Image Filtering

CS172 Computer Vision I

Instructor: Jiayuan Gu

Agenda

- Image Representations
- Filtering
  - Convolution
  - Sharpening
  - Smoothing
    - Gaussian Kernel
  - Denoising

Types of Images

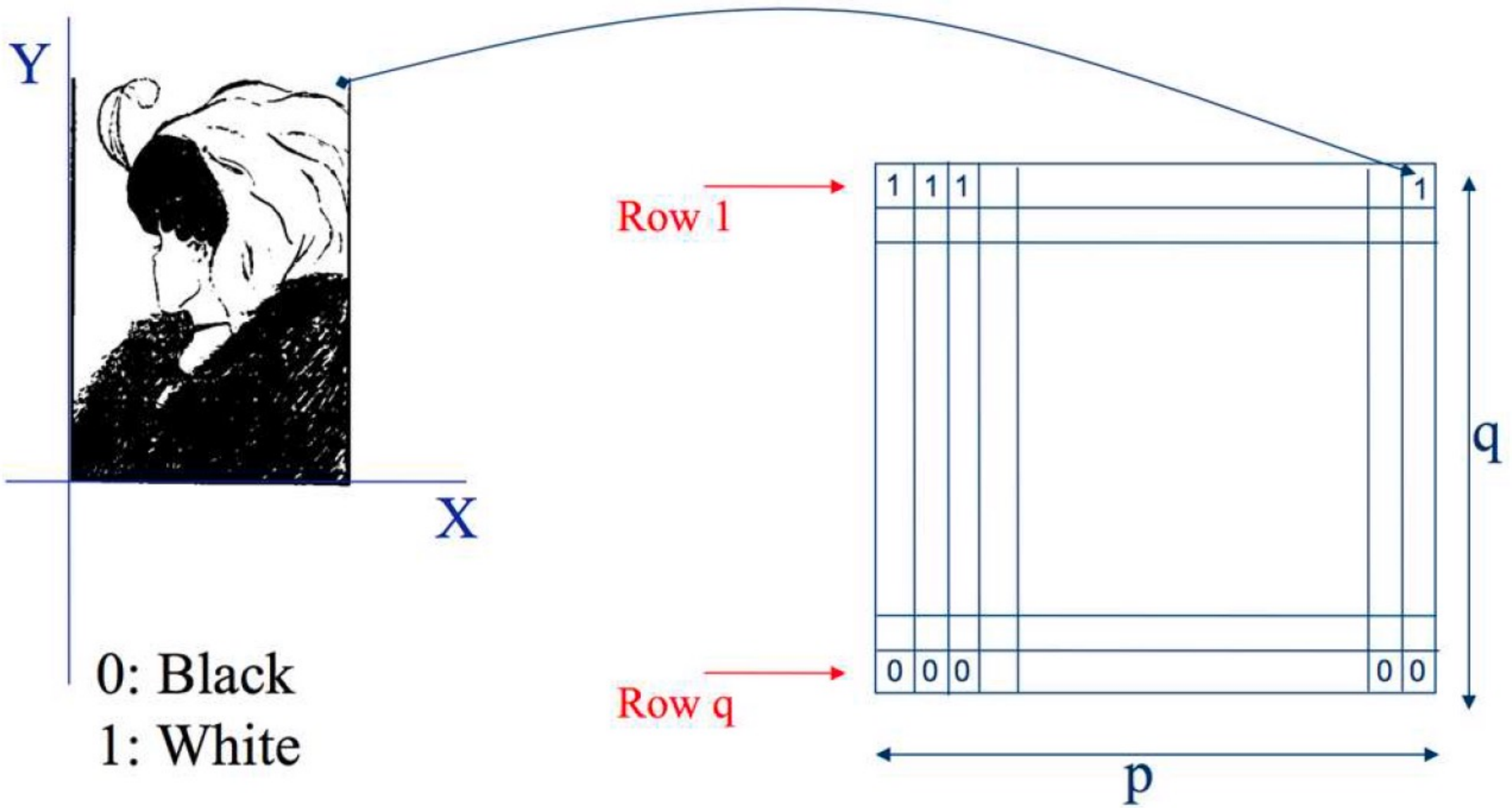
Binary

Gray Scale

Color

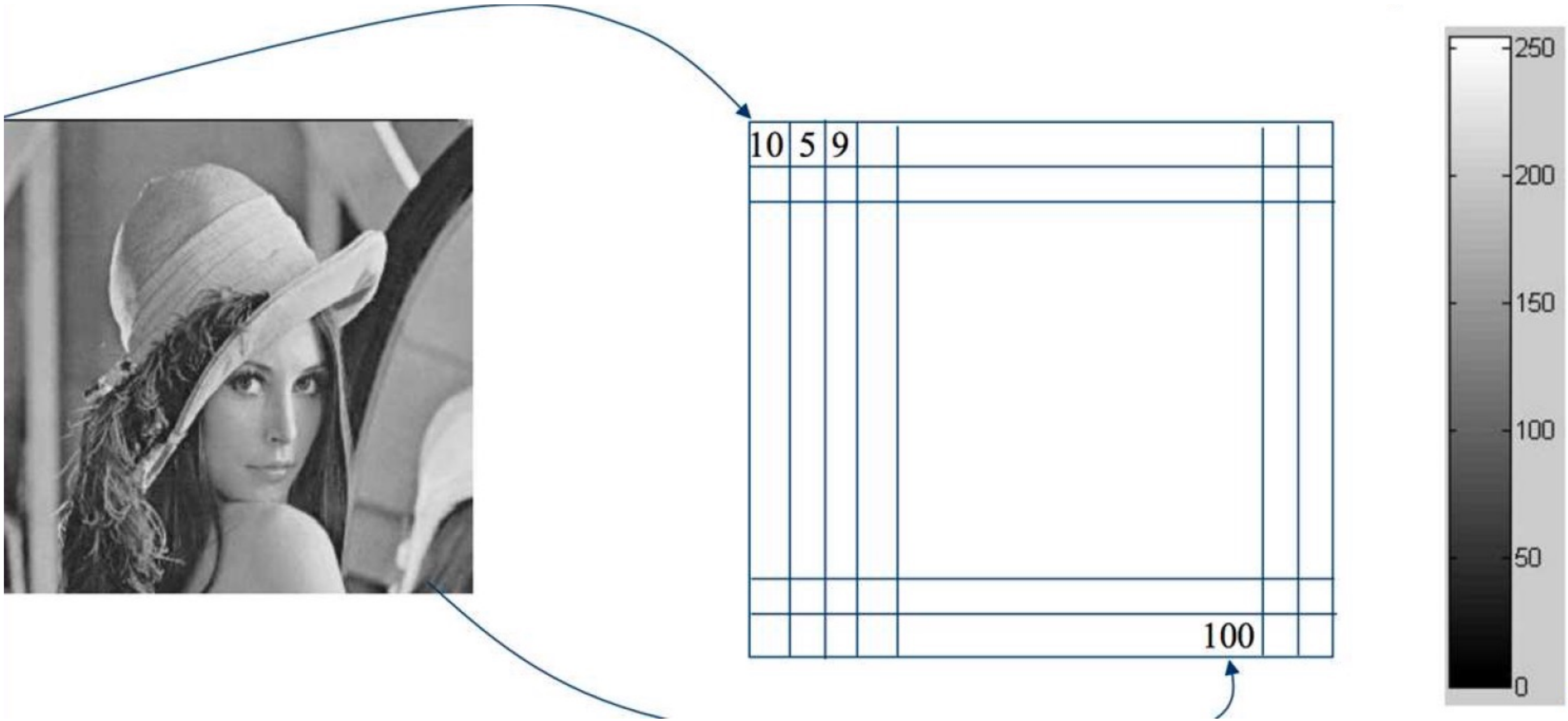
Source： Ulas Bagci

Binary Image



Source： Ulas Bagci

Grayscale Image

