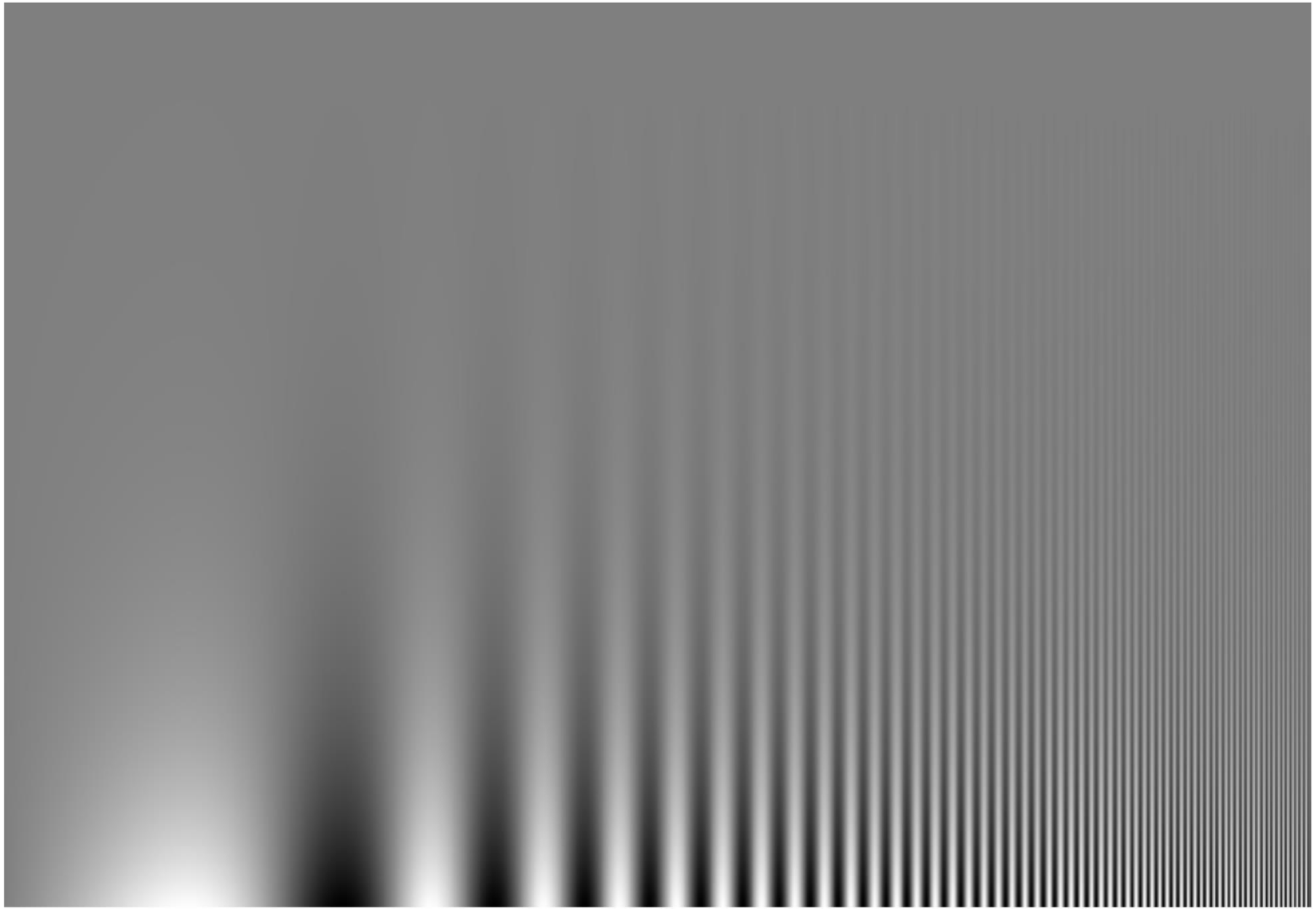
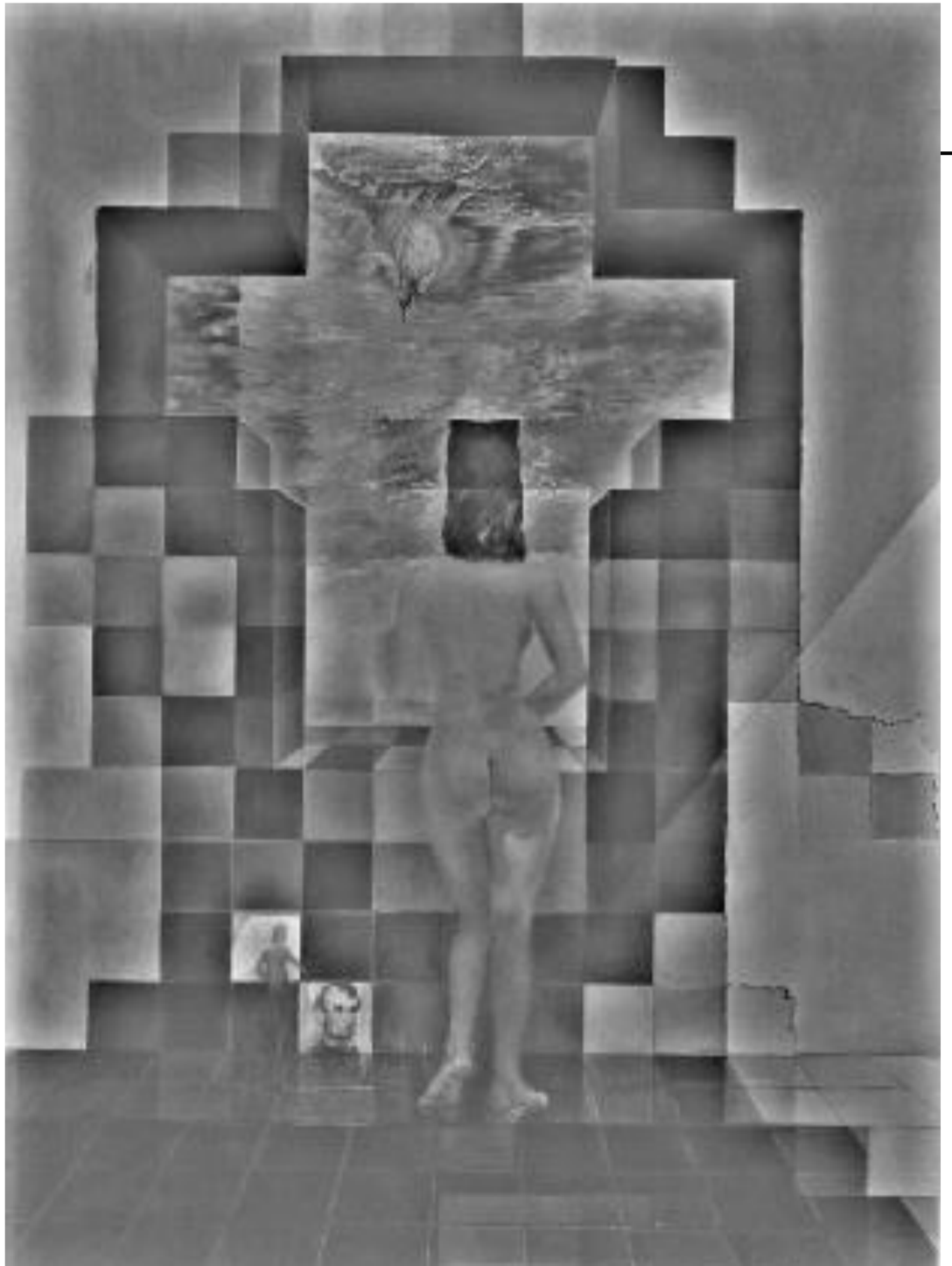
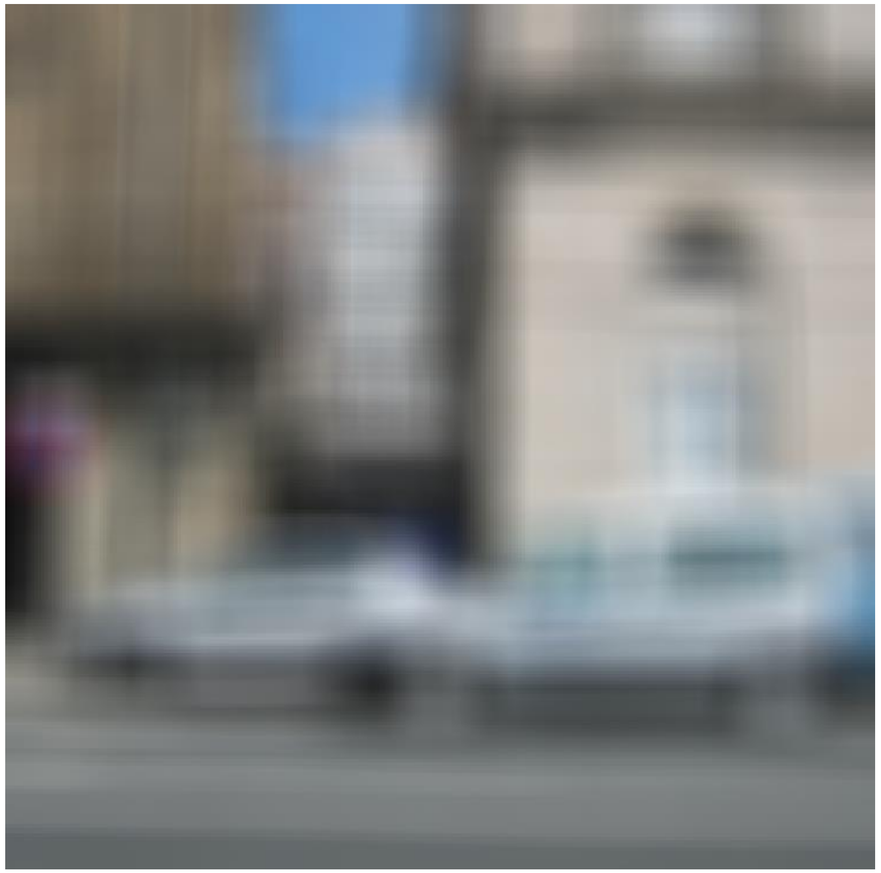
Spatial & Temporal Filtering



Salvador Dali
“Gala Contemplating the Mediterranean Sea, which at 30 meters becomes the portrait of Abraham Lincoln”, 1976





Low-pass filters

High/Band-pass filters

Low pass-filters

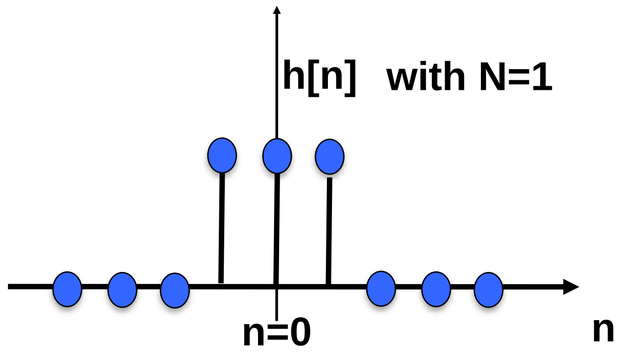
Box filter

2N+1

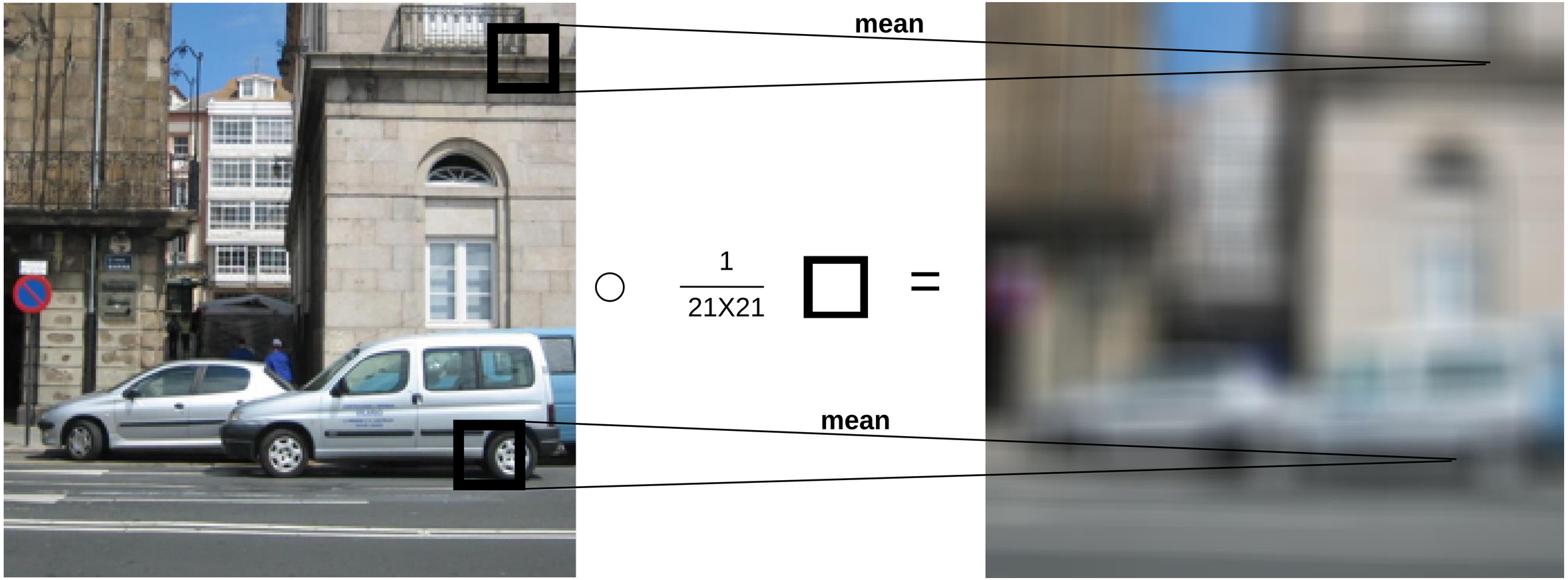
1	1	…	1
1	1		1
1	1		1
…			
1	1	1	1

2M+1

$$h_{N,M}[n,m] = \begin{cases} 1 & \text{if } -N \le n \le N \text{ and } -M \le m \le M \\ 0 & \text{otherwise} \end{cases}$$



Box filter



256X256

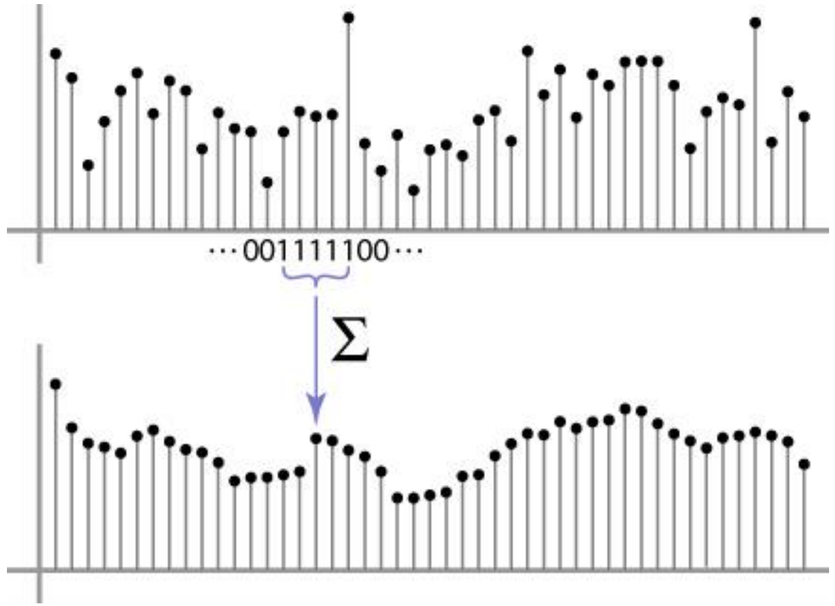
256X256

What does it do?

- Replaces each pixel with an average of its neighborhood
- Achieve smoothing effect (remove sharp features)

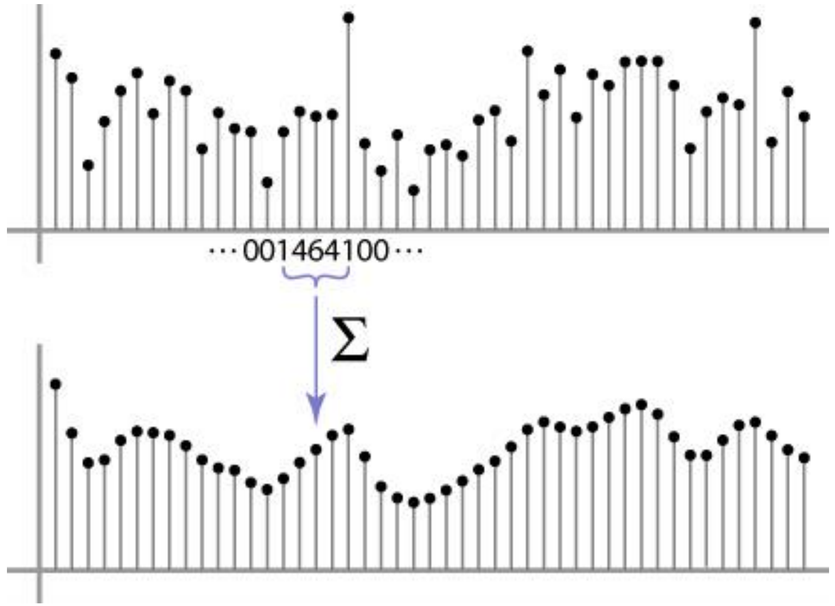
Moving Average

- Can add weights to our moving average
- *Weights* […, 0, 1, 1, 1, 1, 1, 0, …] / 5



Weighted Moving Average

- bell curve (gaussian-like) weights [..., 1, 4, 6, 4, 1, ...]



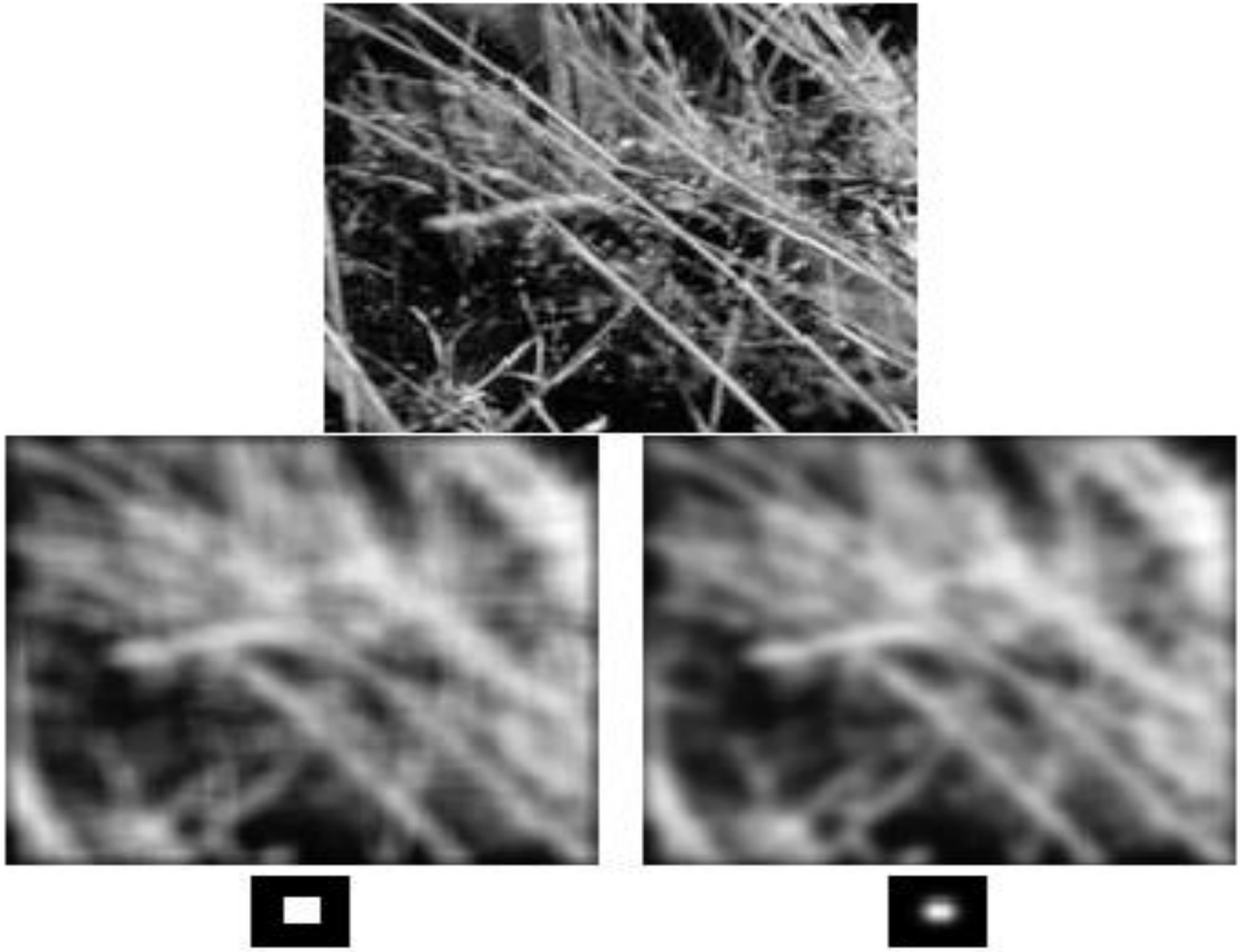
Why Box is bad

$$[1, 1, 1] \circ [\ldots, 1, -1, 1, -1, 1, -1, \ldots] = [\ldots, -1, 1, -1, 1, -1, 1, \ldots]$$

$$[1, 2, 1] \circ [\ldots, 1, -1, 1, -1, 1, -1, \ldots] = [\ldots, 0, 0, 0, 0, 0, 0, \ldots]$$

