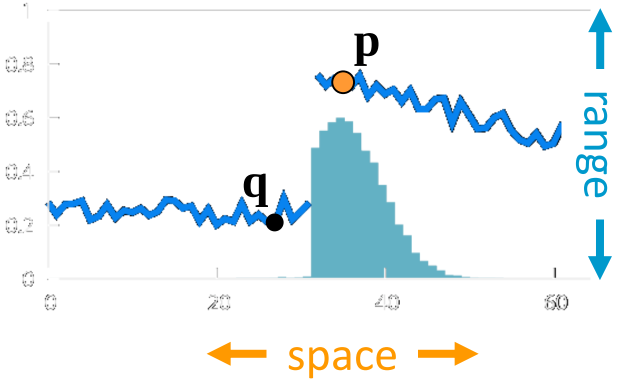
Gaussian blur



$$GB[I]_p = \sum_{q \in S} \underbrace{G_\sigma(\| p - q \|)}_{\text{space}} I_q$$

Bilateral filter

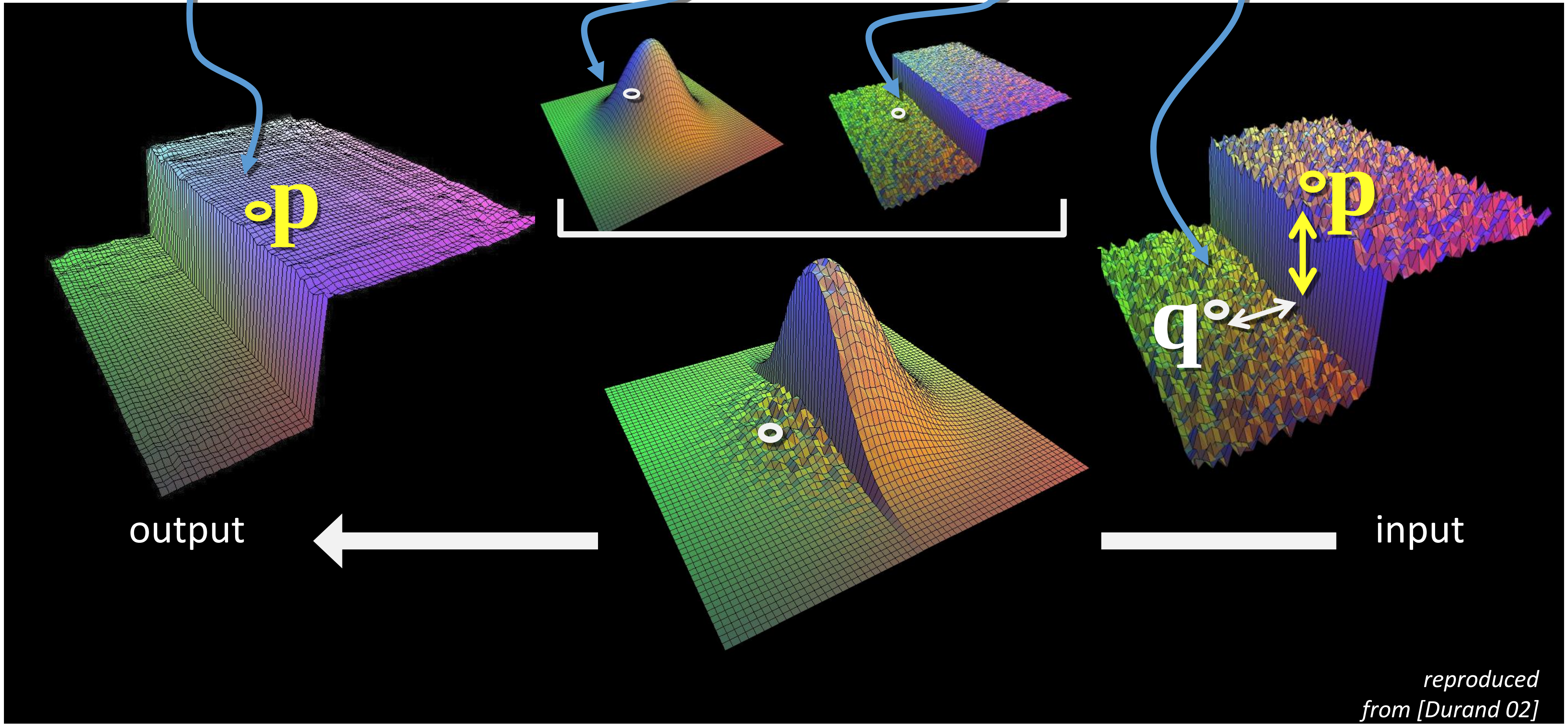
[Aurich 95, Smith 97, Tomasi 98]