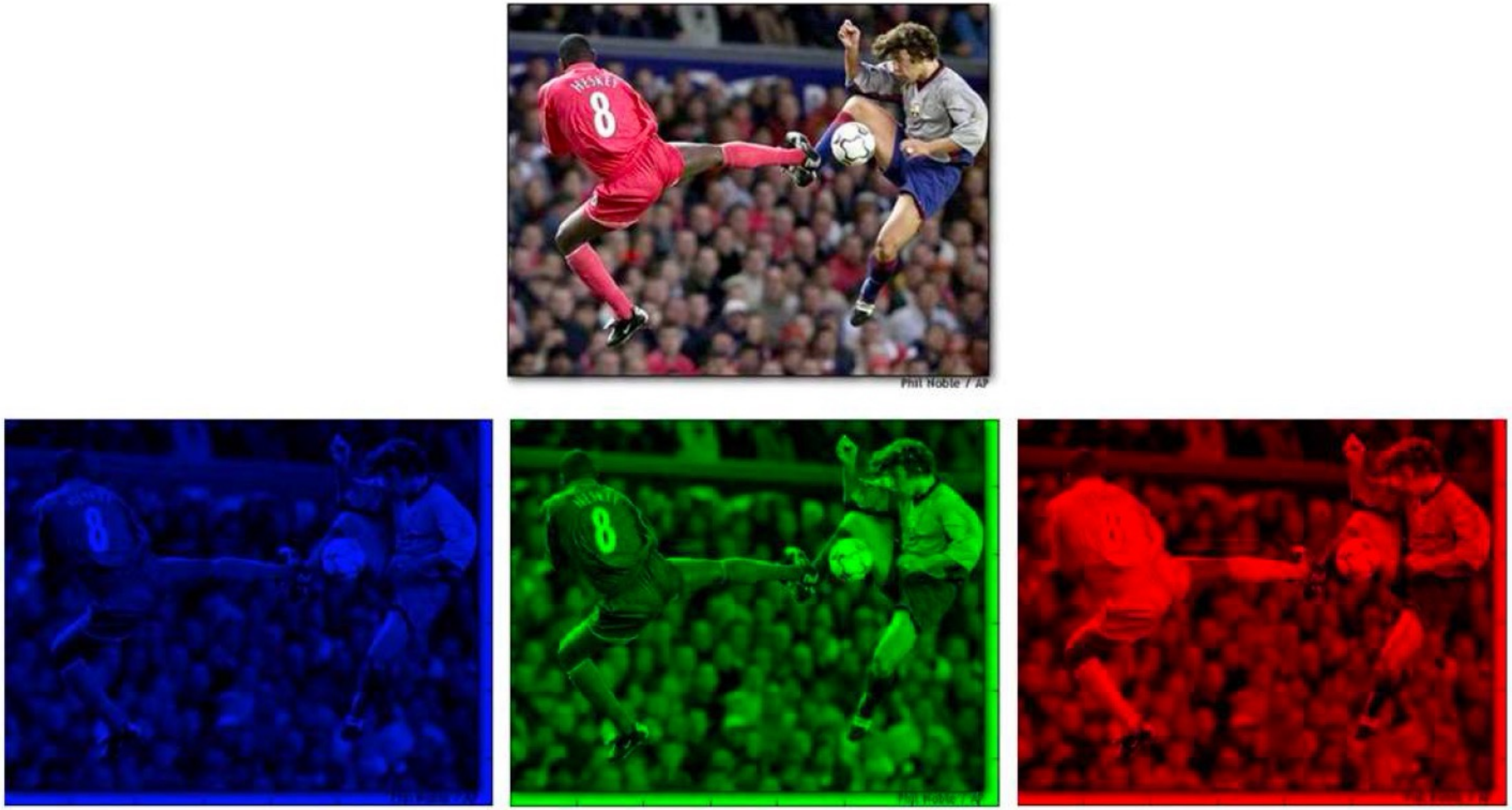


Color Image

Source： Ulas Bagci

One channel (red)

Color Image



Source： Ulas Bagci

Filtering

Convolution

Motivation: Image Denoising

How can we reduce noise in a photograph?