12

Mean vs. Gaussian filtering

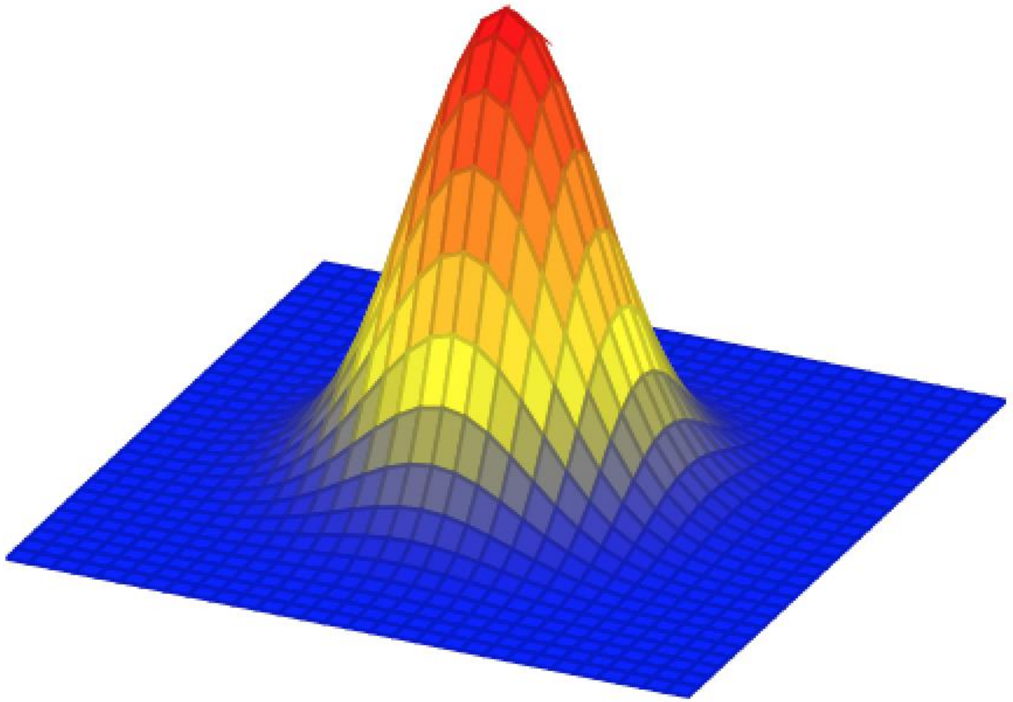


Slide by Steve Seitz

Gaussian filter

In the continuous domain:

$$g(x, y; \sigma) = \frac{1}{2\pi\sigma^2} \exp -\frac{x^2 + y^2}{2\sigma^2}$$



Gaussian filter

$$g(x, y; \sigma) = \frac{1}{2\pi\sigma^2} \exp -\frac{x^2 + y^2}{2\sigma^2}$$

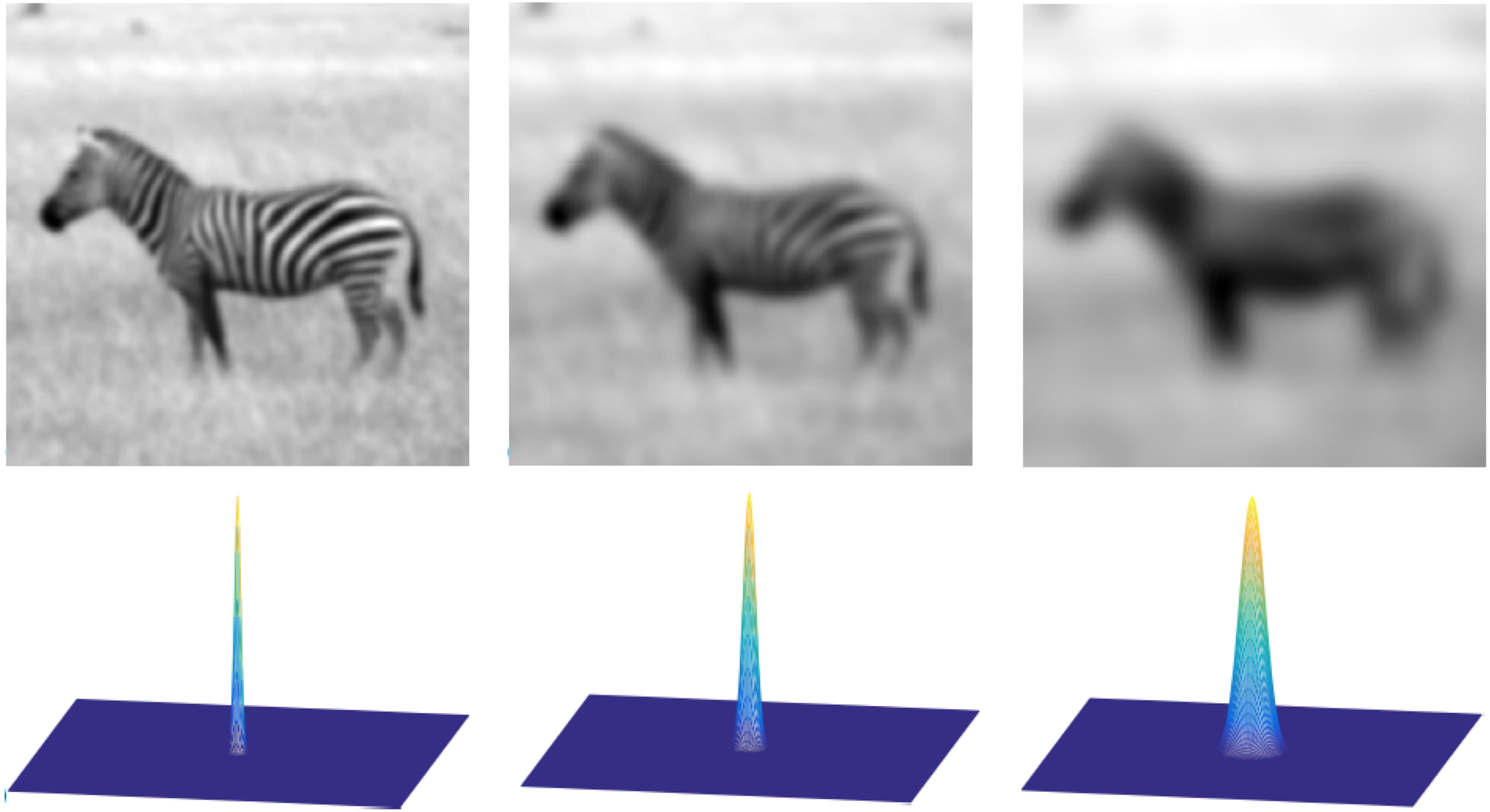
Discretization of the Gaussian:

At 3σ the amplitude of the Gaussian is around 1% of its central value

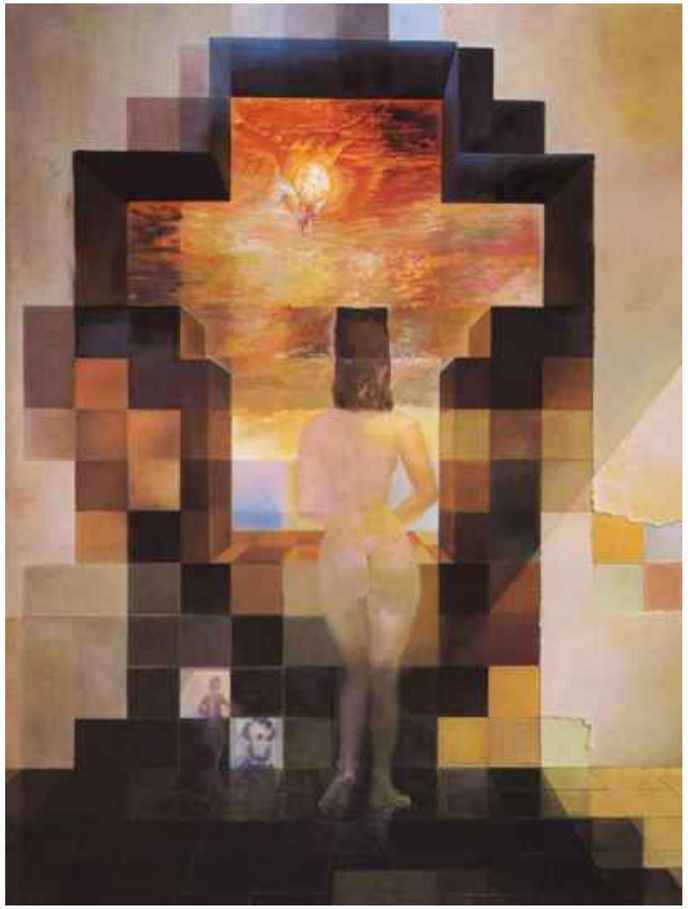
$$g[m, n; \sigma] = \exp -\frac{m^2 + n^2}{2\sigma^2}$$

Scale

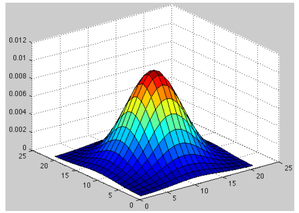
$$g[m,n;\sigma] = \exp -\frac{m^2+n^2}{2\sigma^2}$$



Gaussian filter for low-pass filtering



Dali



Binomial filter

Binomial coefficients provide a compact approximation of the gaussian coefficients using only integers.

The simplest blur filter (low pass) is

$$[1 \quad 1]$$

Binomial filters in the family of filters obtained as successive convolutions of [1 1]

Binomial filter

$$b_1 = [1\ \ 1]$$

$$b_2 = [1\ 1] \circ [1\ 1] = [1\ 2\ 1]$$

$$b_3 = [1\ 1] \circ [1\ 1] \circ [1\ 1] = [1\ 3\ 3\ 1]$$

Binomial filter

b_1							1		1							$\sigma_1^2 = 1/4$
b_2						1		2		1						$\sigma_2^2 = 1/2$
b_3					1		3		3		1					$\sigma_3^2 = 3/4$
b_4				1		4		6		4		1				$\sigma_4^2 = 1$
b_5			1		5		10		10		5		1			$\sigma_5^2 = 5/4$
b_6		1		6		15		20		15		6		1		$\sigma_6^2 = 3/2$
b_7	1		7		21		35		35		21		7		1	$\sigma_7^2 = 7/4$
b_8	1	8		28		56		70		56		28		8	1	$\sigma_8^2 = 2$

Properties of binomial filters

- Sum of the values is 2^n
- The variance of b_n is $\sigma^2 = n/4$
- The convolution of two binomial filters is also a binomial filter

$$b_n \circ b_m = b_{n+m}$$

With a variance:

$$\sigma_n^2 + \sigma_m^2 = \sigma_{n+m}^2$$

These properties are analogous to the gaussian property in the continuous domain (but the binomial filter is different than a discretization of a gaussian)

B2[n]

$$b_{2,2} = b_{2,0} \circ b_{0,2} = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \circ \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

High pass-filters

[-1 1]

$$\frac{\partial \mathbf{I}}{\partial x} \simeq \mathbf{I}(x,y) - \mathbf{I}(x-1,y)$$

$\otimes$ [-1, 1] =

h[m,n]

g[m,n]

f[m,n]

$[-1\ 1]^T$

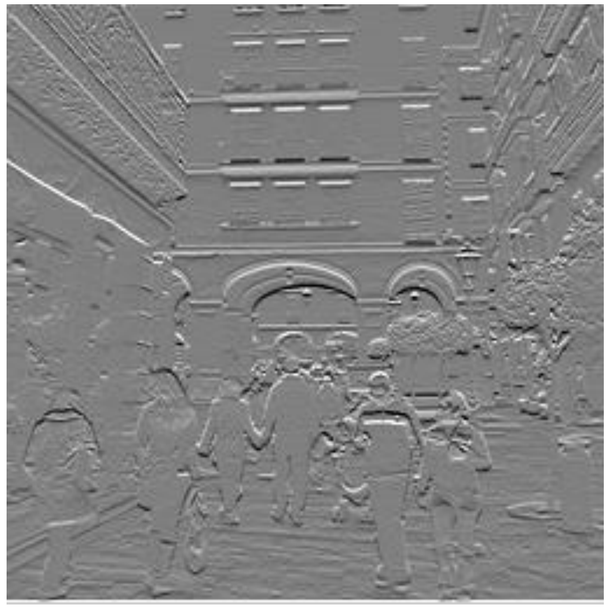
g[m,n]

$\otimes$

$[-1, 1]^T$

h[m,n]

=



f[m,n]

Discrete derivatives

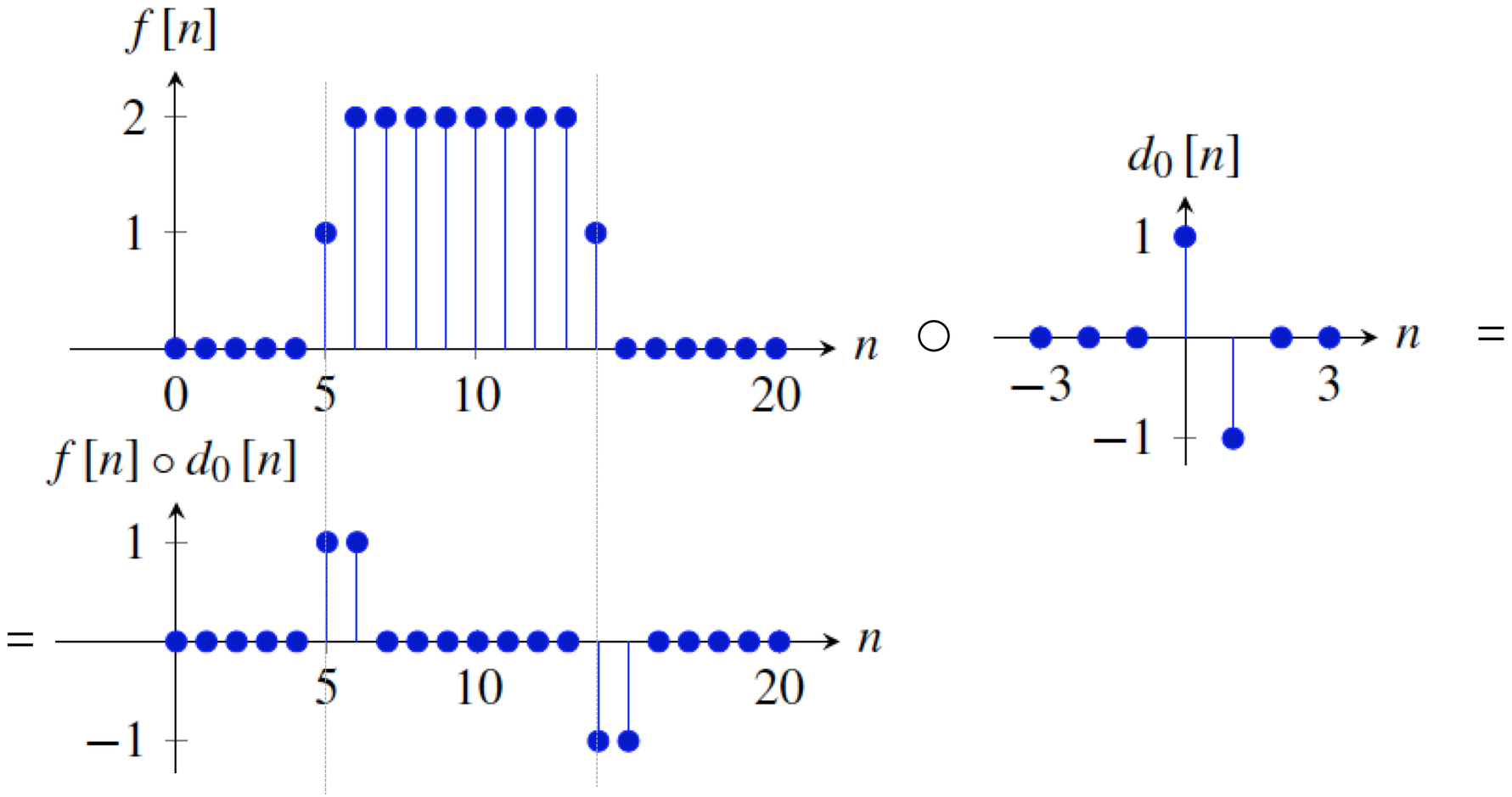
$$d_0 = [1, -1]$$

$$f \circ d_0 = f[n] - f[n-1]$$

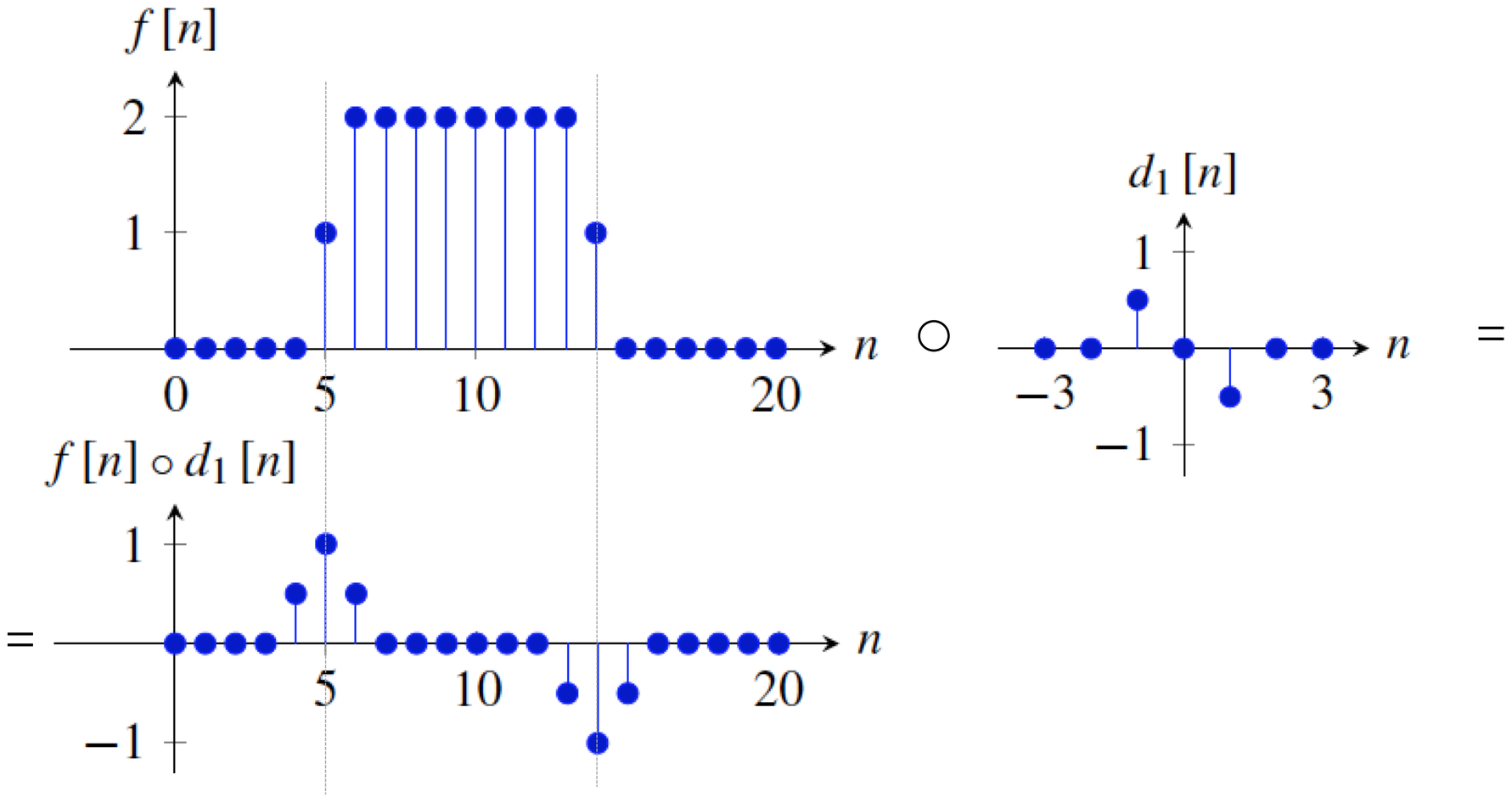
$$d_1 = [1, 0, -1]/2$$

$$f \circ d_1 = \frac{f[n+1] - f[n-1]}{2}$$

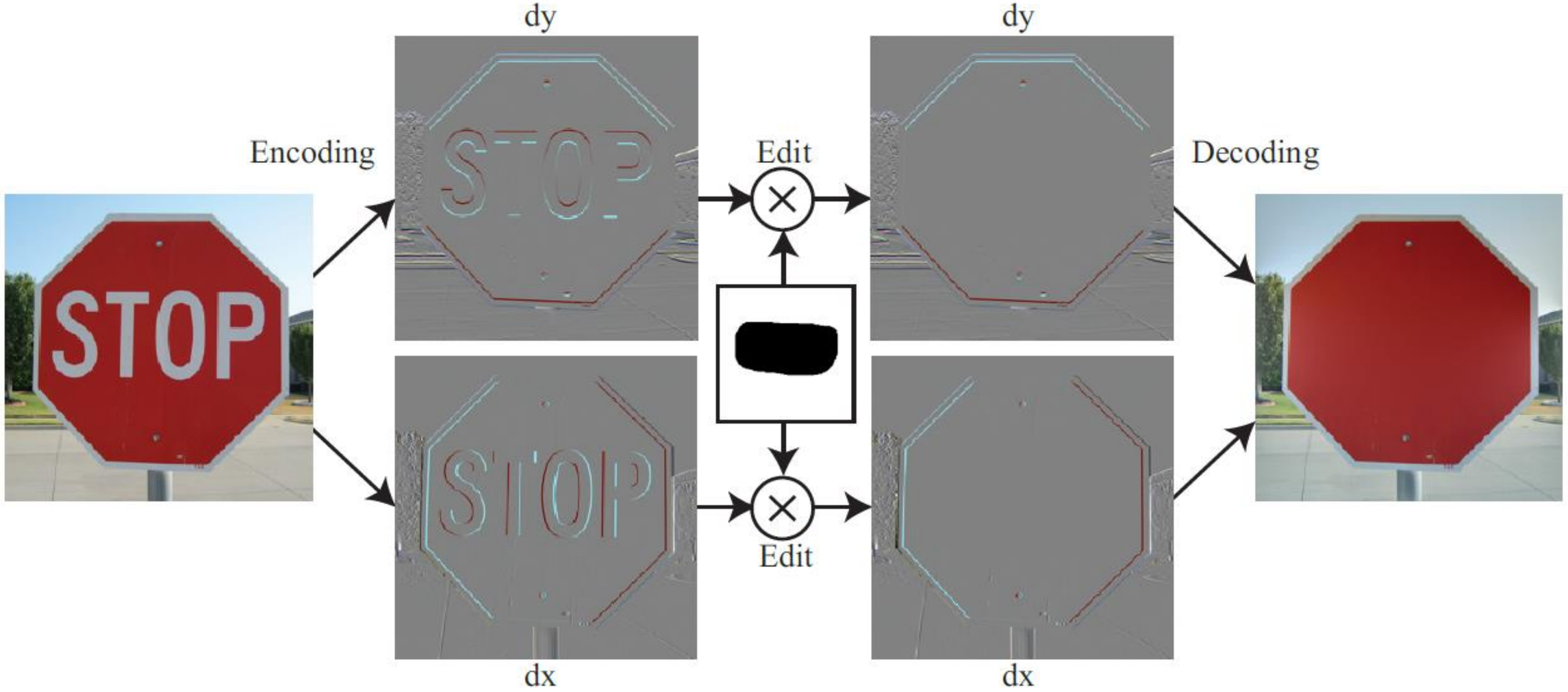
Discrete derivatives



Discrete derivatives



Gradient Domain



Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features

Motivation for gradient-domain filtering?