$$BF[I]_p = \underbrace{\frac{1}{W_p}}_{\text{normalization}} \sum_{q \in S} \underbrace{G_{\sigma_s}(\| p - q \|)}_{\text{space}} \underbrace{G_{\sigma_r}(| I_p - I_q |)}_{\text{range}} I_q$$

Bilateral Filter on a Height Field

$$\underbrace{BF[I]_{\mathbf{p}}}\ = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} \underbrace{G_{\sigma_s}(\|\mathbf{p}-\mathbf{q}\|)} \underbrace{G_{\sigma_r}(|I_{\mathbf{p}} - I_{\mathbf{q}}|)} I_{\mathbf{q}}$$



Space and Range Parameters

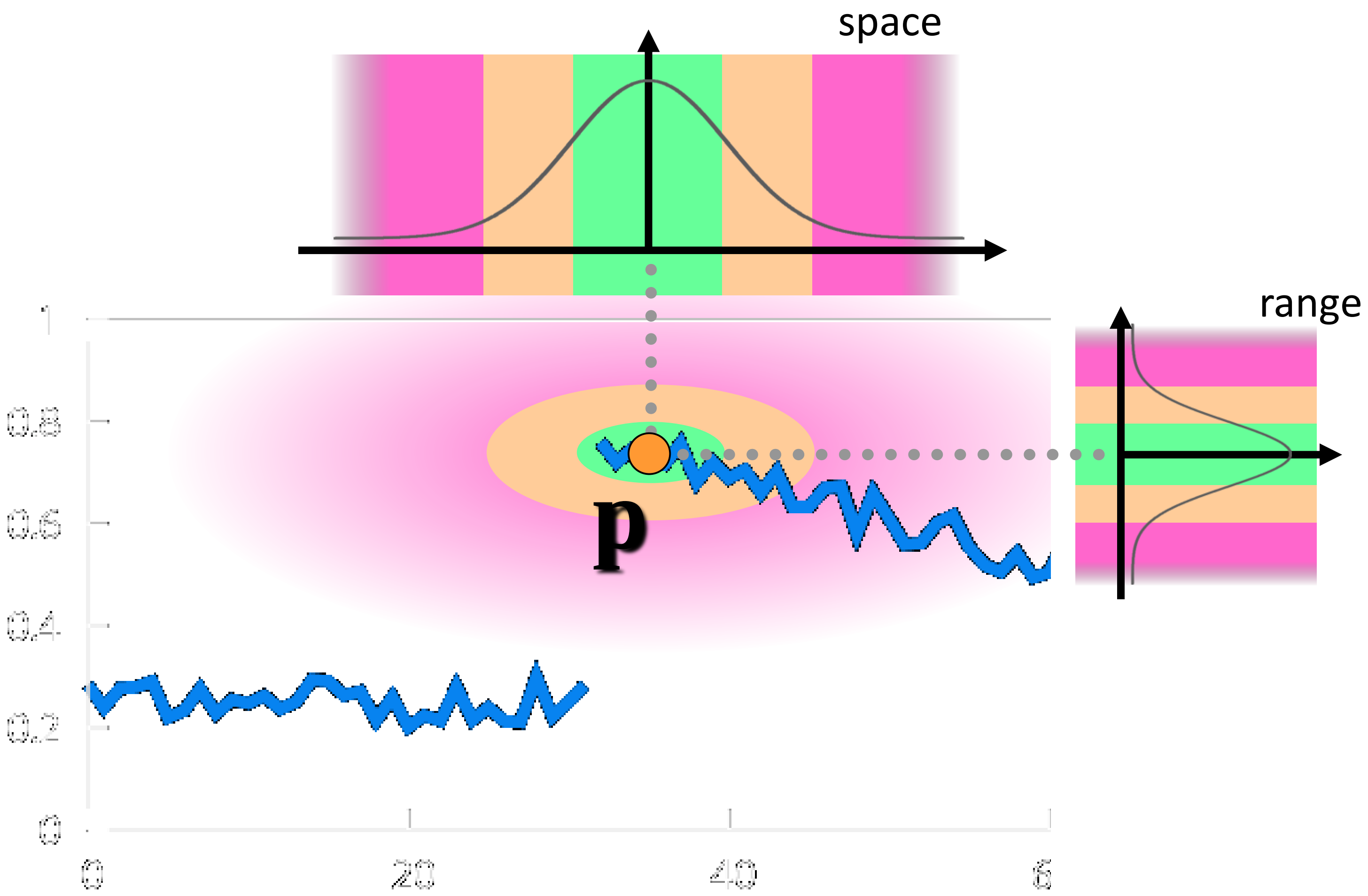
$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q}\in S} G_{\sigma_{\mathrm{s}}}\left(\| \mathbf{p} - \mathbf{q} \|\right) G_{\sigma_{\mathrm{r}}}\left(| I_{\mathbf{p}} - I_{\mathbf{q}} |\right) I_{\mathbf{q}}$$