Credit to Lana Lazebnik

Moving average

- Let’s replace each pixel with a *weighted average* of its neighborhood
- The weights are called the *filter kernel*
- What are the weights for the average of a 3x3 neighborhood?

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

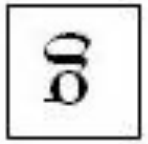
“box filter”

Source: D. Lowe

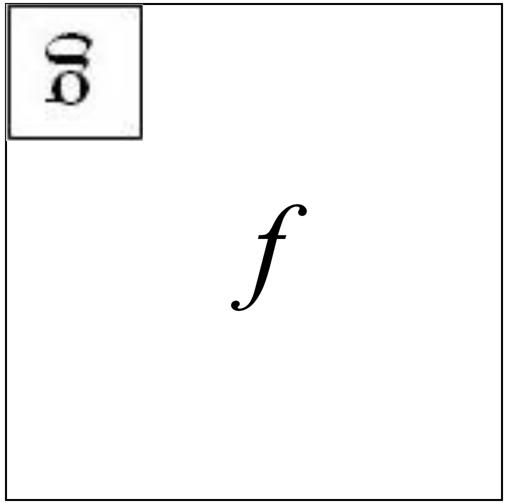
Defining Convolution

- Let *f* be the image and *g* be the kernel. The output of convolving *f* with *g* is denoted *f * g*.

$$(f * g)[m,n] = \sum_{k,l} f[m-k, n-l]\, g[k,l]$$



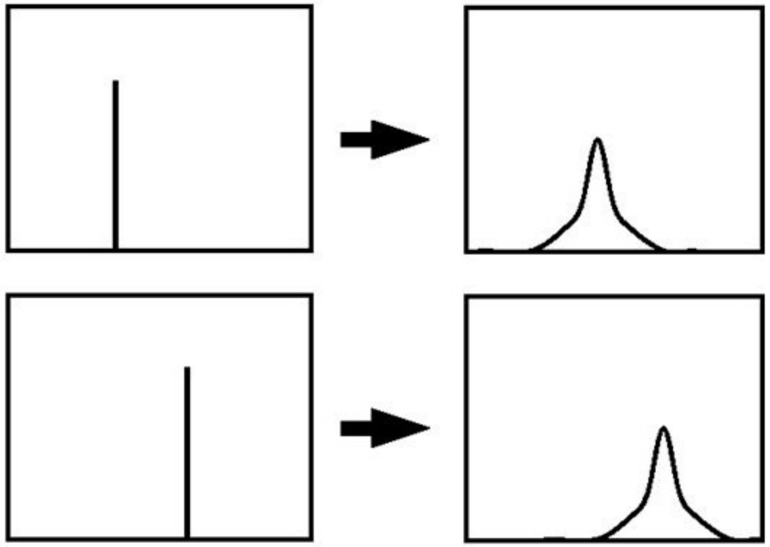
Convention: kernel is “flipped”



Source: F. Durand

Key Properties

- **Shift invariance:** same behavior regardless of pixel location: filter(shift(f)) = shift(filter(f))



- **Linearity:** filter(f_1 + f_2) = filter(f_1) + filter(f_2)

- Theoretical result: any linear shift-invariant operator can be represented as a convolution