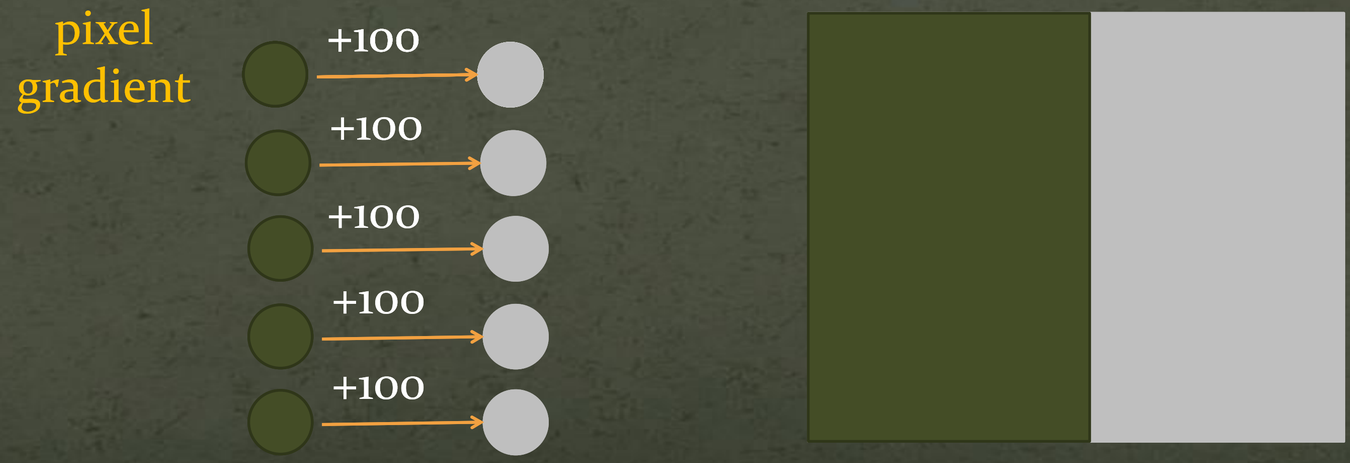
- Can be used to exert high-level control over images
  - gradients – low level image-features

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features

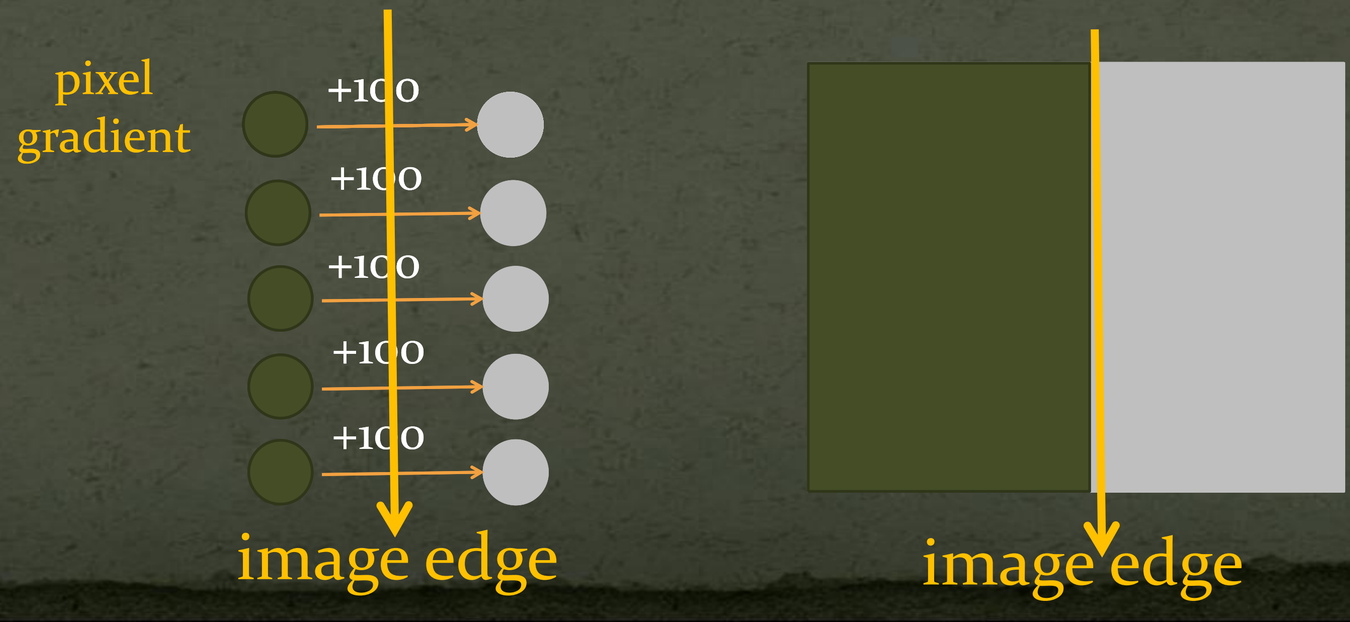
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features



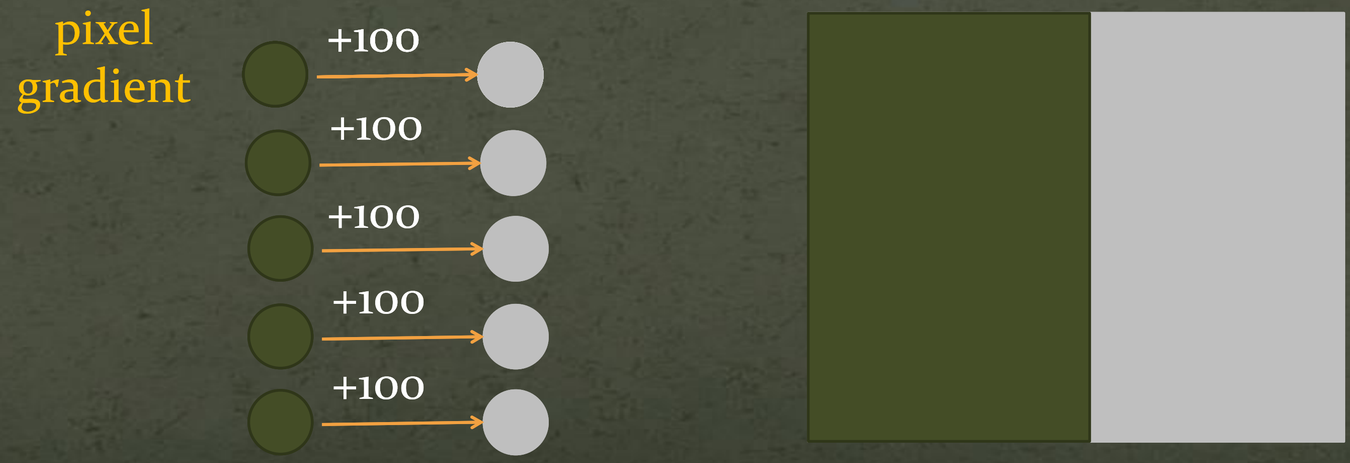
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features



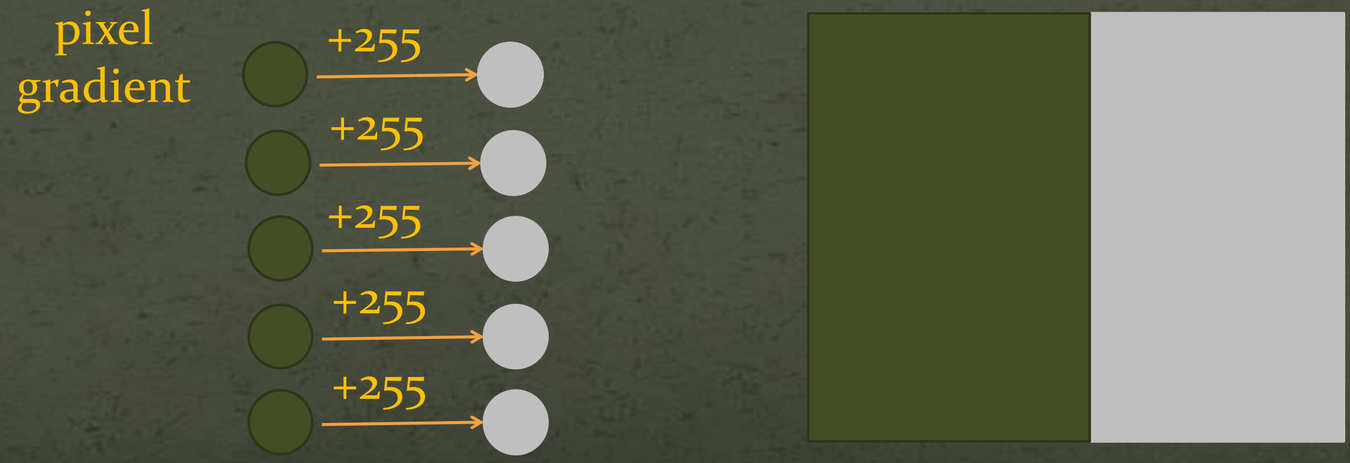
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features
  - manipulate local gradients to manipulate global image interpretation



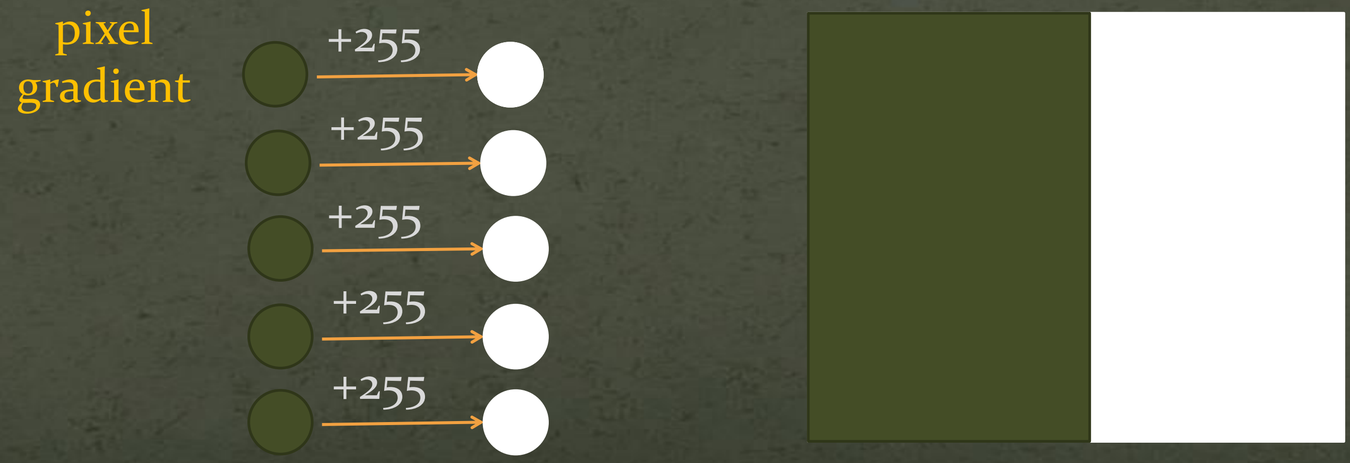
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features
  - manipulate local gradients to manipulate global image interpretation



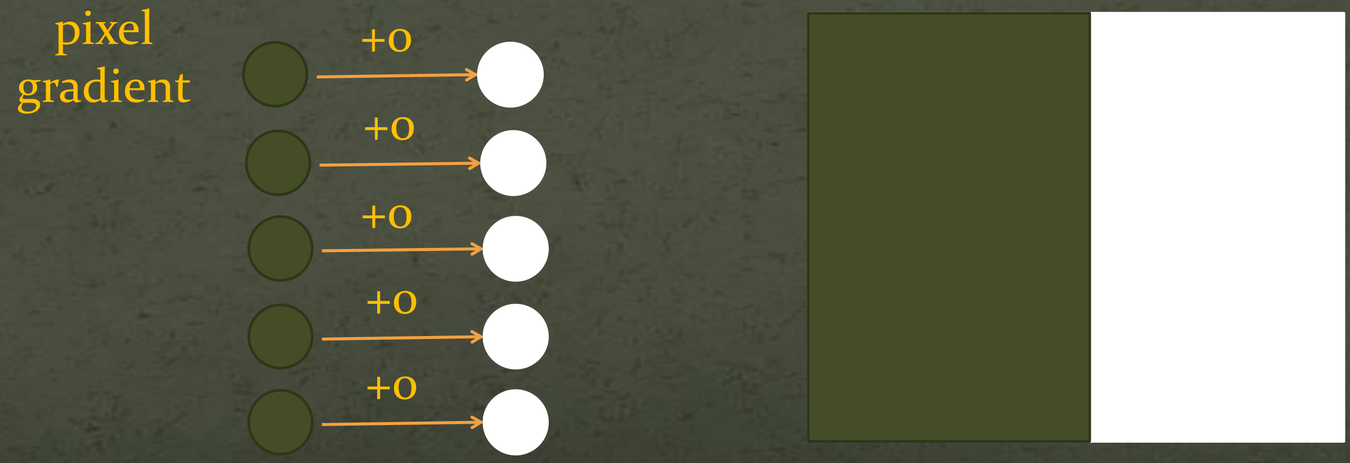
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features
  - manipulate local gradients to manipulate global image interpretation



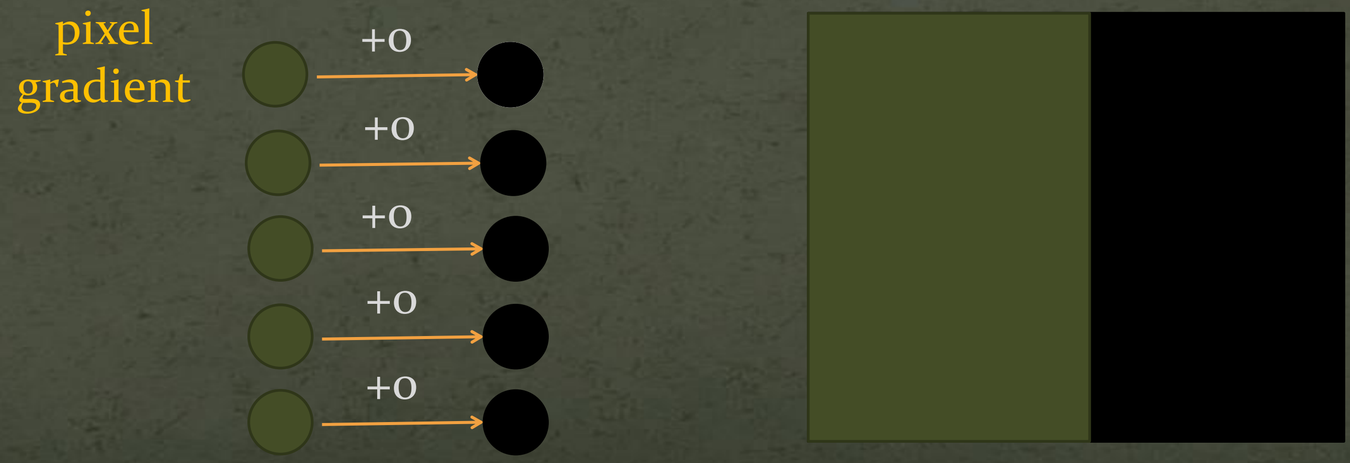
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features
  - manipulate local gradients to manipulate global image interpretation

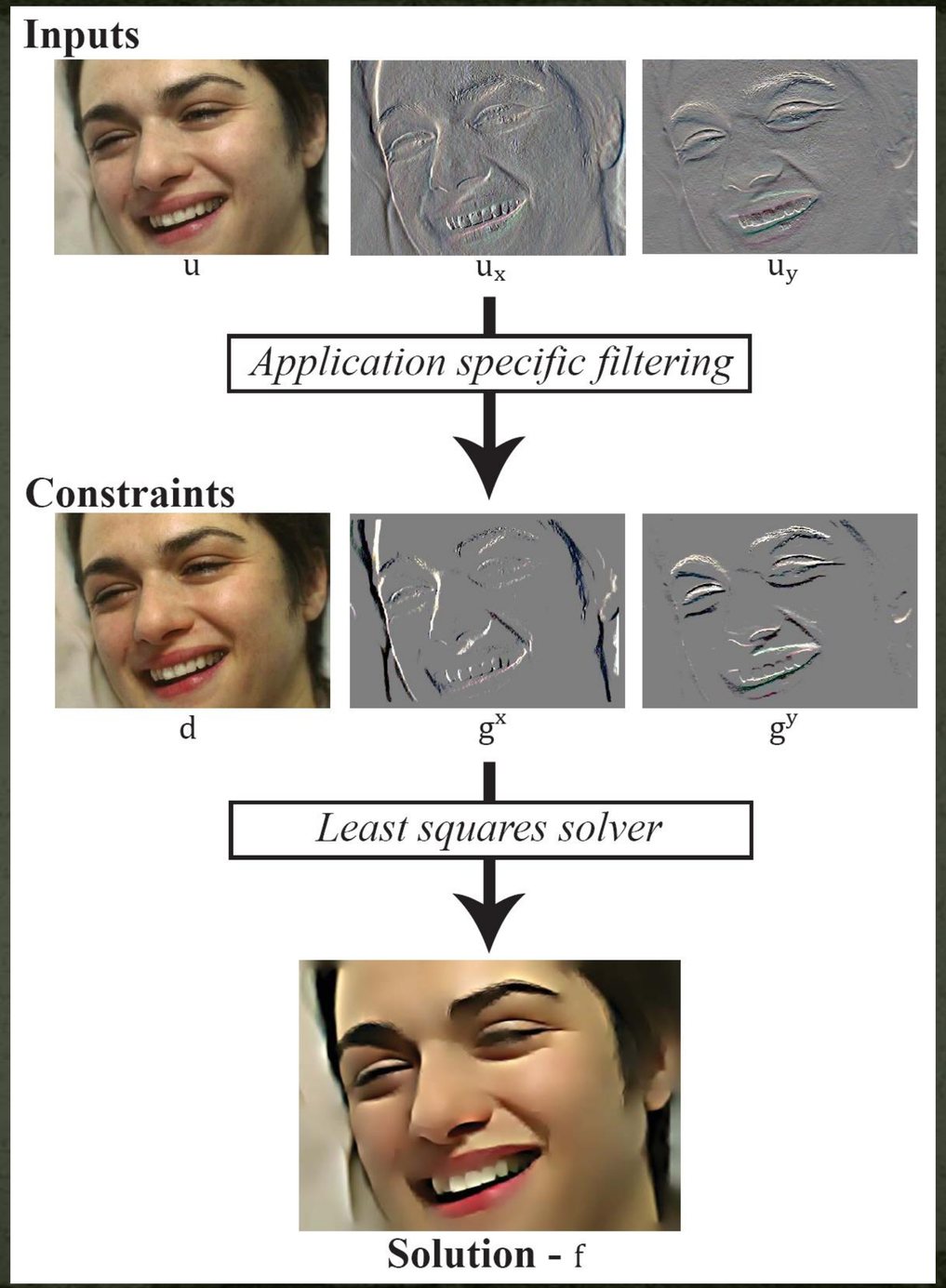


Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features
  - manipulate local gradients to manipulate global image interpretation

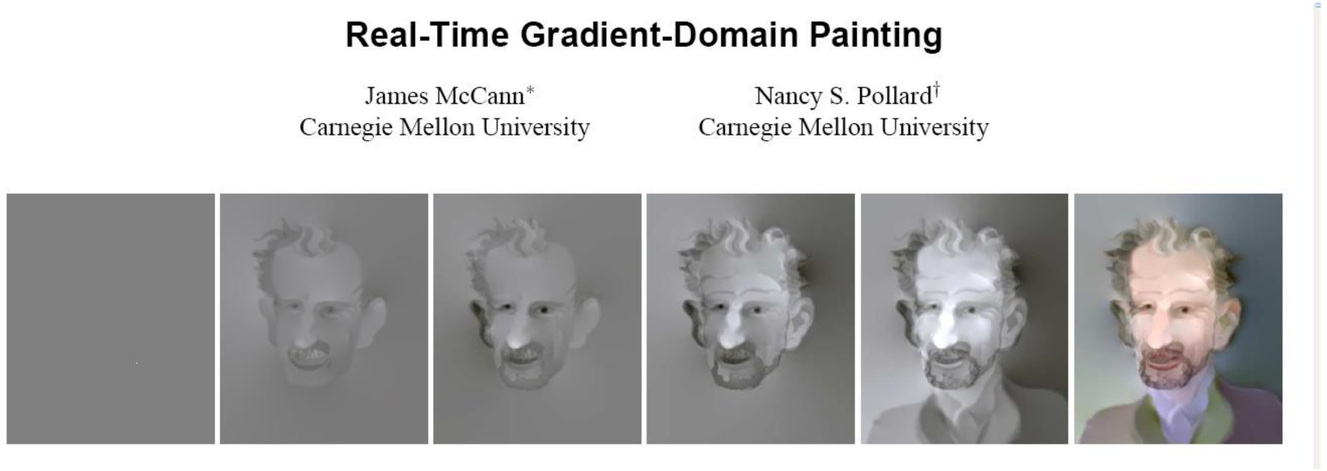


Gradient Domain Editing



Perez et al., “Poisson Image Editing”, 2003

Drawing in Gradient Domain

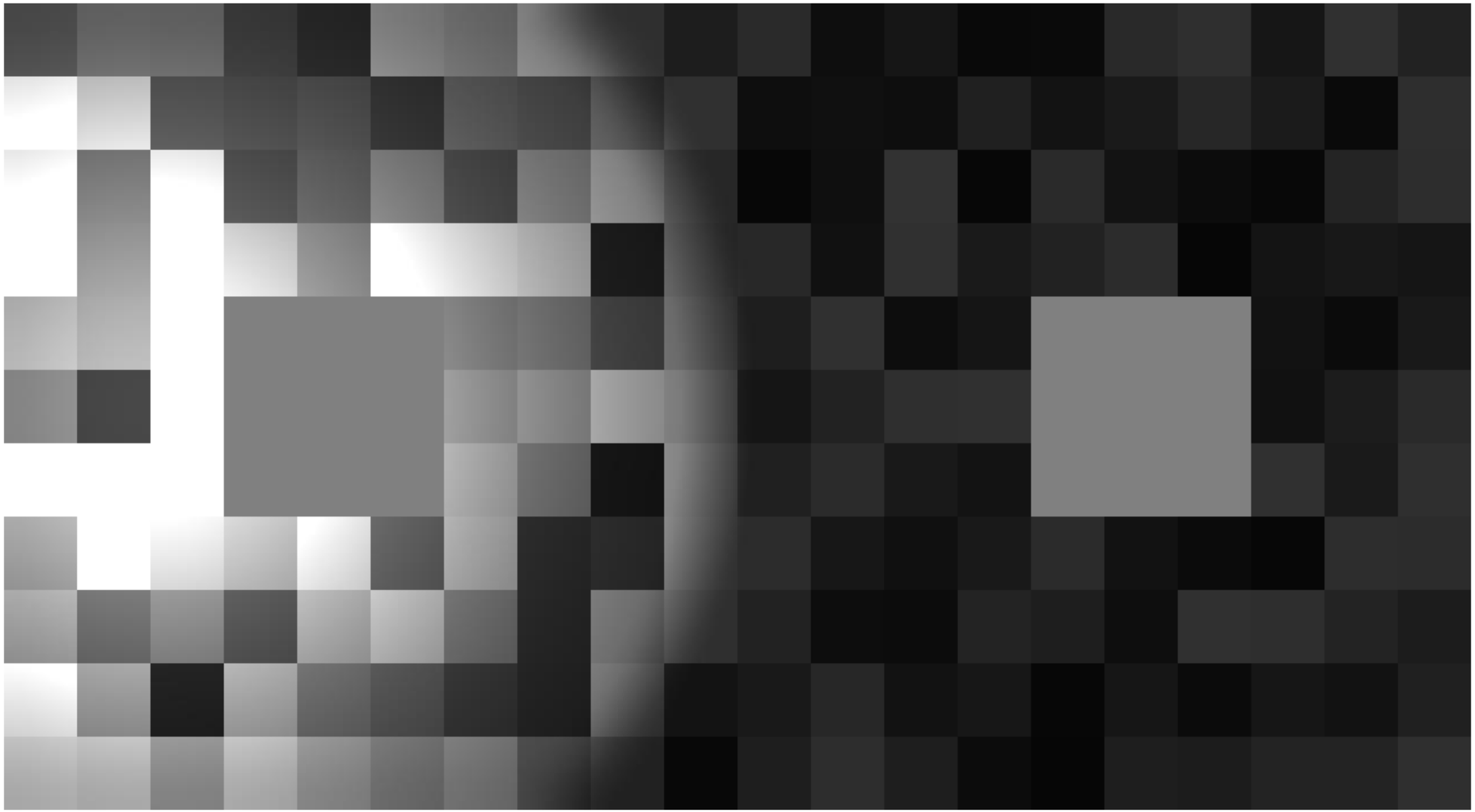


James McCann & Nancy Pollard
Real-Time Gradient-Domain Painting, SIGGRAPH 2009

(paper came out of this class!)

http://www.youtube.com/watch?v=RvhkAfrA0-w&feature=youtu.be

Lightness Constancy



Simultaneous Contrast Illusion

Retinex (Land and McCann)

Intrinsic Images:

- *Image = reflectance image * illumination image*

$$\log \ell(x,y) = \log r(x,y) + \log l(x,y) \tag{18.38}$$

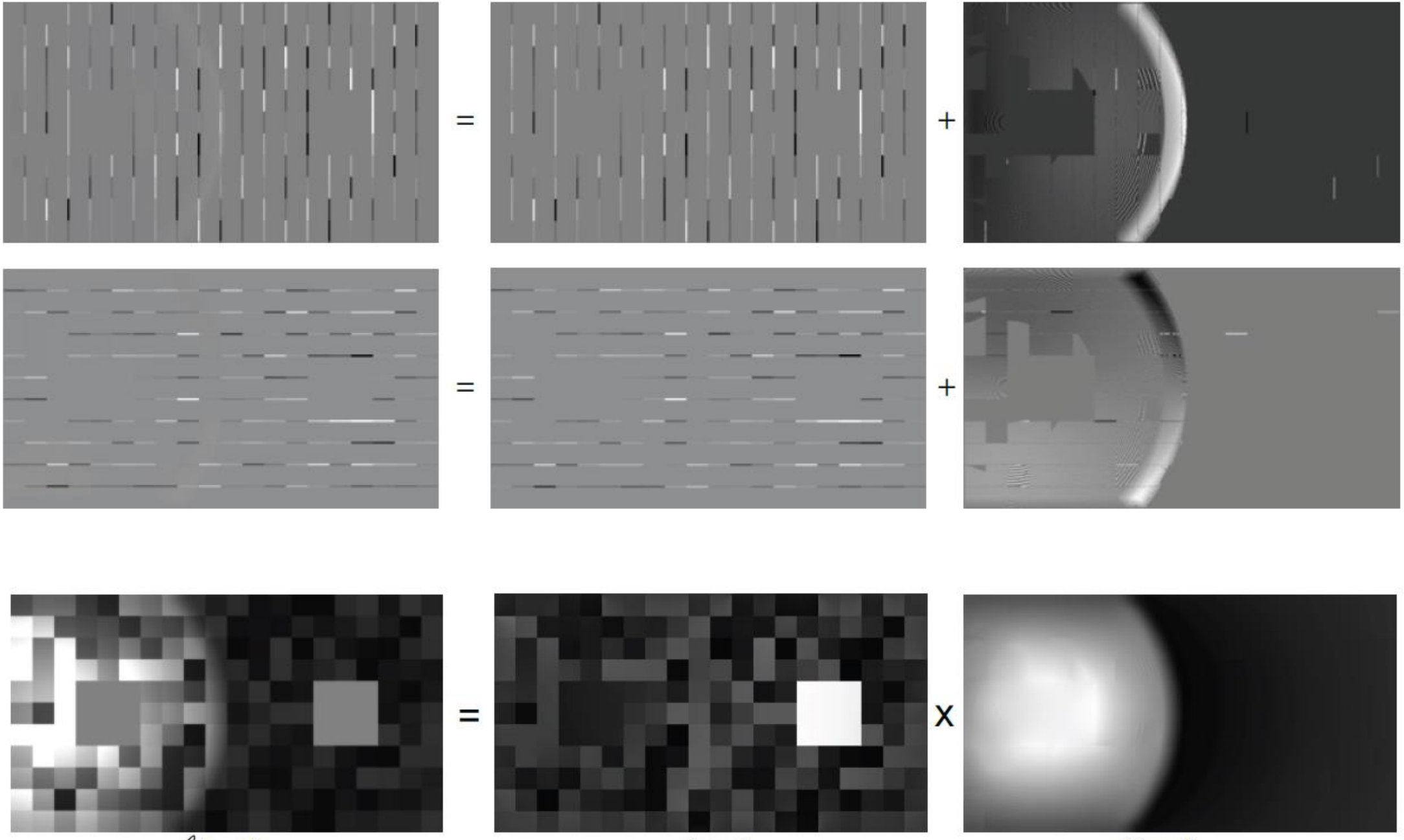
Taking derivatives along x and y is now simple:

$$\frac{\partial \log \ell(x,y)}{\partial x} = \frac{\partial \log r(x,y)}{\partial x} + \frac{\partial \log l(x,y)}{\partial_x} \tag{18.39}$$

And the same thing is done for the derivative along y.

Any derivative larger that the threshold is assigned to the derivative of the reflectance image $r(x,y)$ and the ones smaller than a threshold are assigned to the illumination image $l(x,y)$:

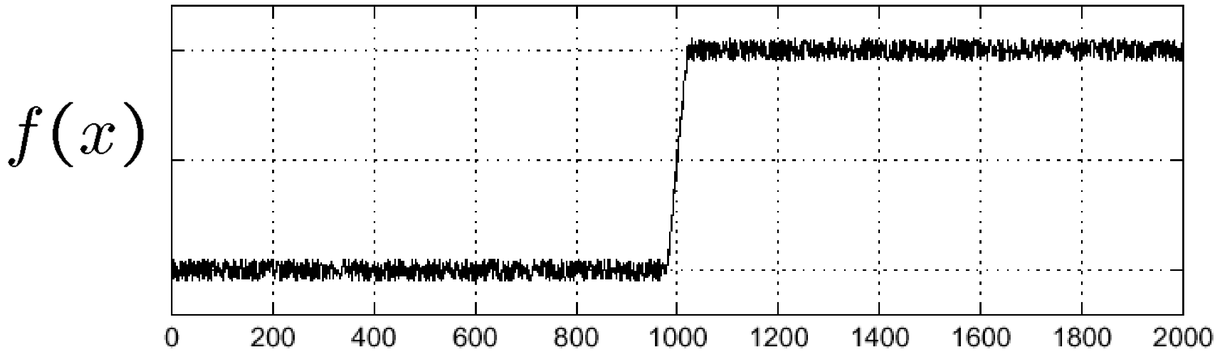
$$\frac{\partial \log r(x,y)}{\partial x} = \begin{cases} \frac{\partial \log \ell(x,y)}{\partial x} & \text{if } \left| \frac{\partial \log \ell(x,y)}{\partial x} \right| > T \\ 0 & \text{otherwise} \end{cases} \tag{18.40}$$

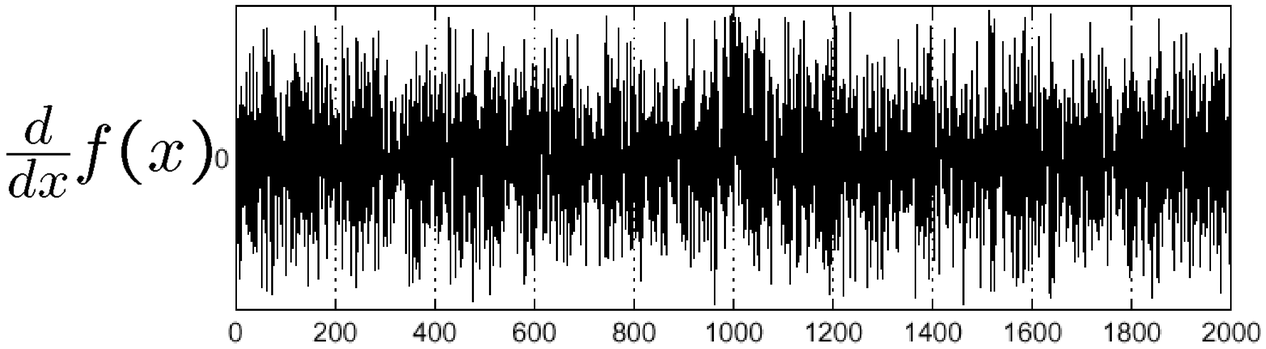

=
+
=
+
=
x
$\ell(x, y)$
$r(x, y)$
$l(x, y)$

Effects of noise

Consider a single row or column of the image

- Plotting intensity as a function of position gives a signal

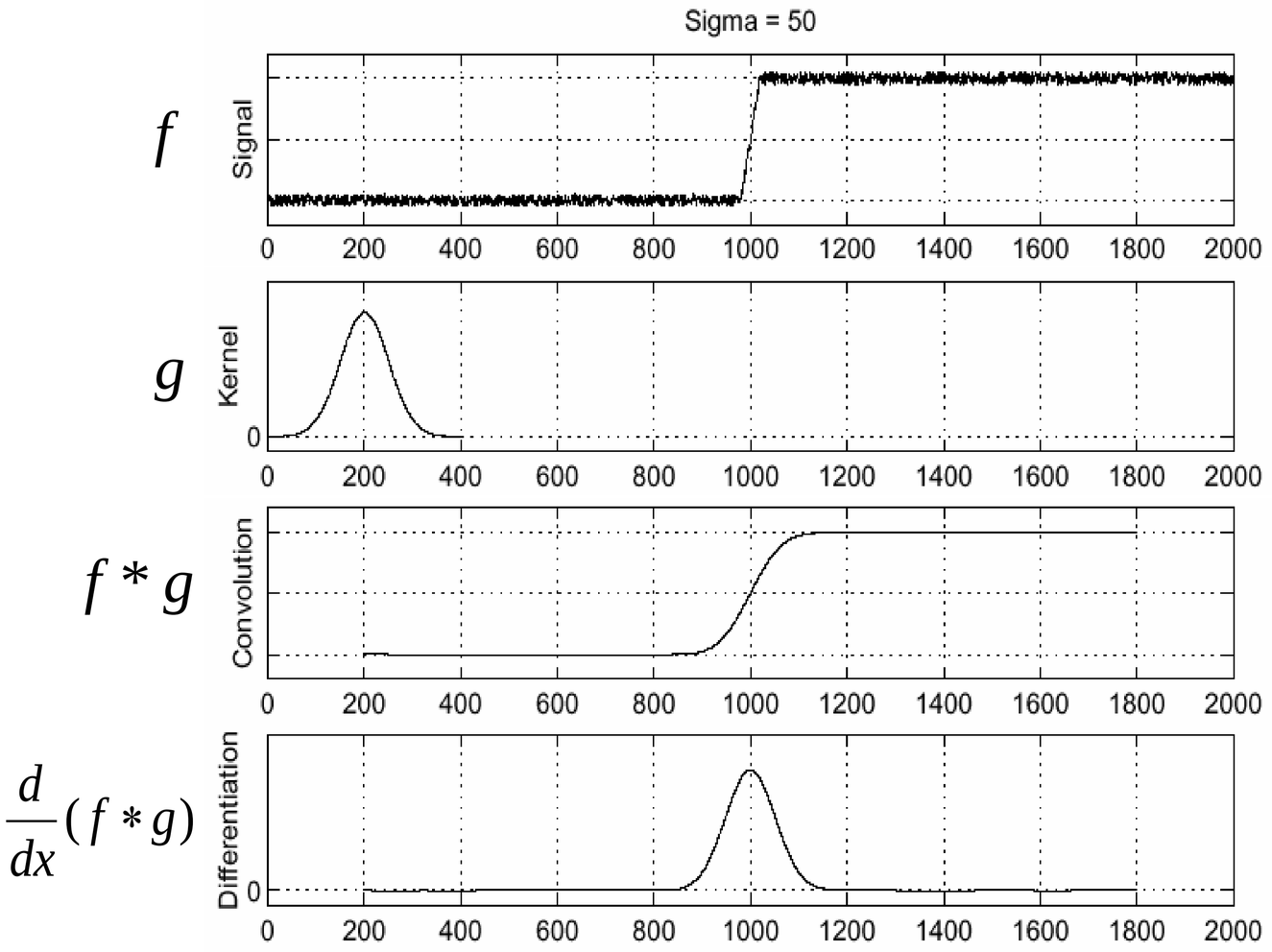




Where is the edge?

Source: S. Seitz

Solution: smooth first



- To find edges, look for peaks in $\frac{d}{dx}(f * g)$

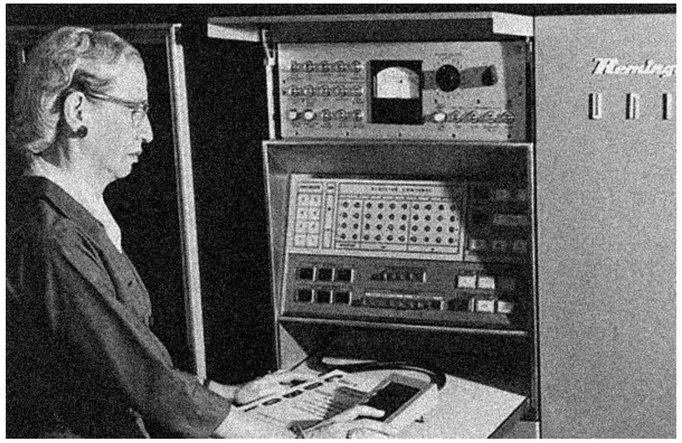
Source: S. Seitz

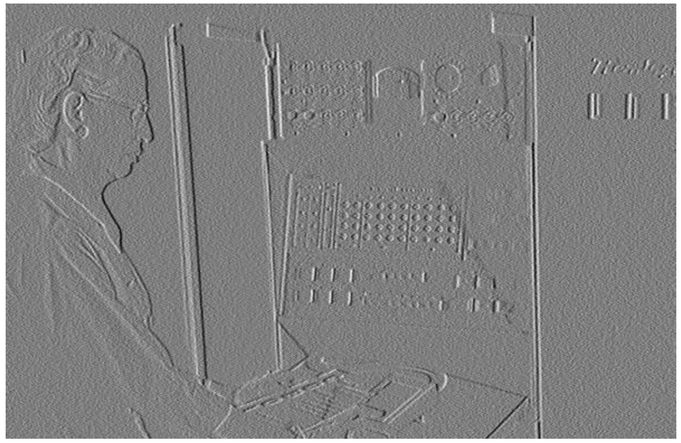
Noise in 2D

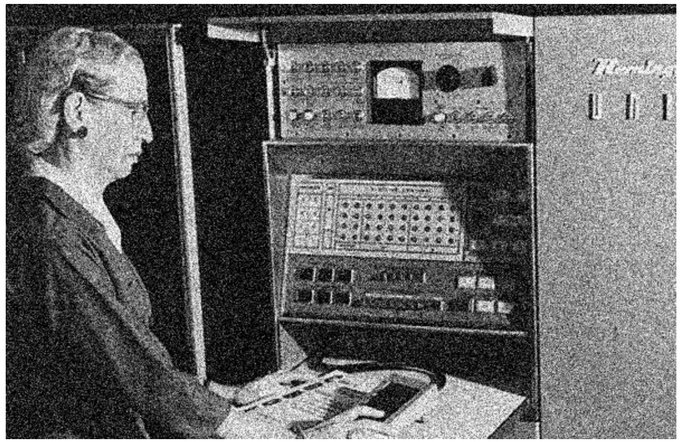
Noisy Input

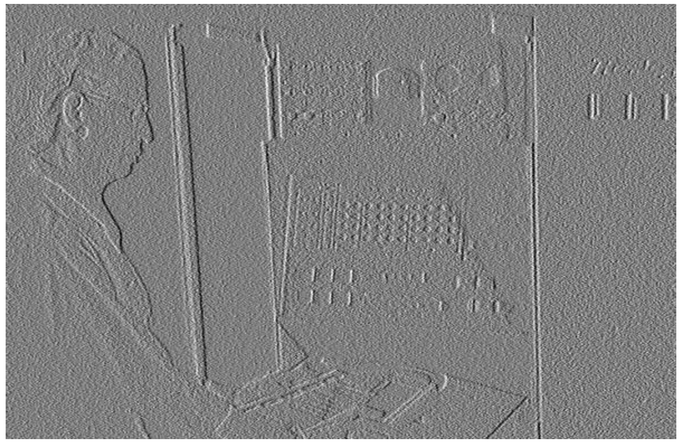
Ix via [-1,01]

Zoom









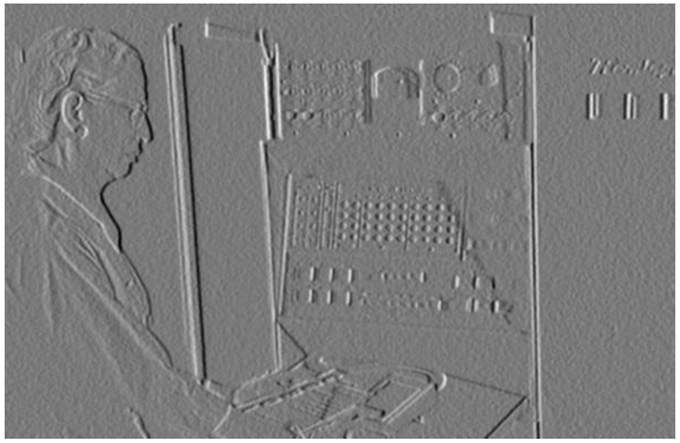
Source: D. Fouhey

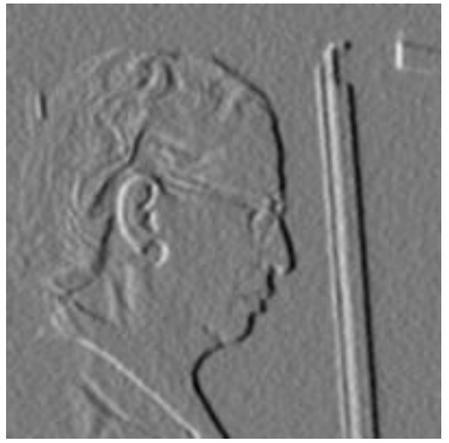
Noise + Smoothing

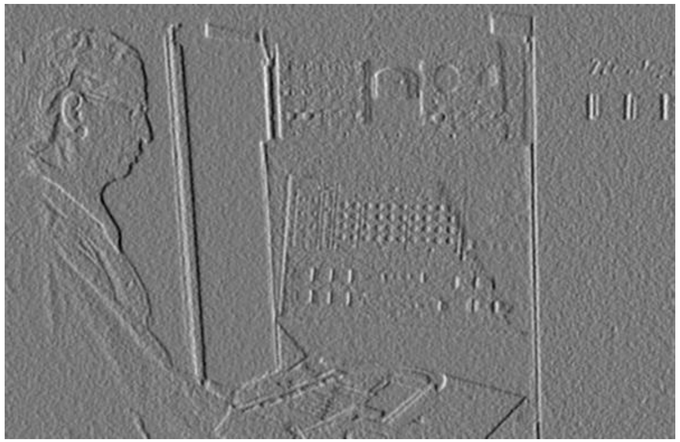
Smoothed Input

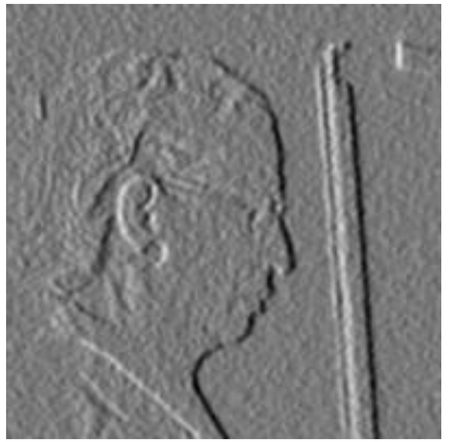
Ix via [-1,01]

Zoom



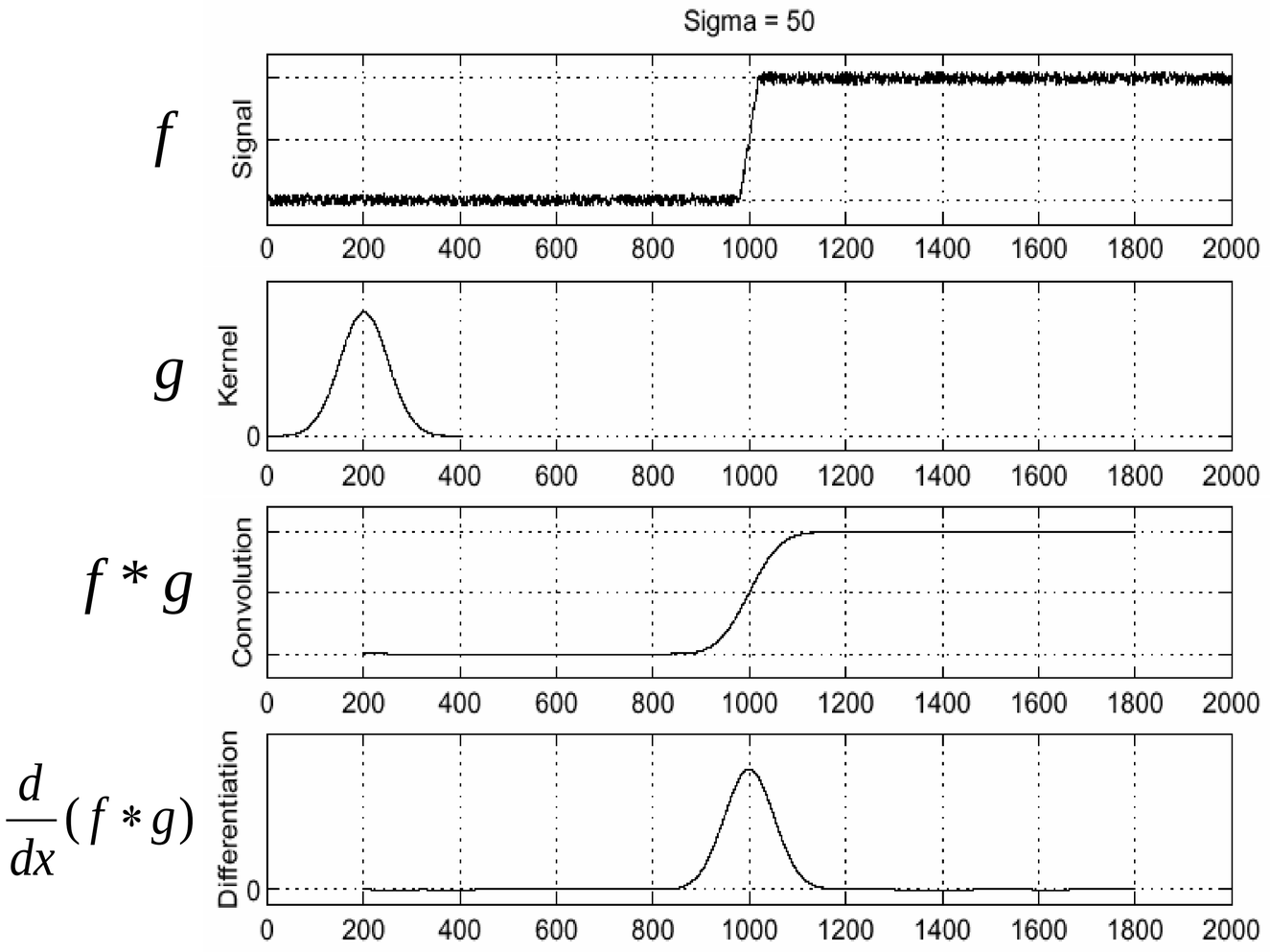






Source: D. Fouhey

How many convolutions here?



f

g

$f * g$

$\frac{d}{dx}(f * g)$

can we reduce this?