- space σ_{s} : spatial extent of the kernel, size of the considered neighborhood.

- range σ_{r} : “minimum” amplitude of an edge

Influence of Pixels

Only pixels close in space and in range are considered.



Varying the Range Parameter

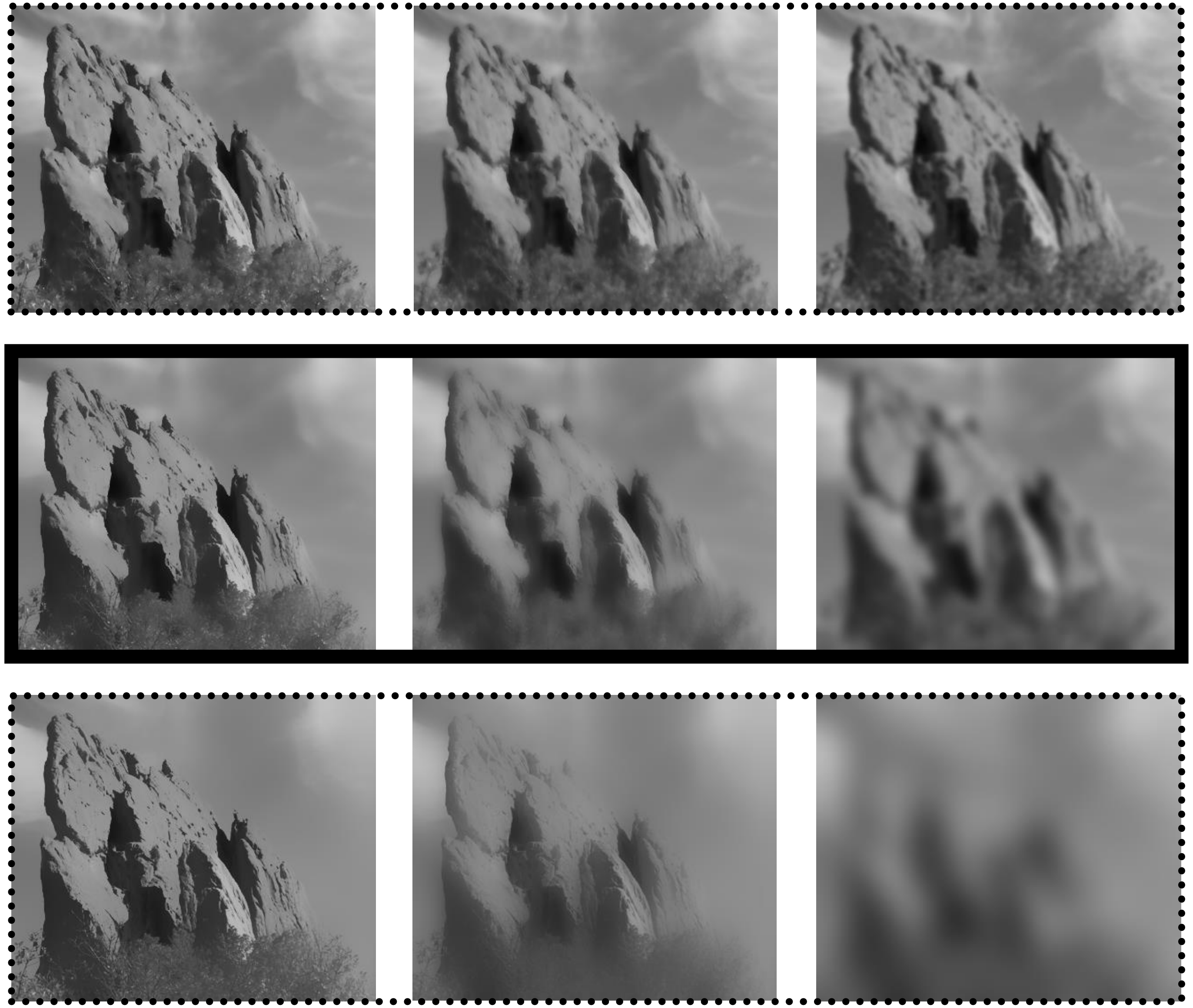
input

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

$\sigma_s = 2$

$\sigma_s = 6$

$\sigma_s = 18$



Varying the Space Parameter

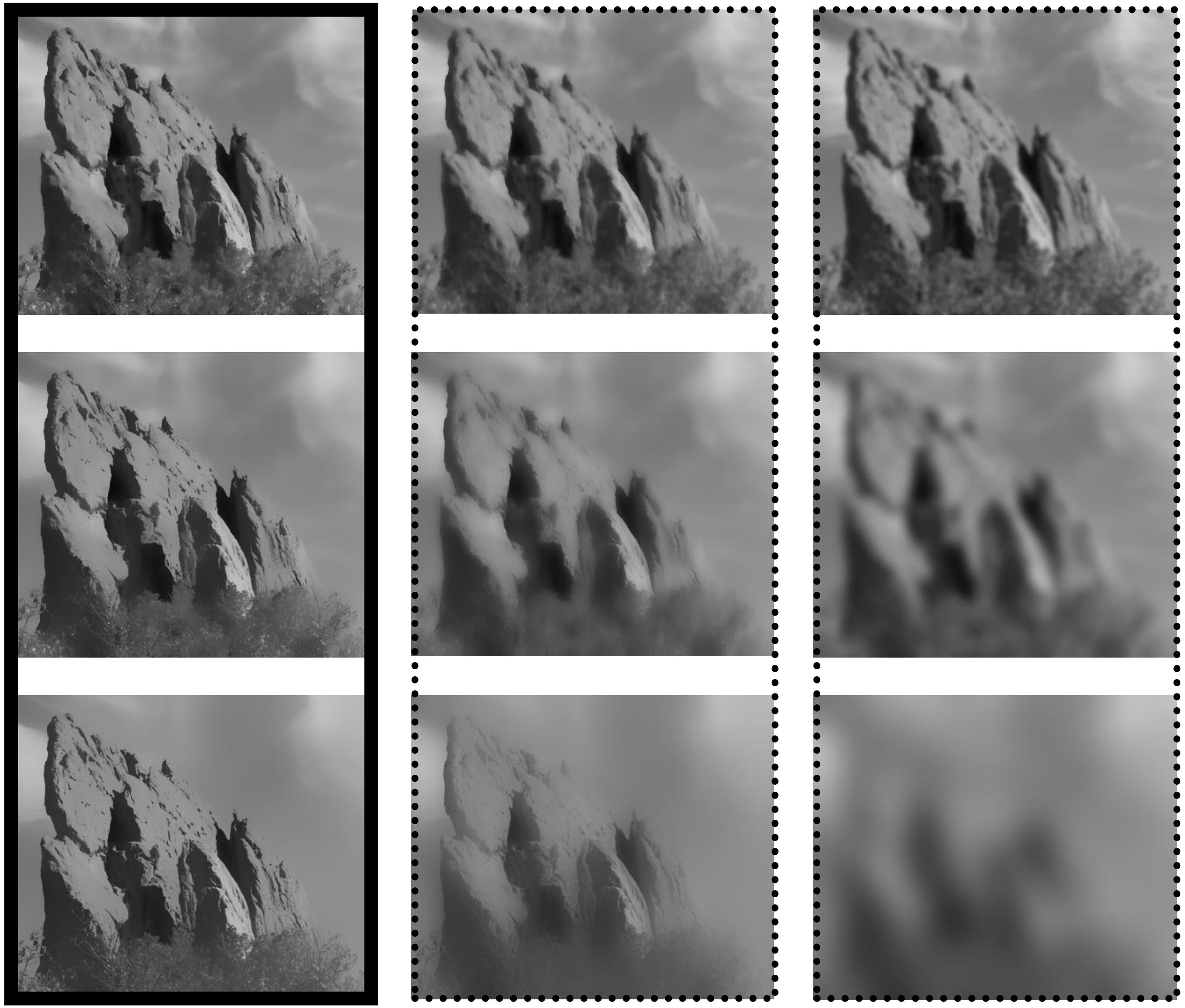