Credit to Lana Lazebnik

Properties in More Detail

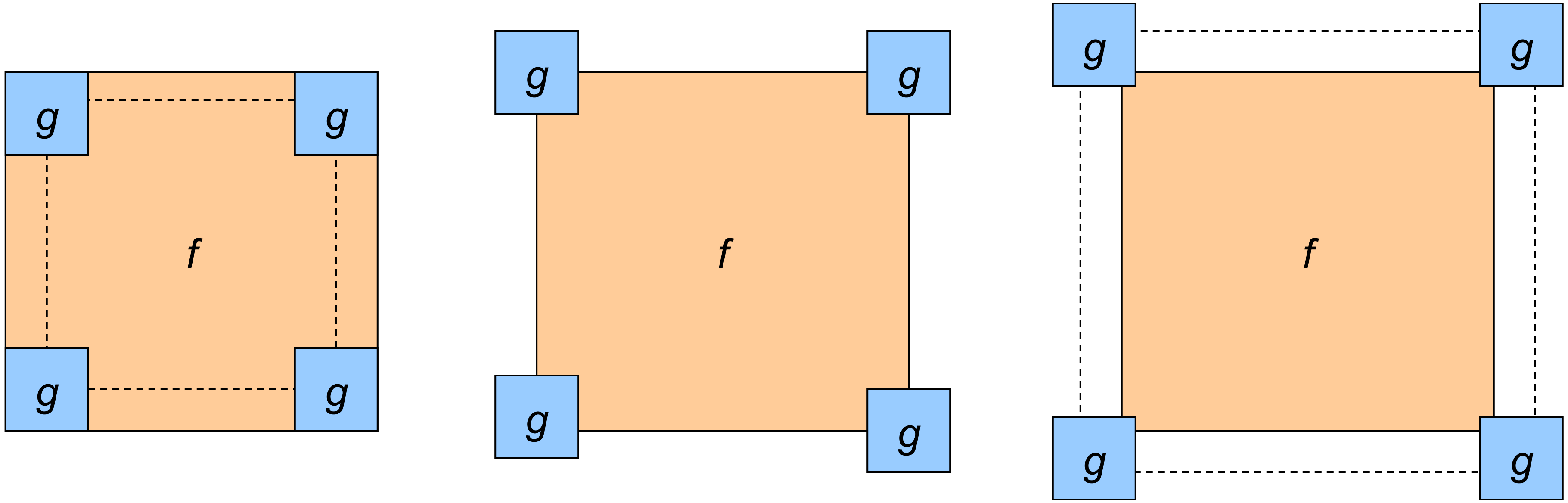
- Commutative: $a * b = b * a$
  - Conceptually no difference between filter and signal
- Associative: $a * (b * c) = (a * b) * c$
  - Often apply several filters one after another: $(((a * b_1) * b_2) * b_3)$
  - This is equivalent to applying one filter: $a * (b_1 * b_2 * b_3)$
- Distributes over addition: $a * (b + c) = (a * b) + (a * c)$
- Scalars factor out: $ka * b = a * kb = k\,(a * b)$
- Identity: unit impulse $e = [\ldots, 0, 0, 1, 0, 0, \ldots]$,
  $a * e = a$

Credit to Lana Lazebnik

Dealing with Edges

If we convolve image *f* with filter *g*, what is the size of the output?

Output is smaller than input | Output is same size as input | Output is larger than input



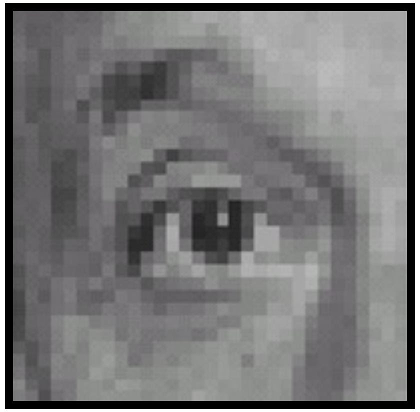
Credit to Lana Lazebnik

Dealing with Ddges

- If the filter window falls off the edge of the image, we need to pad the image
  - Zero pad (or clip filter)
  - Wrap around
  - Copy edge
  - Reflect across edge