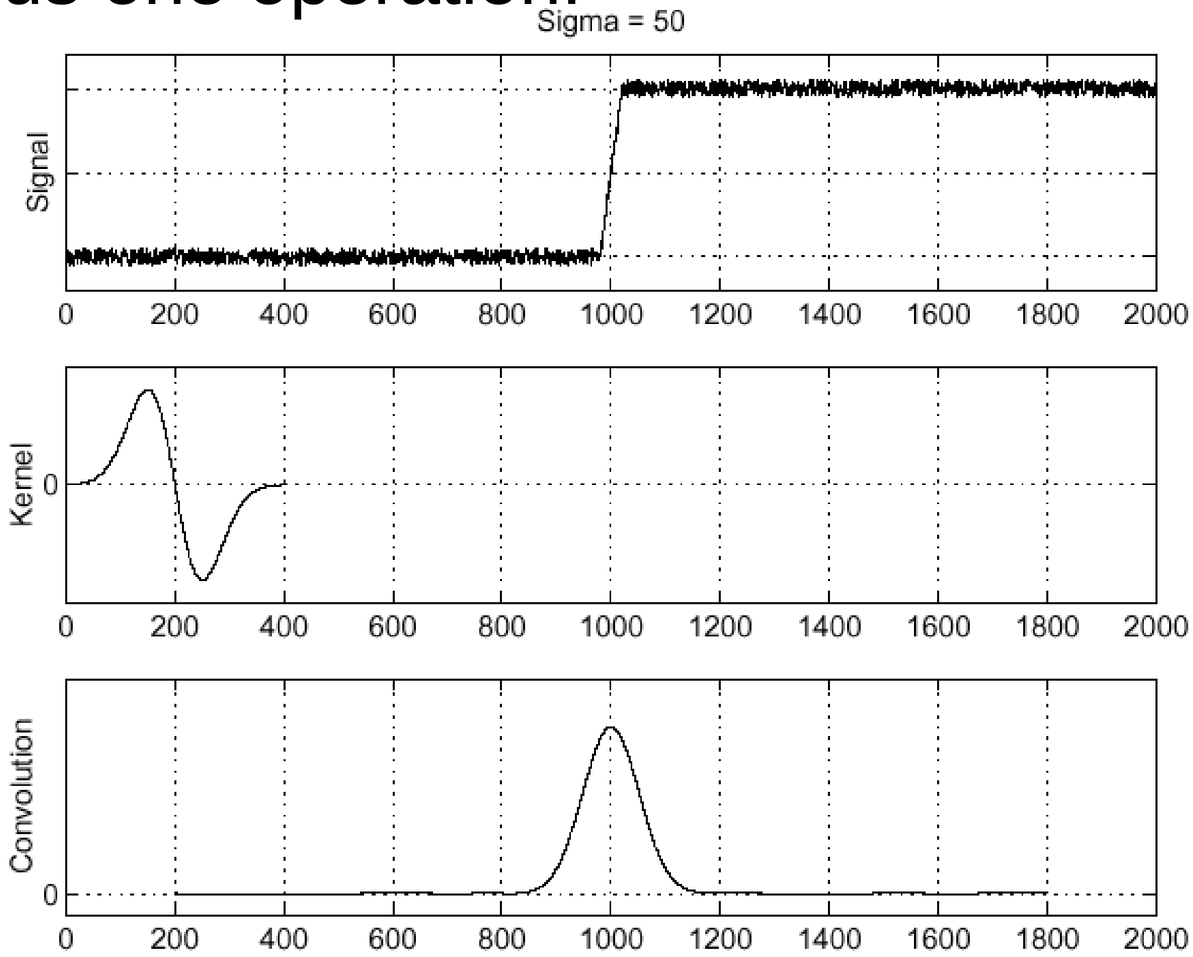
Derivative theorem of convolution

$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$$

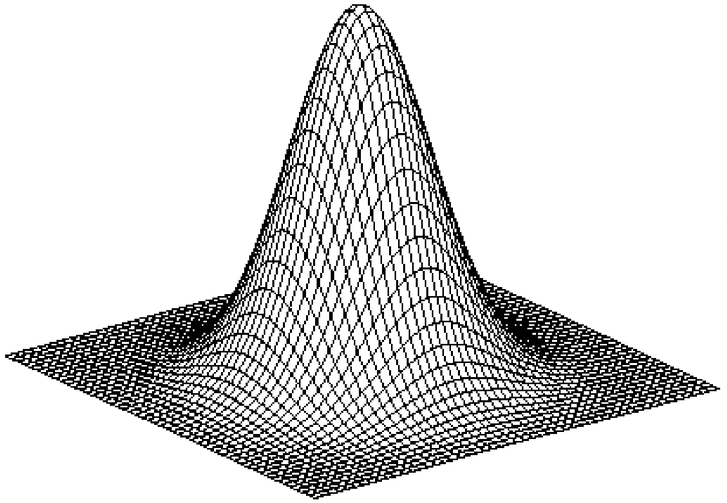
This saves us one operation:

$\frac{\partial}{\partial x}h$

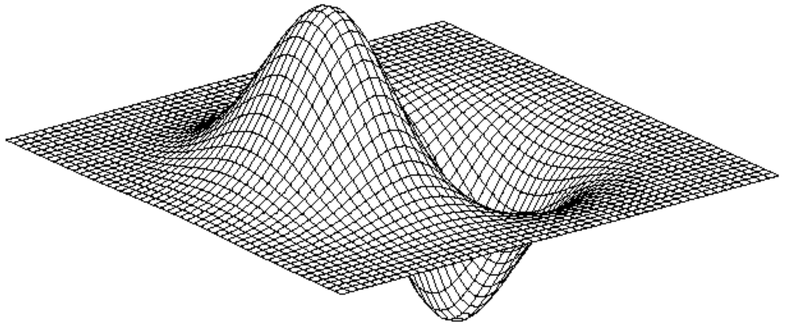
$(\frac{\partial}{\partial x}h) \star f$



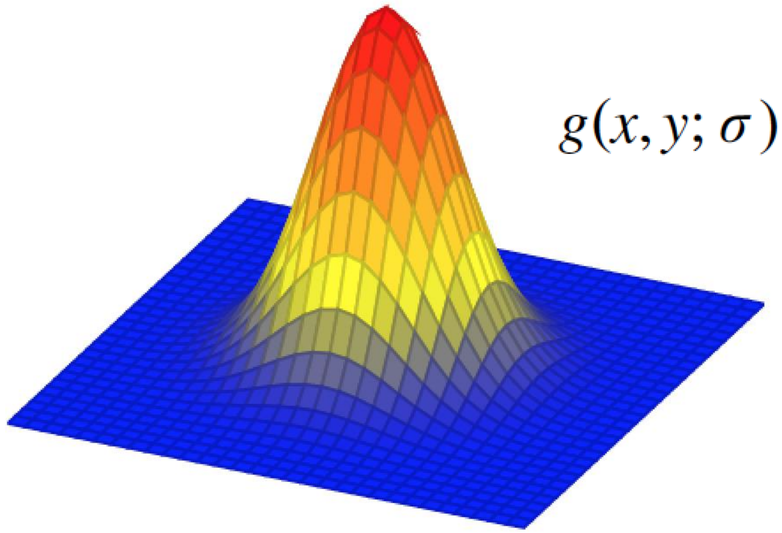
Derivative of Gaussian filter



* [1 0 -1] =



Gaussian derivatives



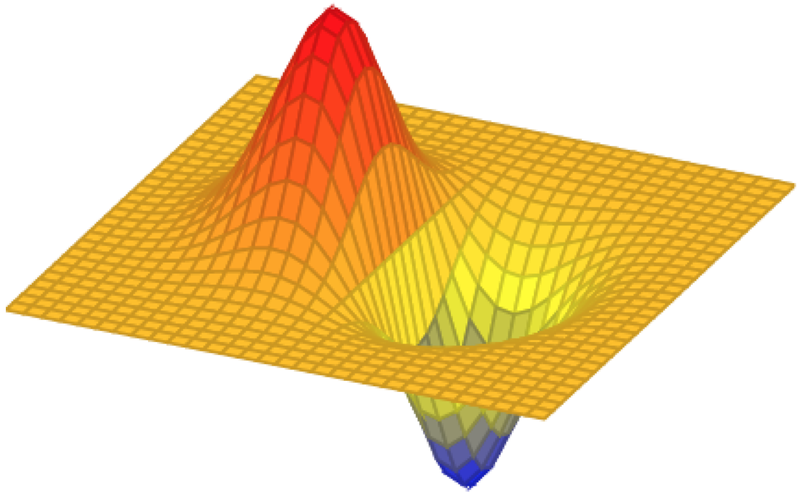
$$g(x, y; \sigma) = \frac{1}{2\pi\sigma^2} \exp -\frac{x^2 + y^2}{2\sigma^2}$$

The continuous derivative is:

$$g_x(x, y; \sigma) = \frac{\partial g(x, y; \sigma)}{\partial x} =$$

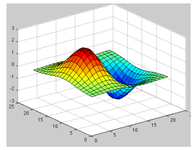
$$= \frac{-x}{2\pi\sigma^4} \exp -\frac{x^2 + y^2}{2\sigma^2}$$

$$= \frac{-x}{\sigma^2} g(x, y; \sigma)$$

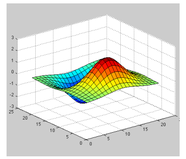


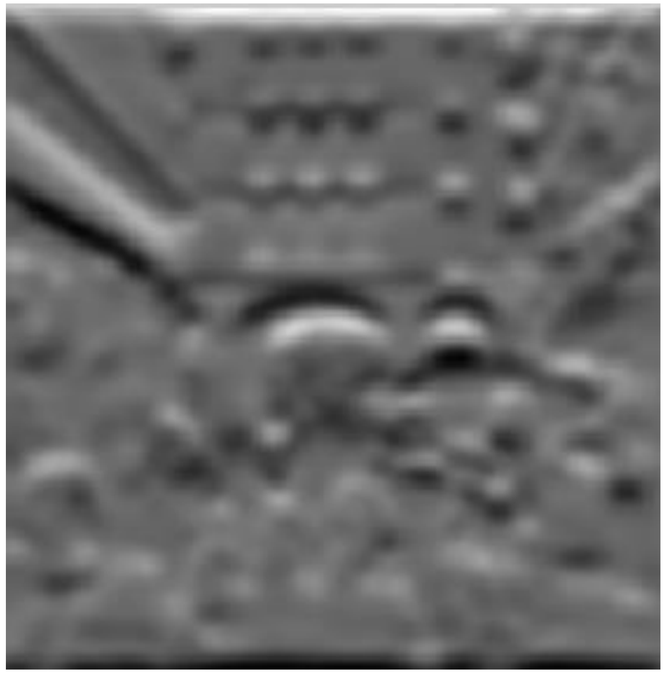
Orientation

$$g_x(x,y) = \frac{\partial g(x,y)}{\partial x} = \frac{-x}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



$$g_y(x,y) = \frac{\partial g(x,y)}{\partial y} = \frac{-y}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

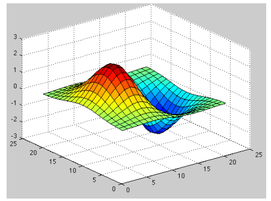




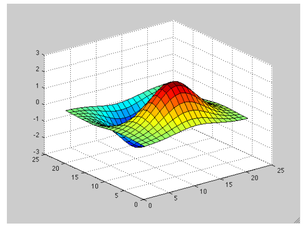
What about other orientations not axis aligned?

Orientation

$$g_x(x,y) = \frac{\partial g(x,y)}{\partial x} = \frac{-x}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



$$g_y(x,y) = \frac{\partial g(x,y)}{\partial y} = \frac{-y}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



The smoothed directional gradient is a linear combination of two kernels

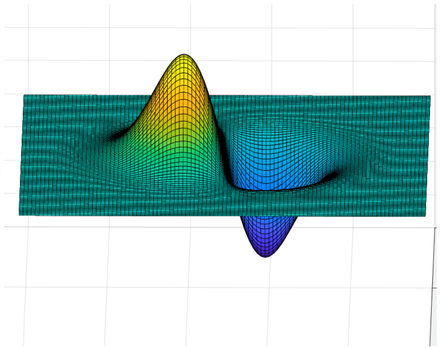
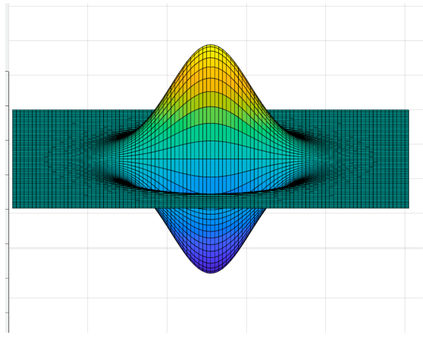
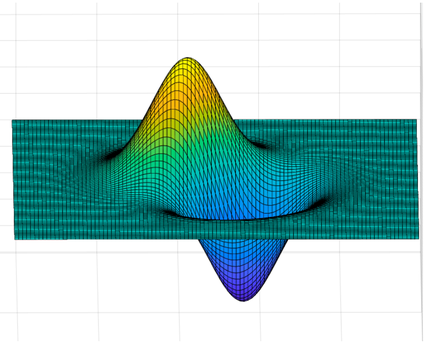
$$u^T \nabla g \otimes I = \left(\cos(\alpha) g_x(x,y) + \sin(\alpha) g_y(x,y)\right) \otimes I(x,y) =$$

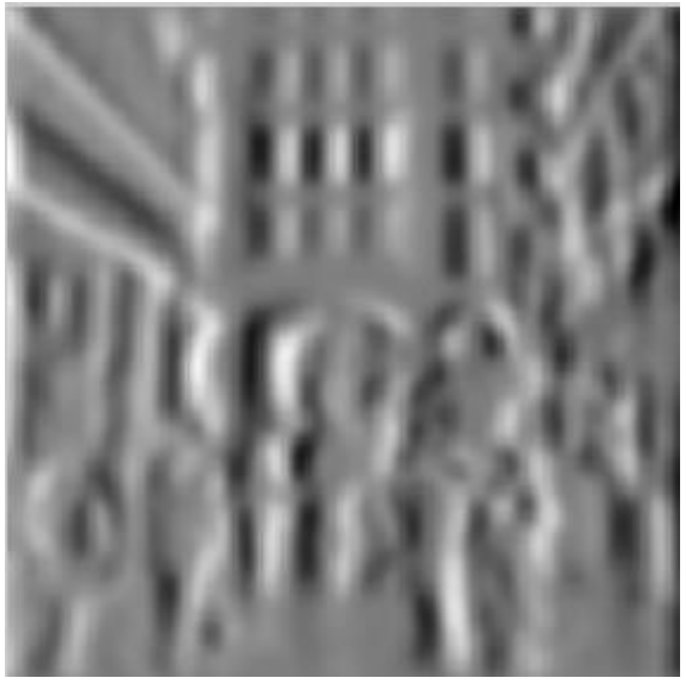
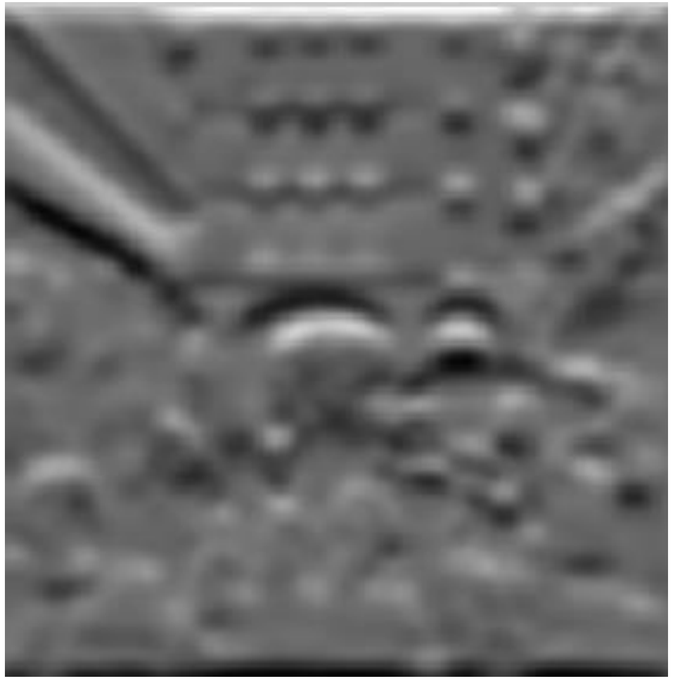
Any orientation can be computed as a linear combination of two filtered images

$$= \cos(\alpha) g_x(x,y) \otimes I(x,y) + \sin(\alpha) g_y(x,y) \otimes I(x,y)$$

Steereability of gaussian derivatives, Freeman & Adelson 92

Example: “steering” to 45°

$\cos(\alpha)$  + $\sin(\alpha)$  = 

$\cos(\alpha)$  $+\sin(\alpha)$  = 

Steereability of gaussian derivatives, Freeman & Adelson 92

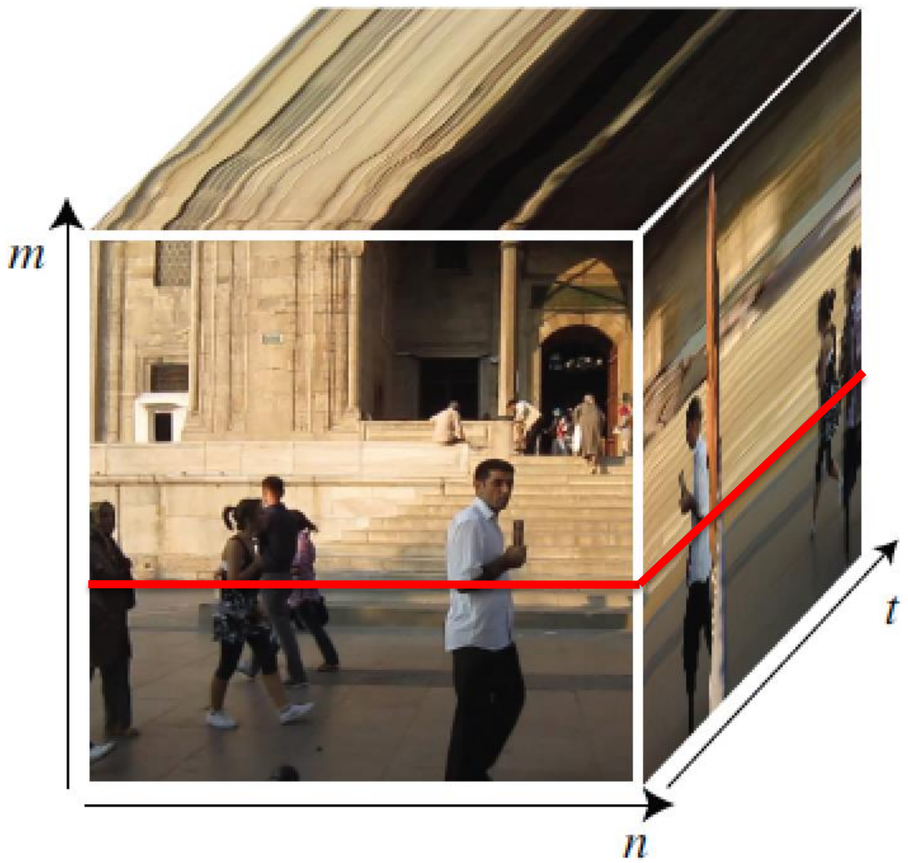
Temporal filtering

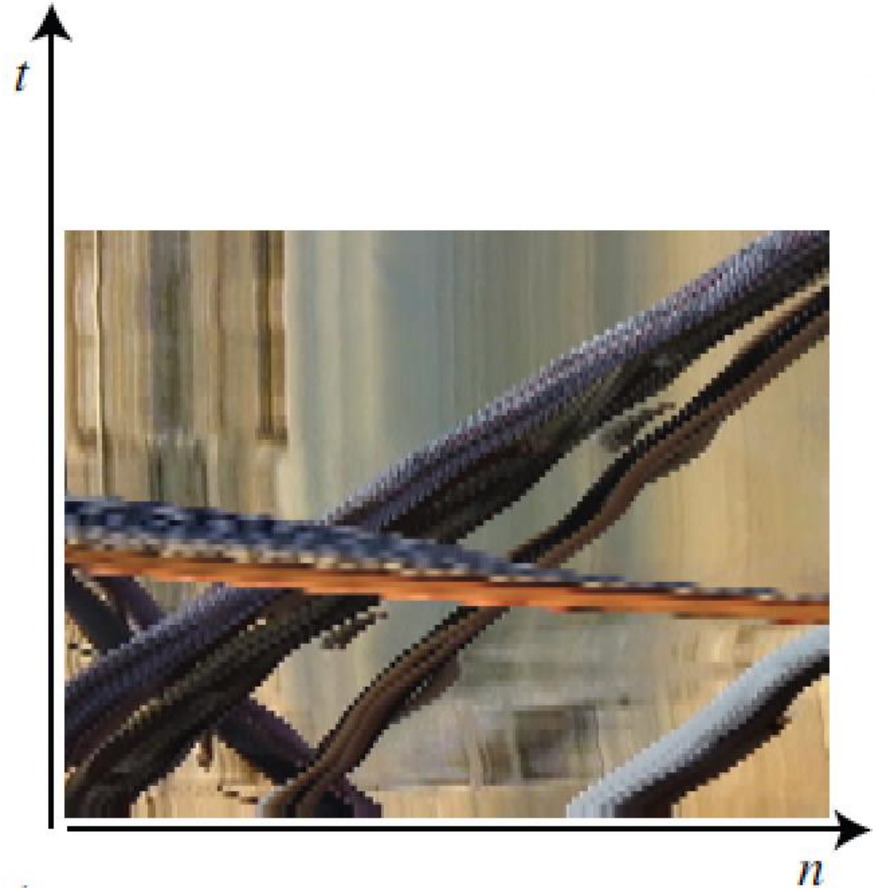
why filter videos over time?