input

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

$\sigma_s = 2$

$\sigma_s = 6$

$\sigma_s = 18$



How to Set the Parameters

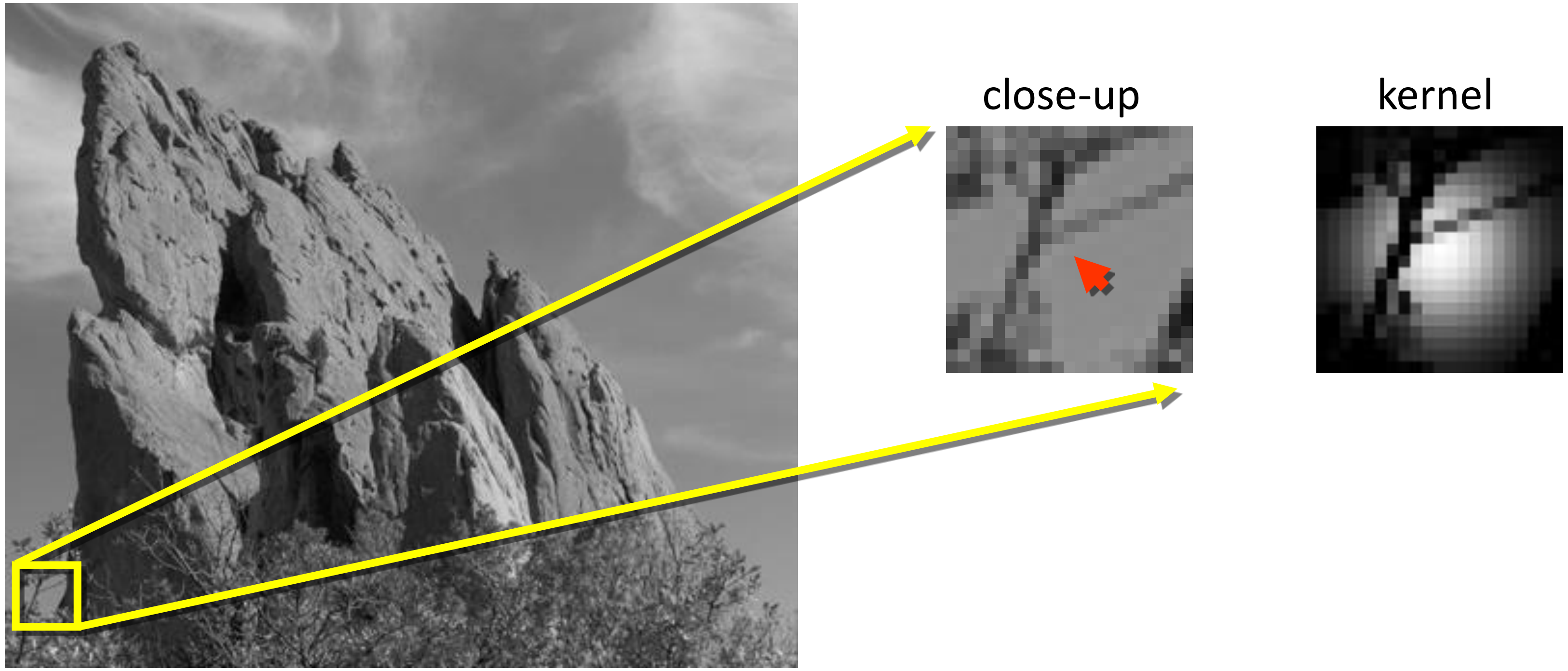
Depends on the application. For instance:

- space parameter: proportional to image size
  - e.g., 2% of image diagonal
- range parameter: proportional to edge amplitude
  - e.g., mean or median of image gradients
- independent of resolution and exposure

A Few
More Advanced
Remarks

Bilateral Filter Crosses Thin Lines

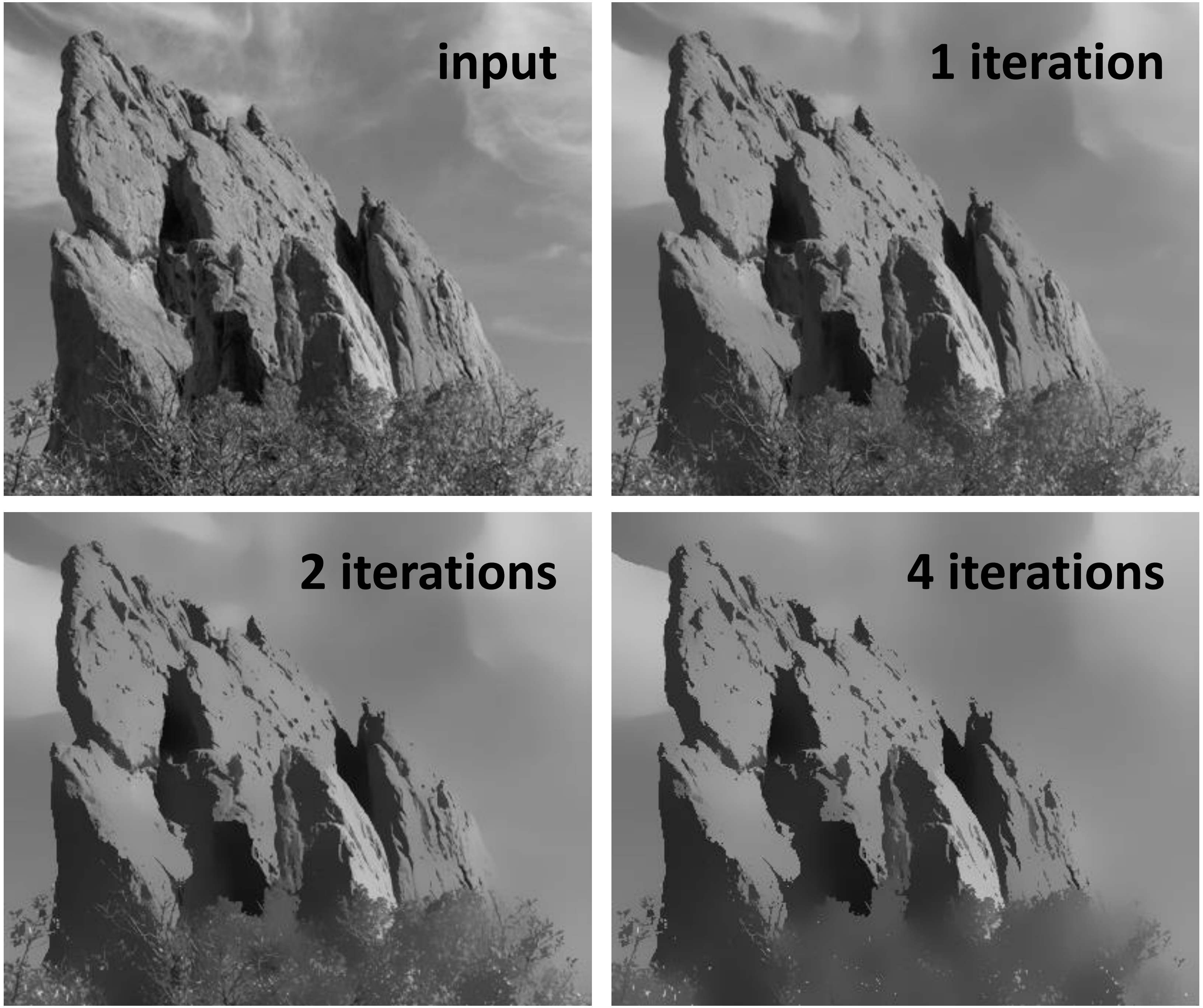
- Desirable for smoothing: more pixels = more robust
- Different from diffusion that stops at thin lines
- Bilateral filter averages across features thinner than ~2σ_s



Iterating the Bilateral Filter

$$I_{(n+1)} = BF\left[I_{(n)}\right]$$

- Generate more piecewise-flat images
- Often not needed in computational photo.


input
1 iteration
2 iterations
4 iterations

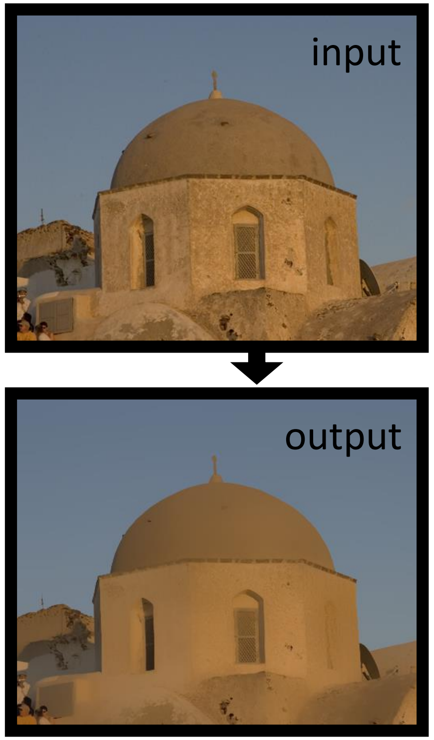
Bilateral Filtering Color Images

For gray-level images

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} G_{\sigma_s}(\| \mathbf{p} - \mathbf{q} \|)\, G_{\sigma_r}(\overbrace{| I_{\mathbf{p}} - I_{\mathbf{q}} |}^{\text{intensity difference}})\, \underbrace{I_{\mathbf{q}}}_{\text{scalar}}$$

For color images

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} G_{\sigma_s}(\| \mathbf{p} - \mathbf{q} \|)\, G_{\sigma_r}(\overbrace{\| \mathrm{C}_{\mathbf{p}} - \mathrm{C}_{\mathbf{q}} \|}^{\text{color difference}})\, \underbrace{\mathrm{C}_{\mathbf{q}}}_{\text{3D vector (RGB, Lab)}}$$