Sequences

time

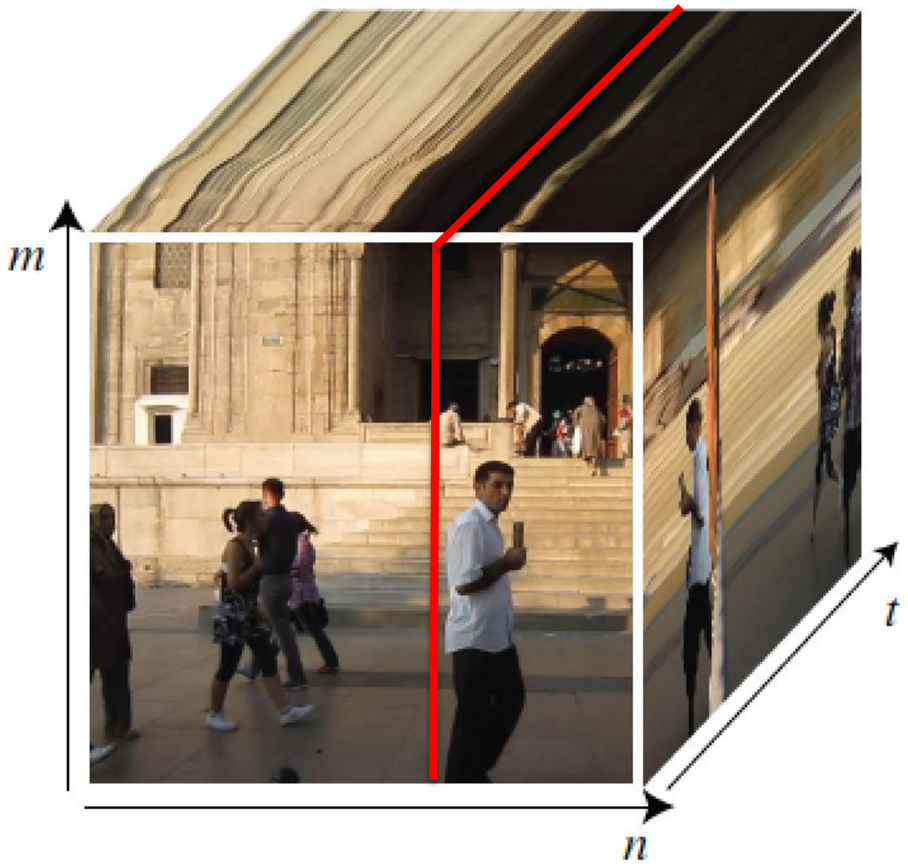
Sequences





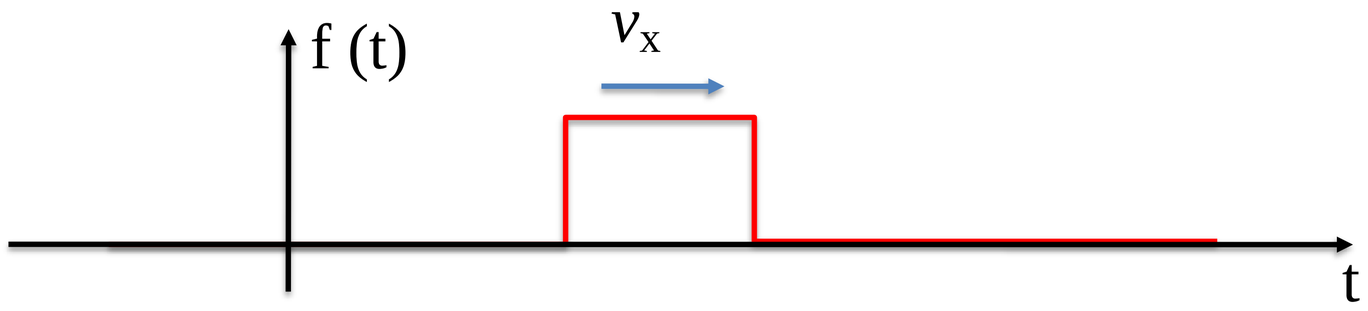
Cube size = 128x128x90

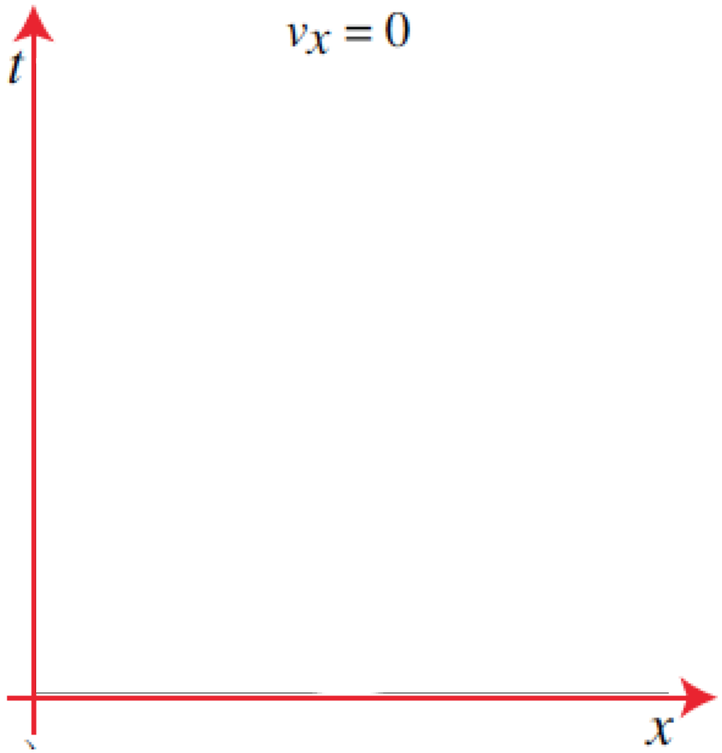
Sequences

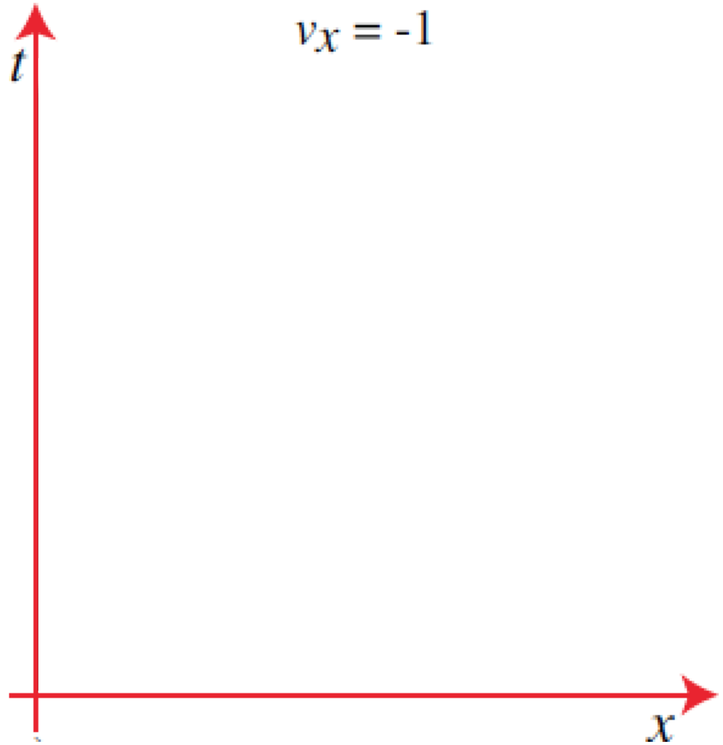


Cube size = 128x128x90

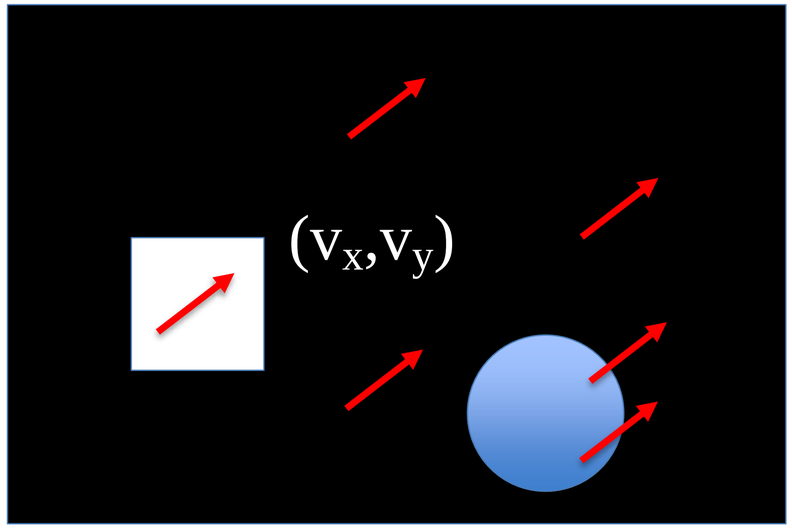
A box moving with speed v_x







Global constant motion



A global motion of the image can be written as:

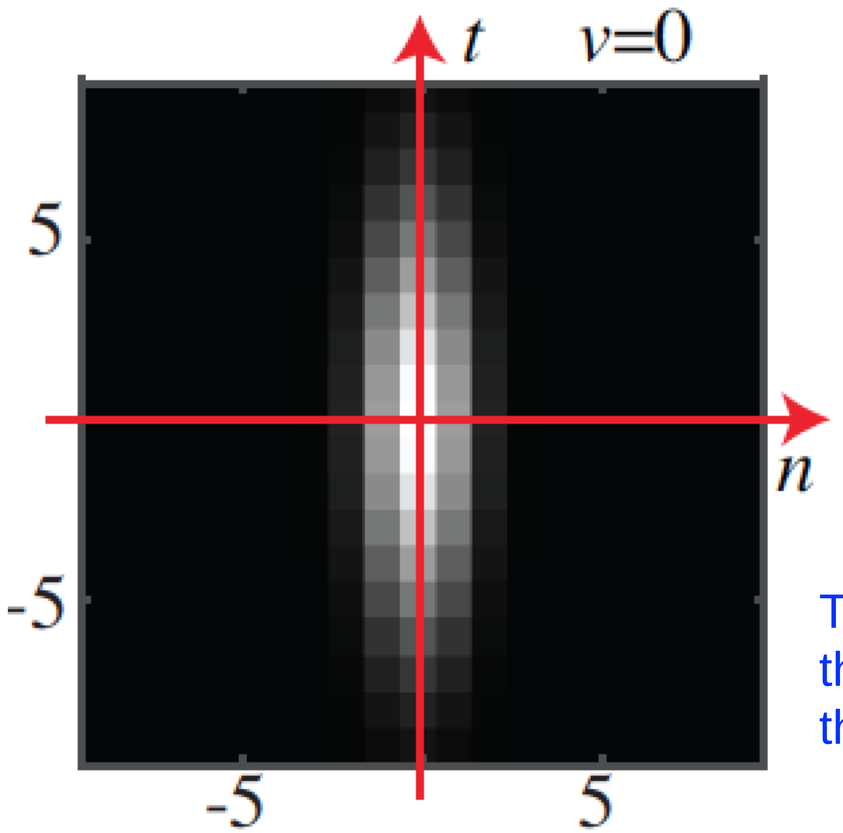
$$f(x, y, t) = f_0(x - v_x t, y - v_y t)$$

Where:

$$f_0(x, y) = f(x, y, 0)$$

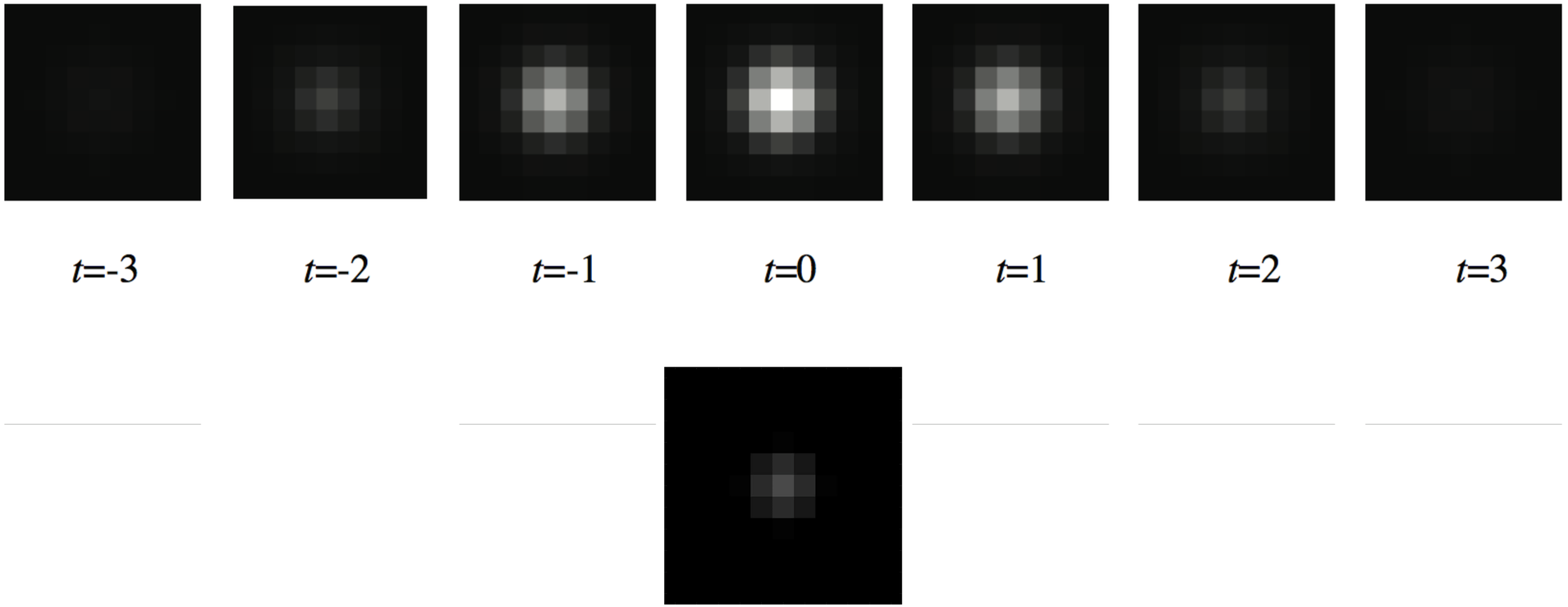
Temporal Gaussian

$$g(x, y, t; \sigma_x, \sigma_t) = \frac{1}{(2\pi)^{3/2}\sigma_x^2\sigma_t} \exp -\frac{x^2 + y^2}{2\sigma_x^2} \exp -\frac{t^2}{2\sigma_t^2}$$



This filter keeps stationary things sharp, and blurs moving things.

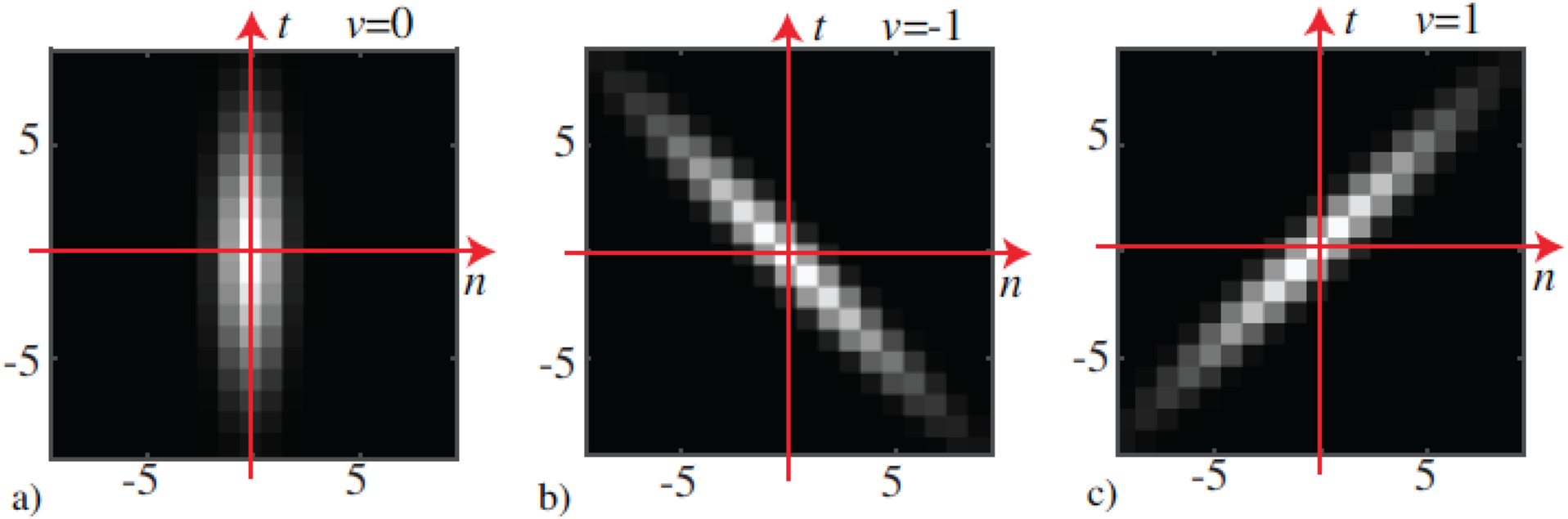
Spatio-temporal Gaussian



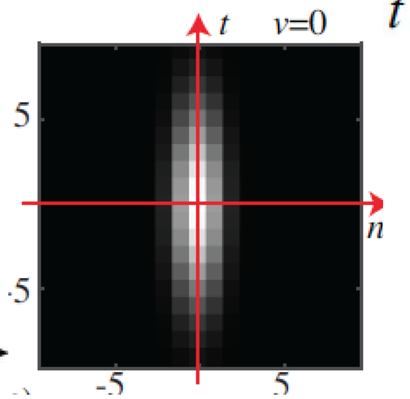
Spatio-temporal Gaussian

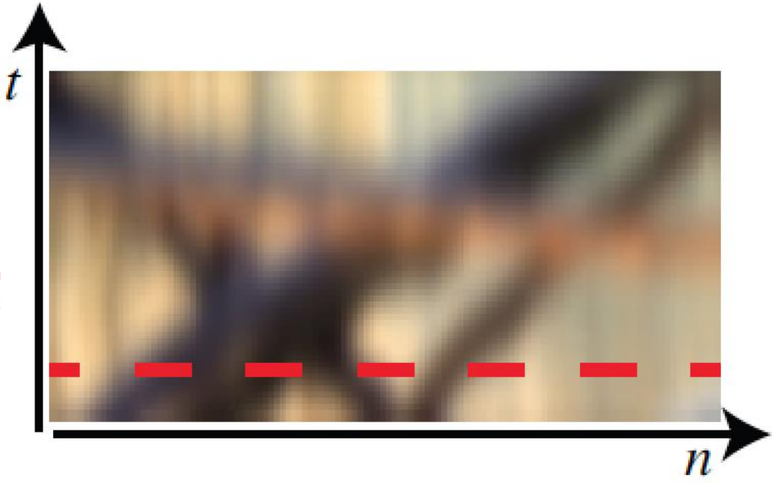
How could we create a filter that keeps sharp objects that move at some velocity **(vx , vy)** while blurring the rest?

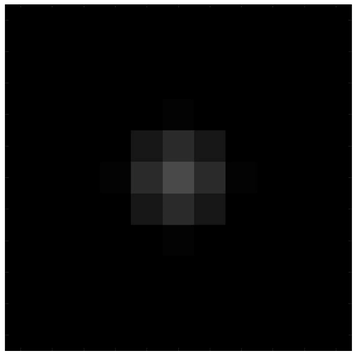
$$g_{v_x,v_y}(x,y,t) = g(x - v_x t, y - v_y t, t)$$

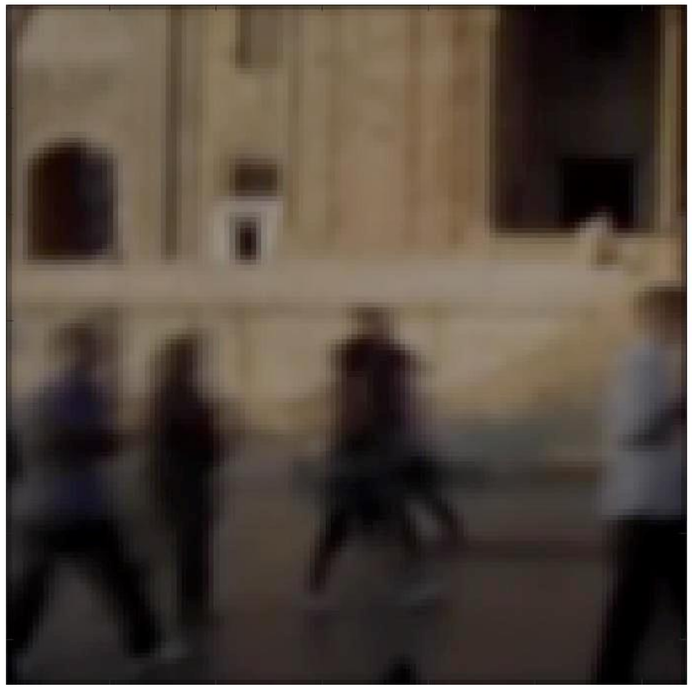



t
n

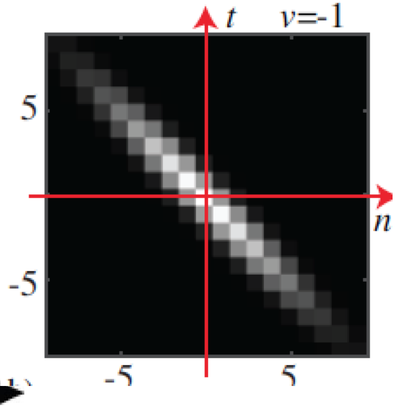

t
v=0
5
-5
n
-5
5


t
n

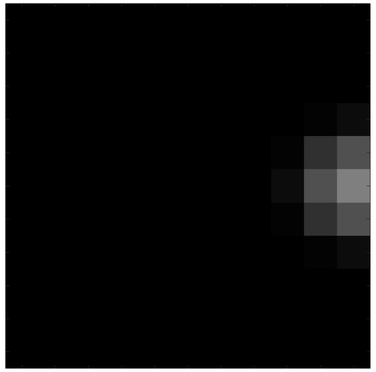


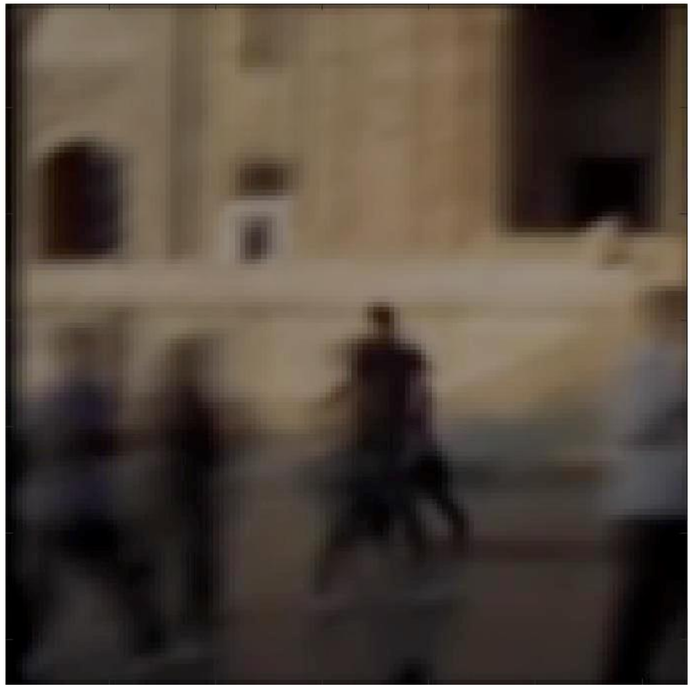


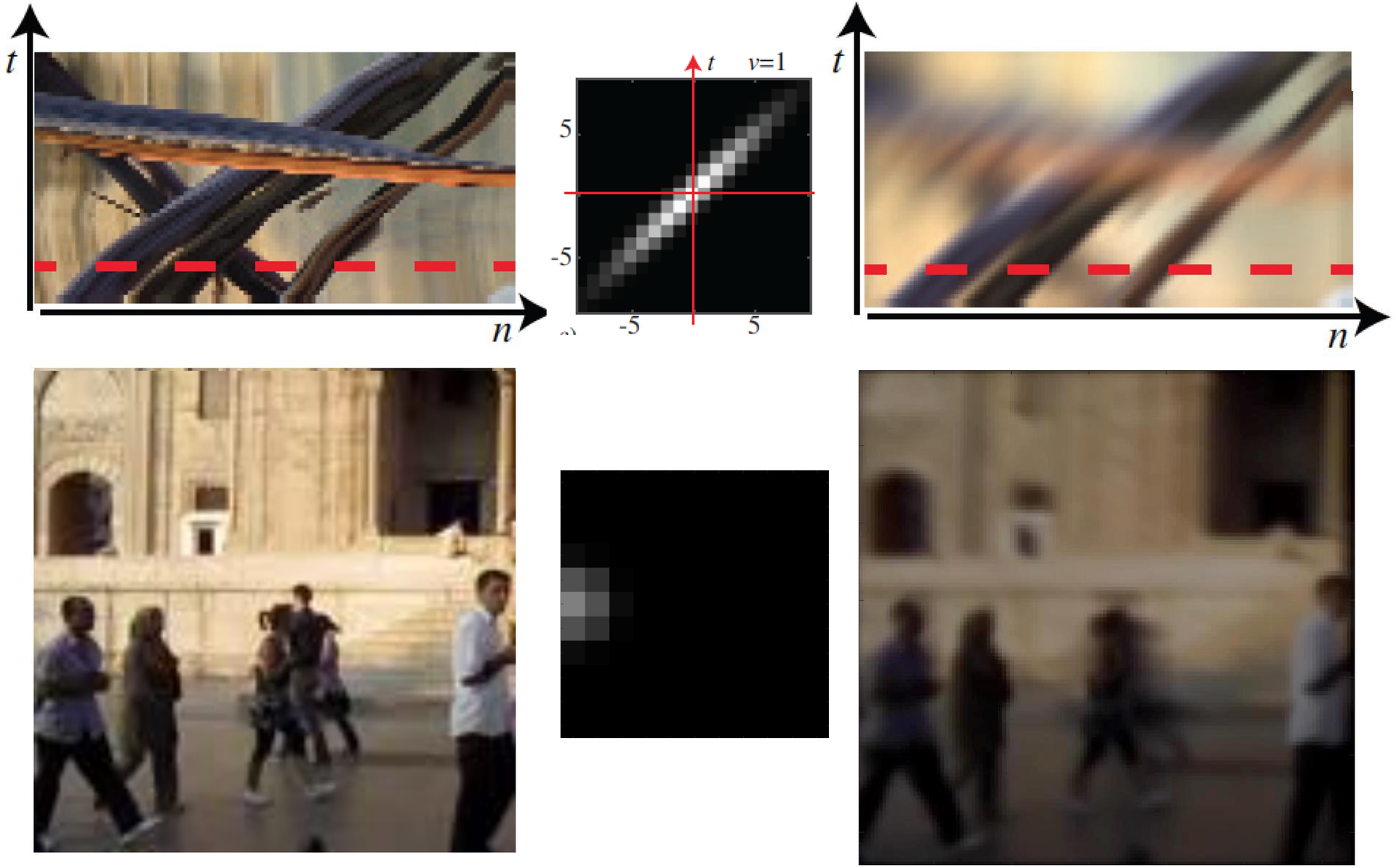

t
n


t
v=-1
5
-5
n
-5
5

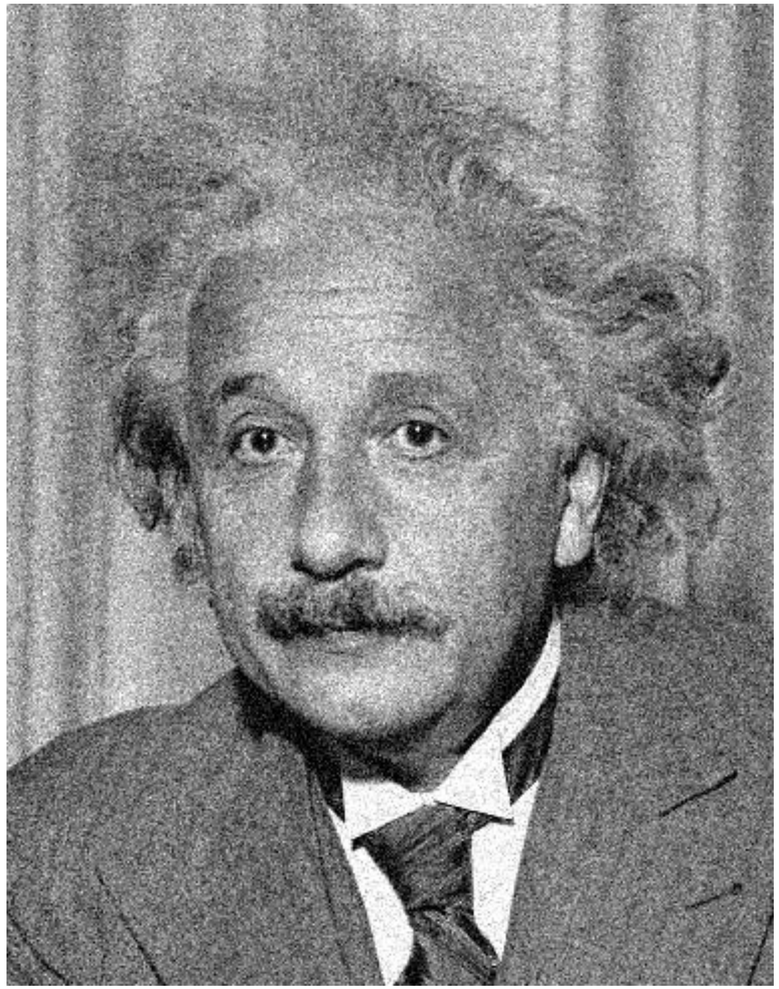

t
n



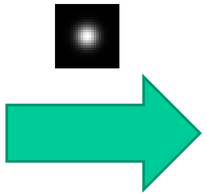



t
n
t
v=1
5
-5
-5
5
t
n

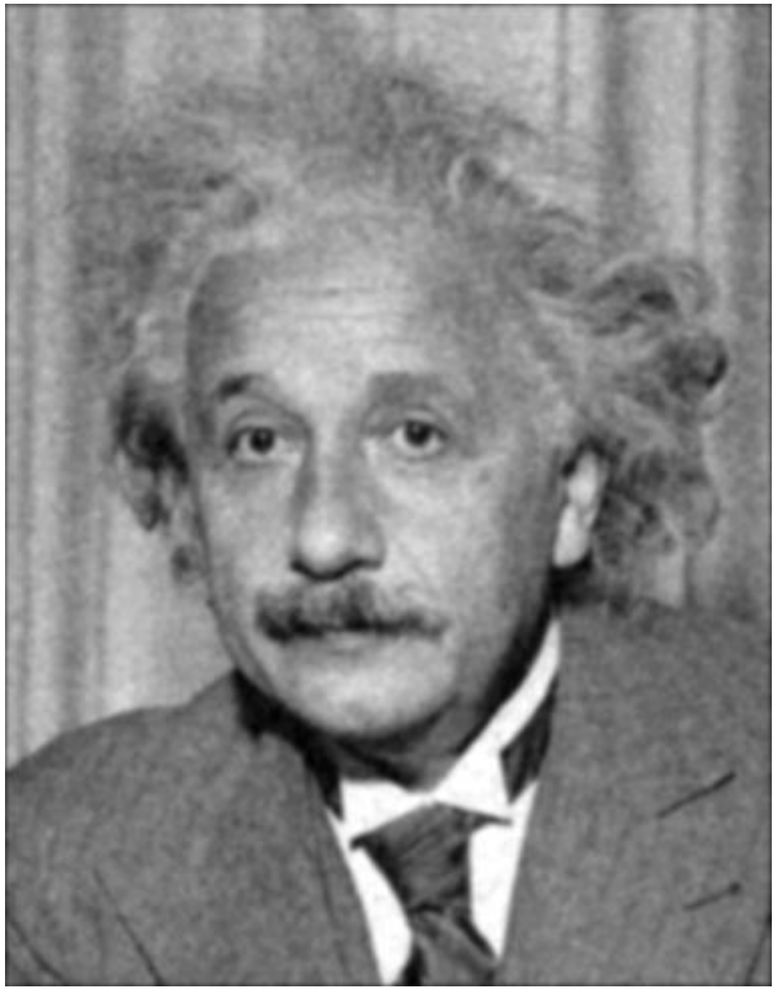
Non-linear Filters



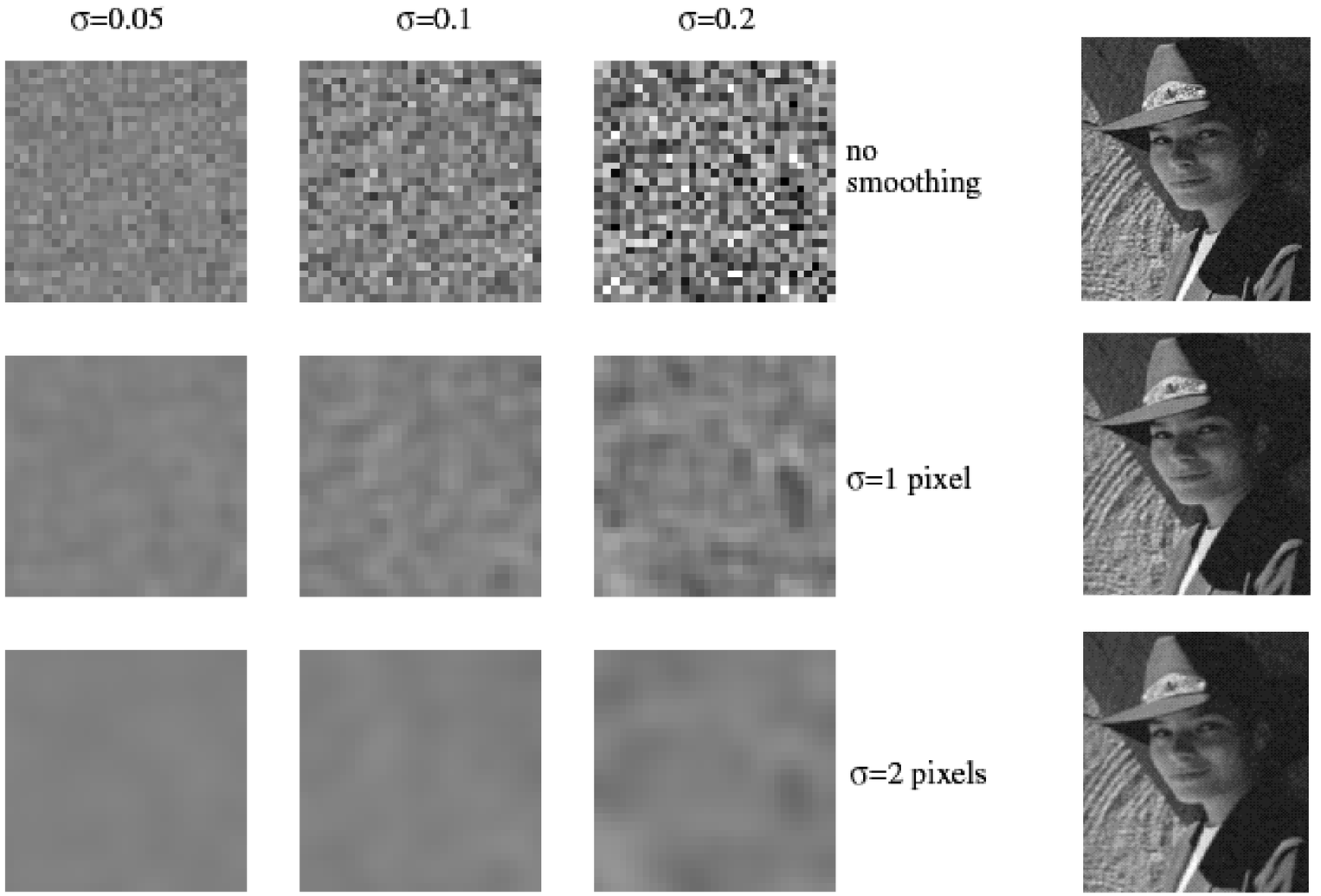
Additive Gaussian Noise



Gaussian
Filter



Reducing Gaussian noise



Smoothing with larger standard deviations suppresses noise, but also blurs the image

Source: S. Lazebnik

Reducing salt-and-pepper noise by Gaussian smoothing

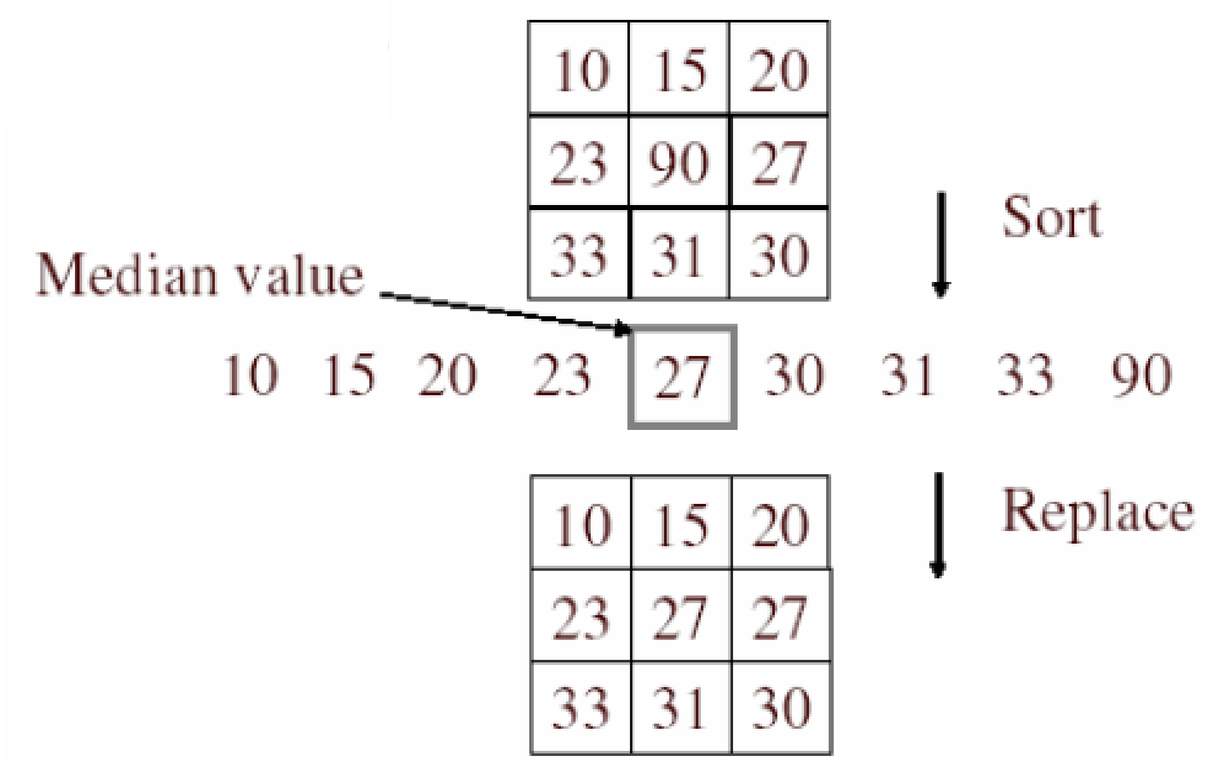
3x3

5x5

7x7

Alternative idea: Median filtering

A **median filter** operates over a window by selecting the median intensity in the window



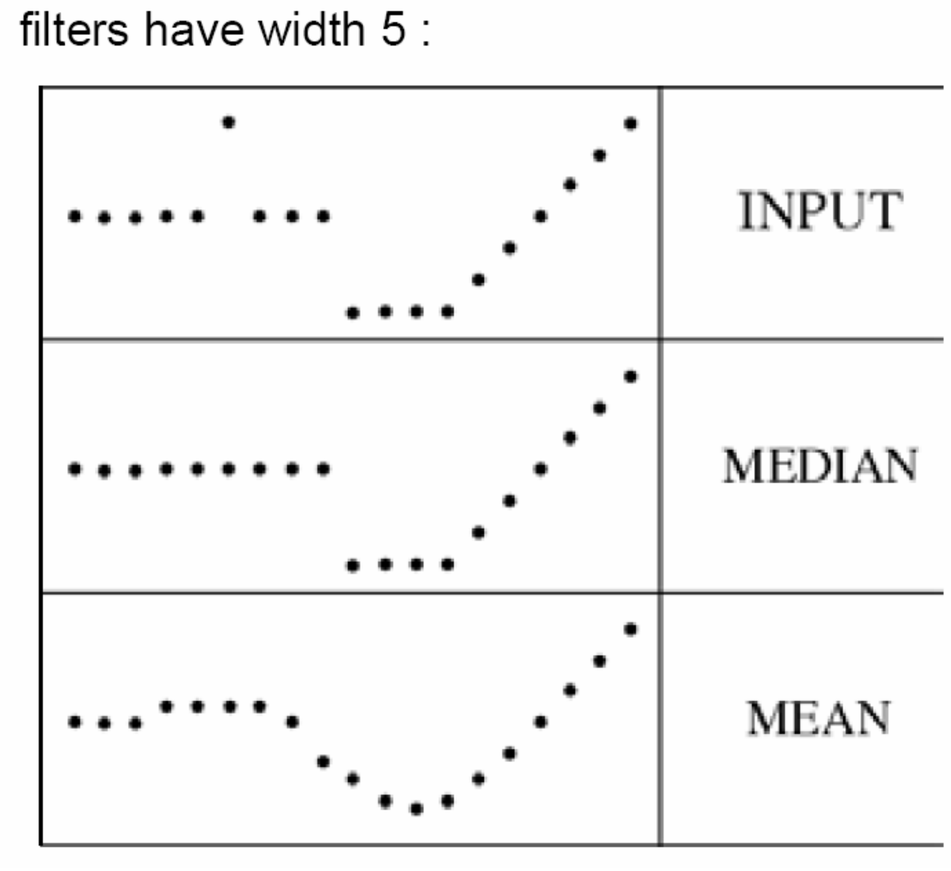
- Is median filtering linear?

Source: K. Grauman

Median filter

What advantage does median filtering have over Gaussian filtering?

- Robustness to outliers

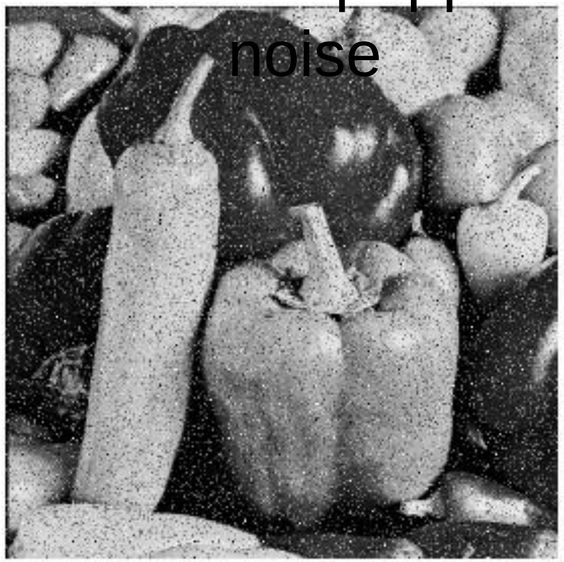


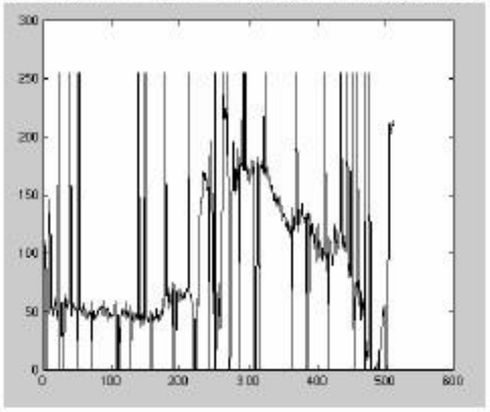
Source: K. Grauman

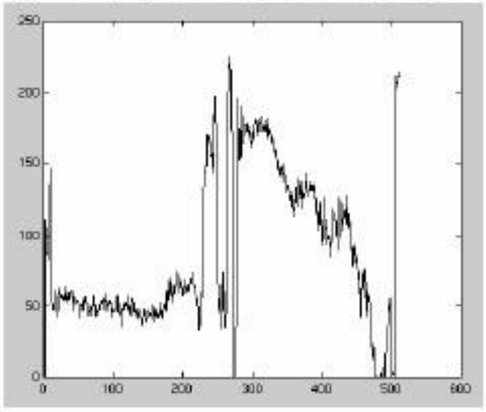
Median filter

Salt-and-pepper noise

Median filtered



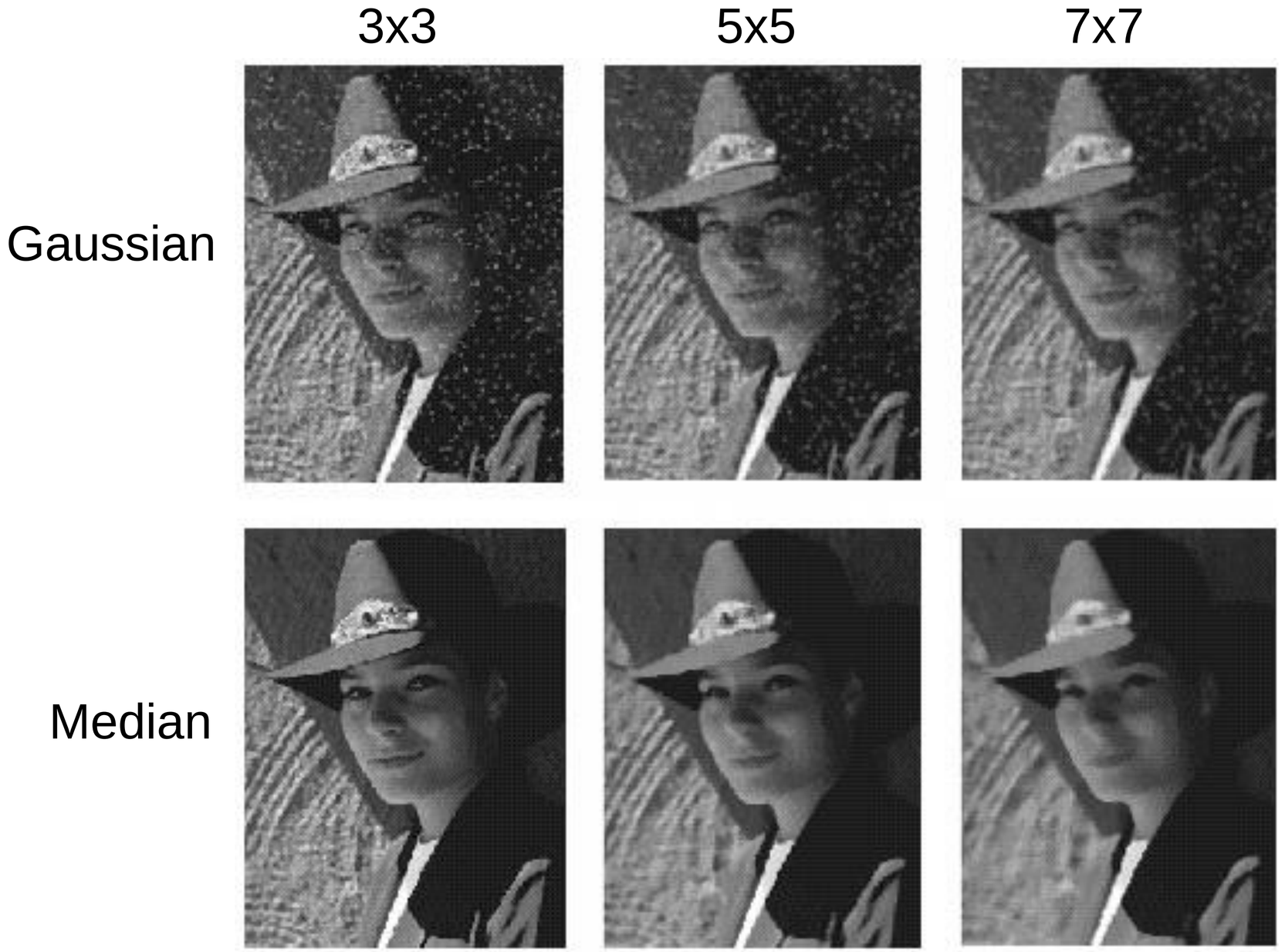




MATLAB: medfilt2(image, [h w])