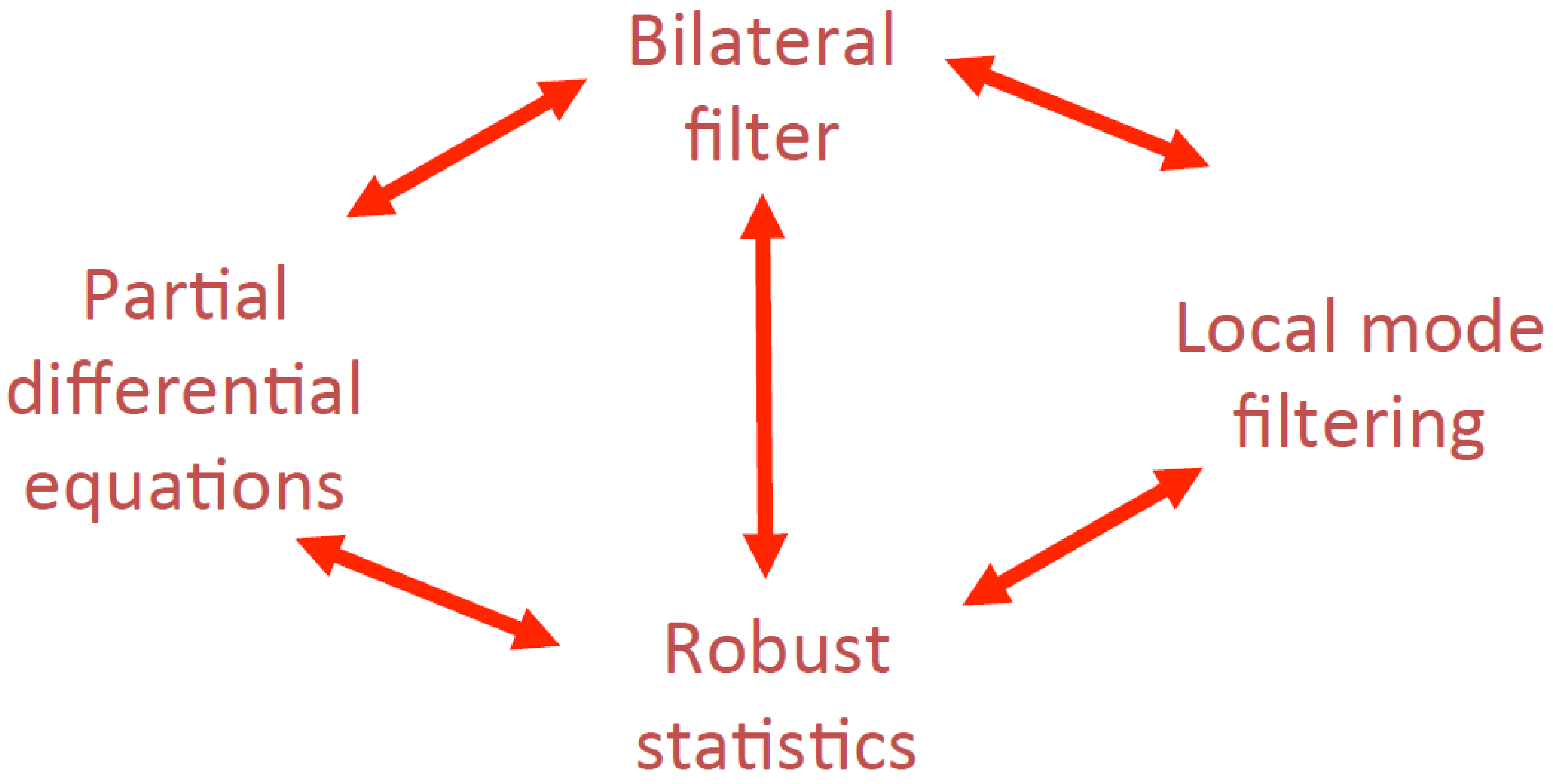
The bilateral filter is extremely easy to adapt to your need.

Hard to Compute

- Nonlinear $$BF[I]_{\mathbf{p}} = \boxed{\frac{1}{W_{\mathbf{p}}}} \sum_{\mathbf{q} \in S} G_{\sigma_s}(\| \mathbf{p} - \mathbf{q} \|) \, \boxed{G_{\sigma_r}(| I_{\mathbf{p}} - I_{\mathbf{q}} |)} \, I_{\mathbf{q}}$$
- Complex, spatially varying kernels
  - Cannot be precomputed, no FFT…

- Brute-force implementation is slow > 10min

Goal: Understand how does bilateral filter relates with other methods



Additional Reading: *Generalised Nonlocal Image Smoothing*, L. Pizarro, P. Mrazek, S. Didas, S. Grewenig and J. Weickert, IJCV, 2010


Any questions?

References - courses

- "A Gentle Introduction to Bilateral Filtering and its Applications" given by **Sylvain Paris**, Pierre Kornprobst, Jack Tumblin, and Frédo Durand (http://people.csail.mit.edu/sparis/bf_course/)
- Bilateral Filtering, and Non-local Means Denoising, **Erkut Erdem**

References

- Siggraph 15: An L1 Image Transform for Edge-Preserving Smoothing and Scene-Level Intrinsic Decomposition. [code]
- Siggraph 14: Bilateral texture filtering
- Siggraph 14: Fast Local Laplacian Filters: Theory and Applications. [code]