Source: M. Hebert

Median vs. Gaussian filtering

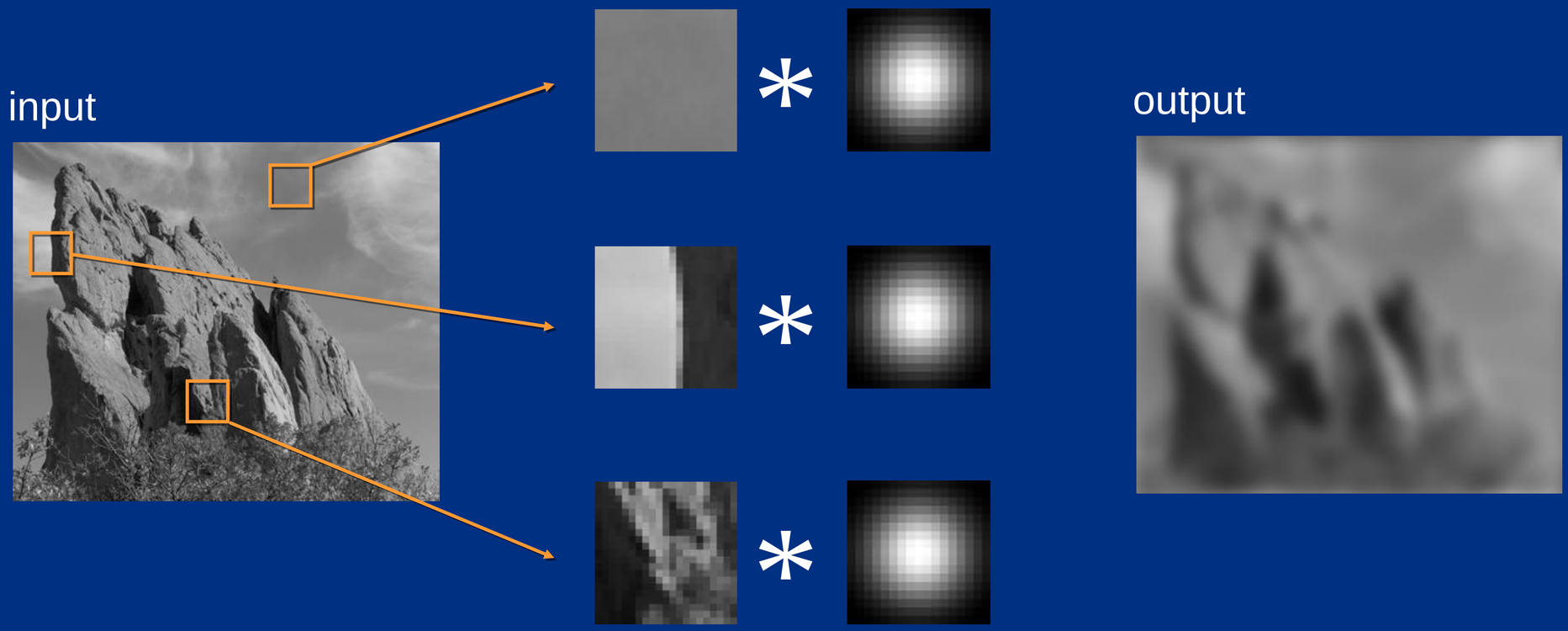


A Gentle Introduction
to Bilateral Filtering
and its Applications

“Fixing the Gaussian Blur”: the Bilateral Filter

Sylvain Paris – MIT CSAIL

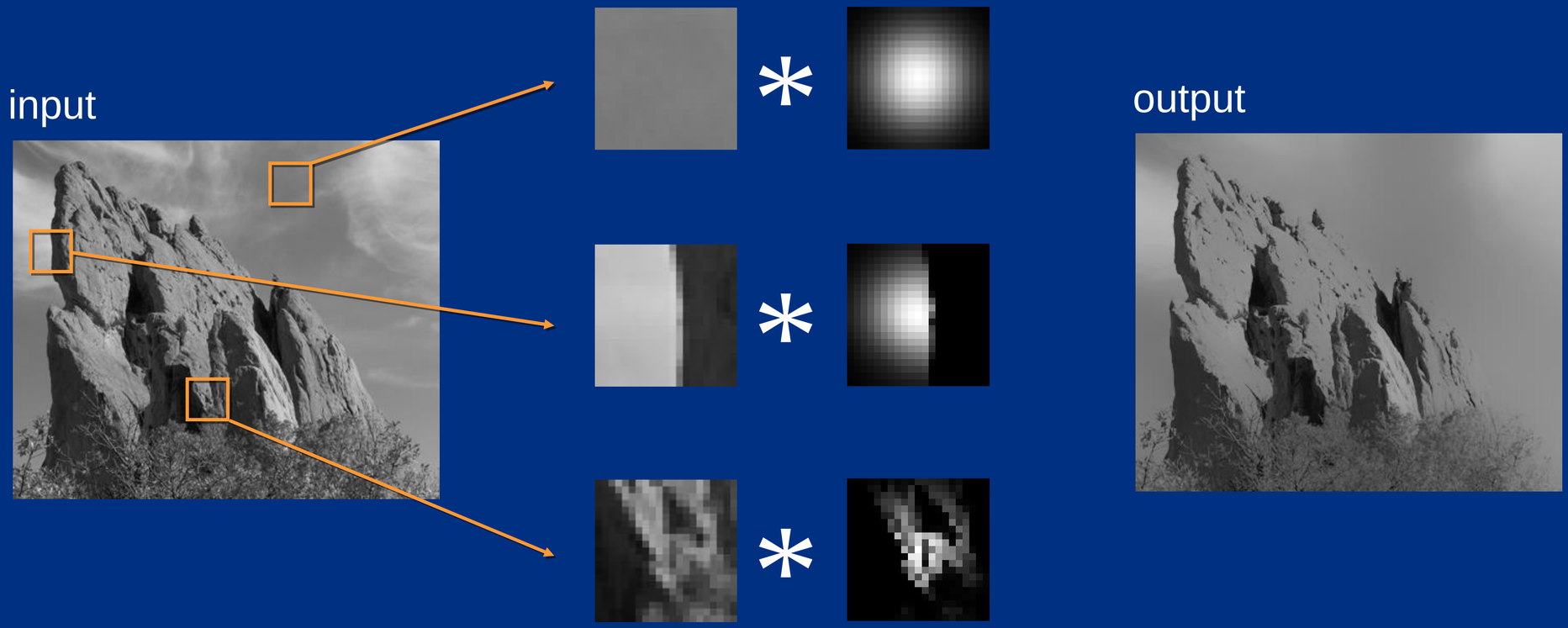
Blur Comes from Averaging across Edges



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

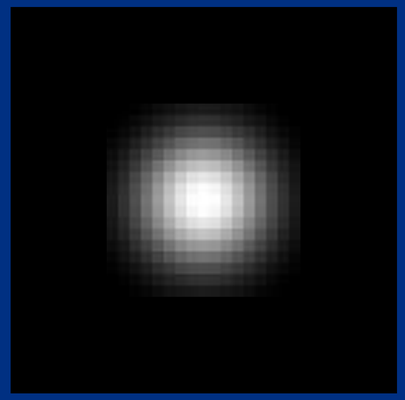
Same idea: **weighted average of pixels**.

new | not new | new

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

normalization factor

space weight

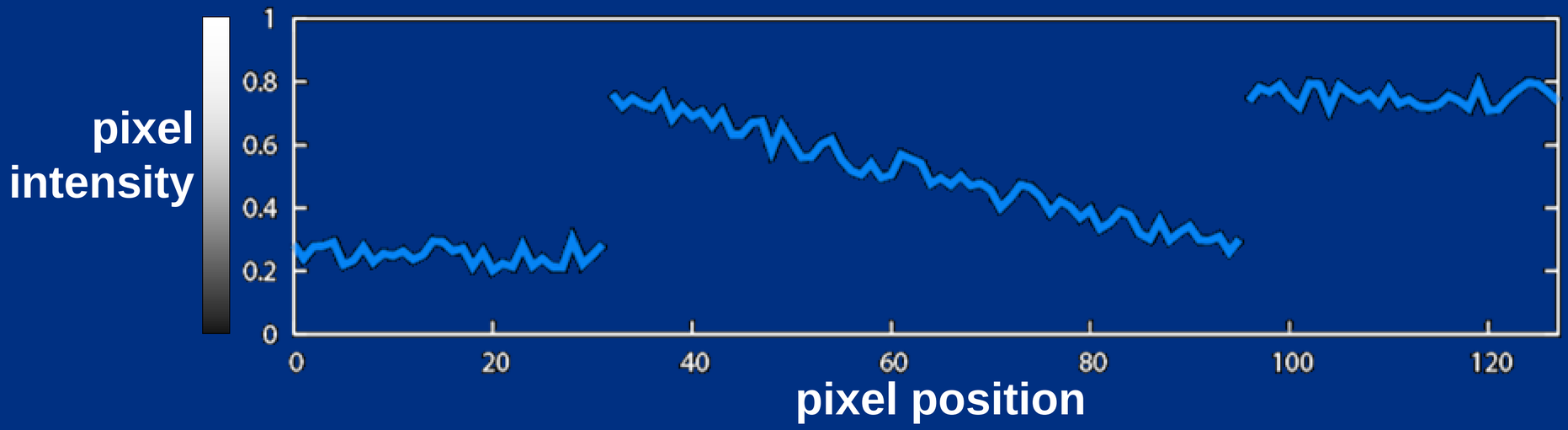


range weight

Illustration a 1D Image

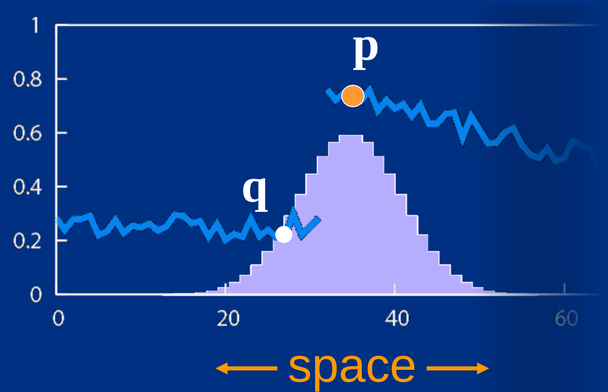
- 1D image = line of pixels

- Better visualized as a plot



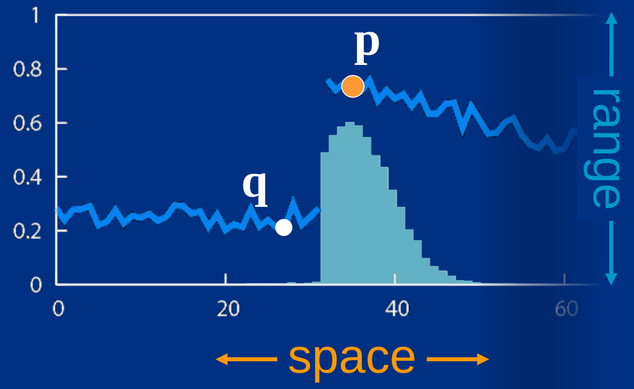
Gaussian Blur and Bilateral Filter

Gaussian blur



Bilateral filter

[Aurich 95, Smith 97, Tomasi 98]



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

space

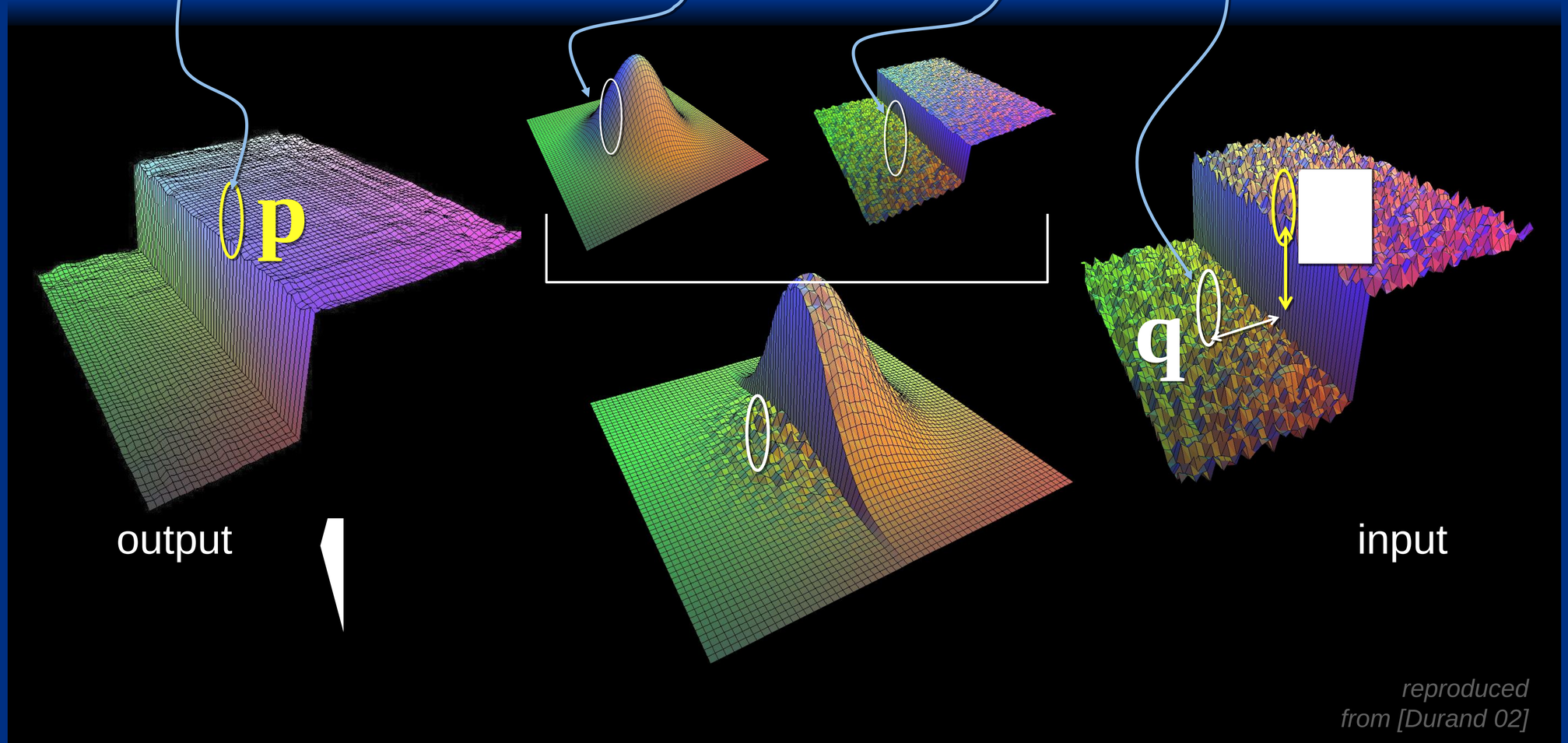
$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q}\in S} G_{\sigma_s}\left(\|\mathbf{p}-\mathbf{q}\|\right) G_{\sigma_r}\left(|I_{\mathbf{p}}-I_{\mathbf{q}}|\right) I_{\mathbf{q}}$$

space range

normalization

Bilateral Filter on a Height Field

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



reproduced from [Durand 02]

Space and Range Parameters

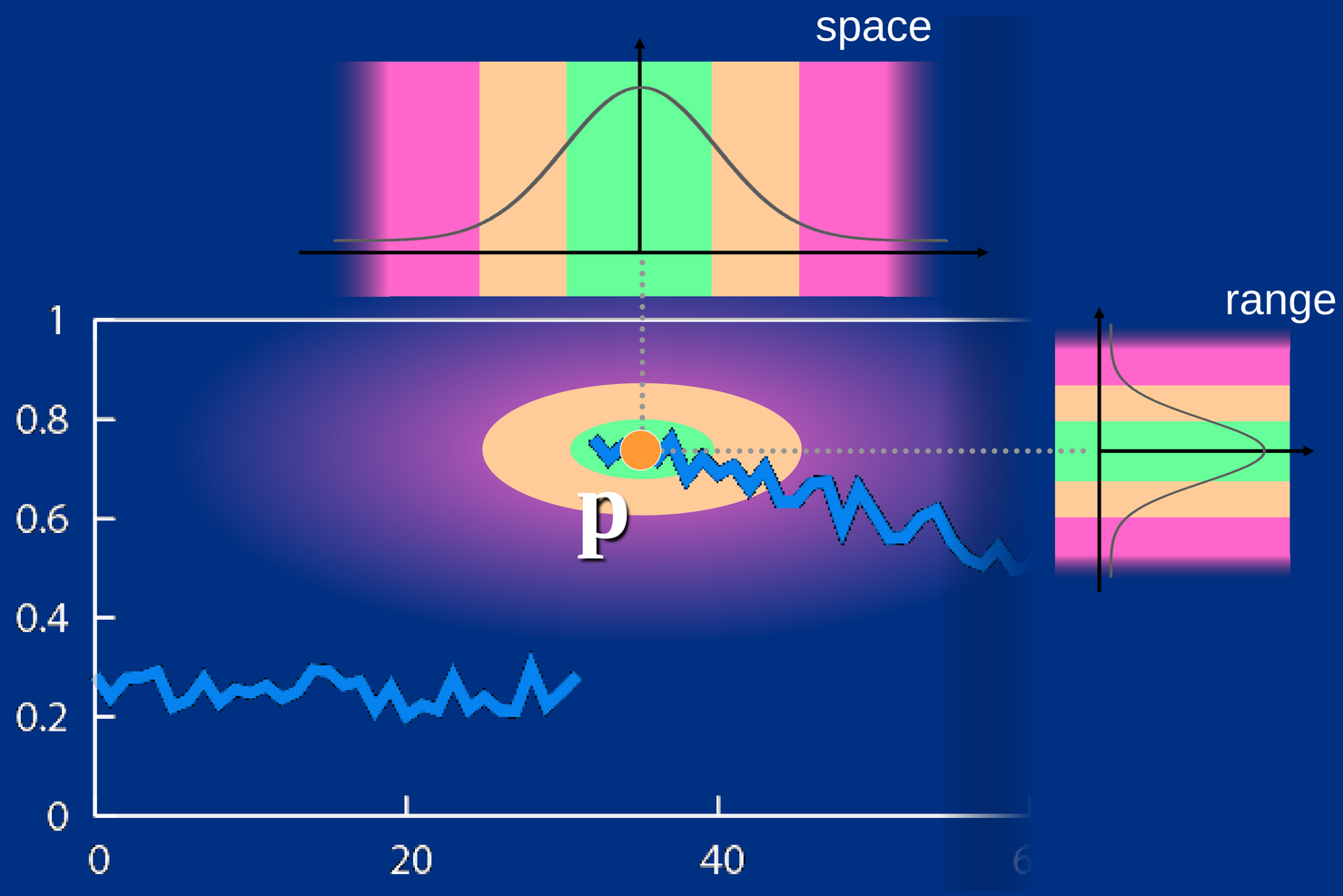
$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} G_{\sigma_s}\left(\|\mathbf{p}-\mathbf{q}\|\right) G_{\sigma_r}\left(|I_{\mathbf{p}} - I_{\mathbf{q}}|\right) I_{\mathbf{q}}$$

- space σ_s : spatial extent of the kernel, size of the considered neighborhood.

- range σ_r : “minimum” amplitude of an edge

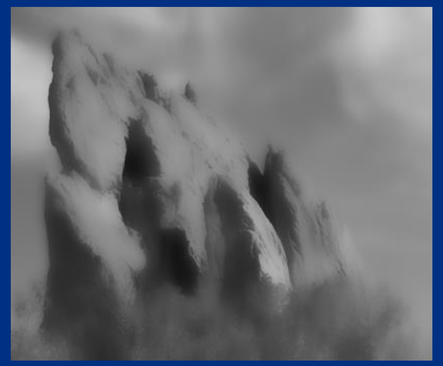
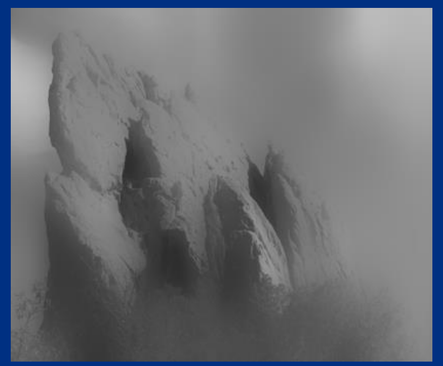
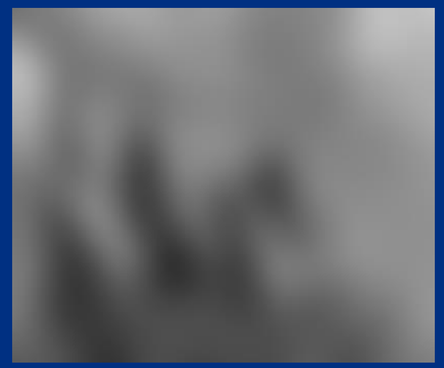
Influence of Pixels

Only pixels close in space and in range are considered.



Exploring the Parameter Space

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 Range Parameter

input

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

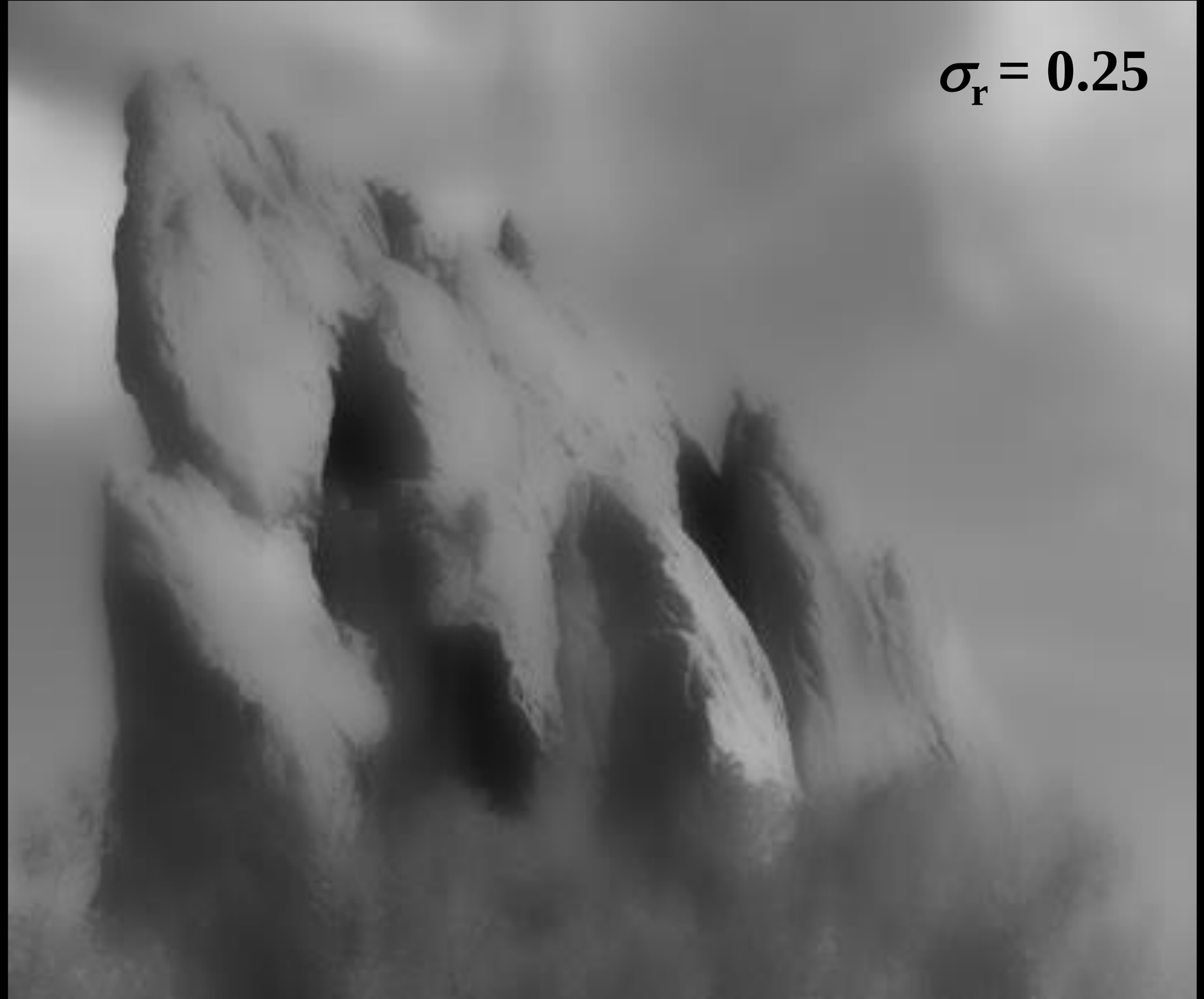
$\sigma_s = 2$

$\sigma_s = 6$

$\sigma_s = 18$


input


$\sigma_r = 0.1$


$\sigma_r = 0.25$


$\sigma_r = \infty$
(Gaussian blur)

Varying the Space Parameter

input

$\sigma_r = 0.1$

$\sigma_r = 0.25$

$\sigma_r = \infty$
(Gaussian blur)

$\sigma_s = 2$

$\sigma_s = 6$

$\sigma_s = 18$


input


$\sigma_s = 2$


$\sigma_s = 6$


$\sigma_s = 18$