Source: S. Marschner

Practice with Linear Filters



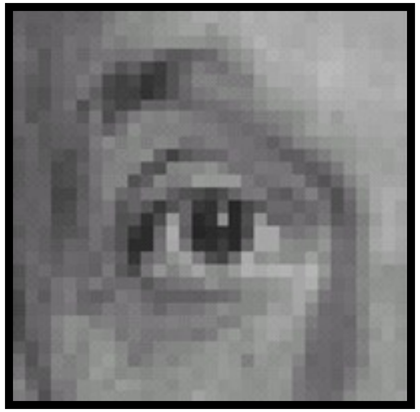
Original

0	0	0
0	1	0
0	0	0

?

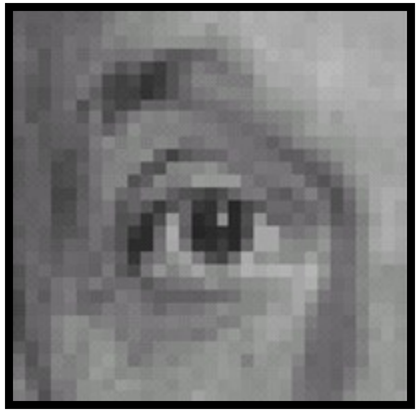
Source: D. Lowe

Practice with Linear Filters



Original

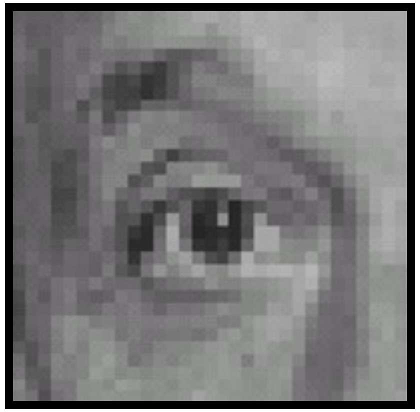
0	0	0
0	1	0
0	0	0



Filtered
(no change)

Source: D. Lowe

Practice with Linear Filters



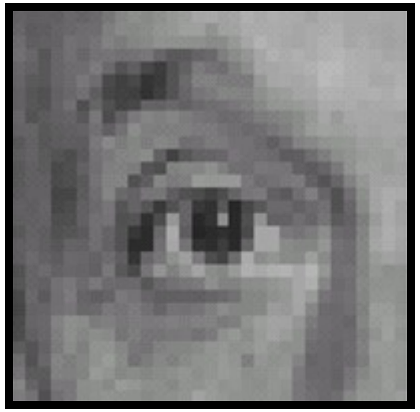
Original

0	0	0
0	0	1
0	0	0

?

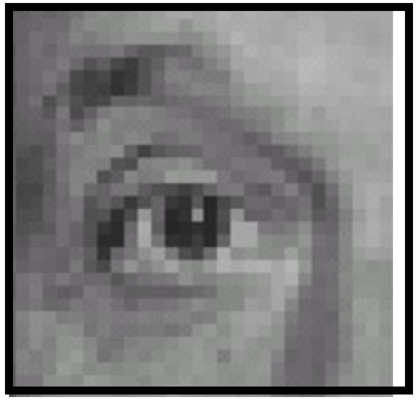
Source: D. Lowe

Practice with Linear Filters



Original

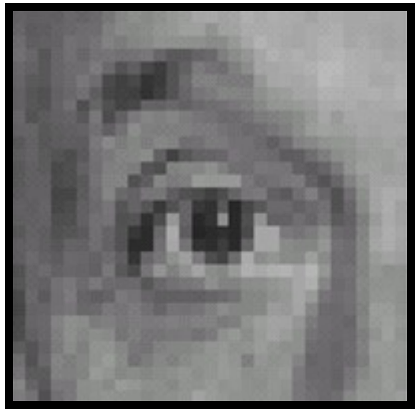
0	0	0
0	0	1
0	0	0



Shifted *left*
By 1 pixel

Source: D. Lowe

Practice with Linear Filters



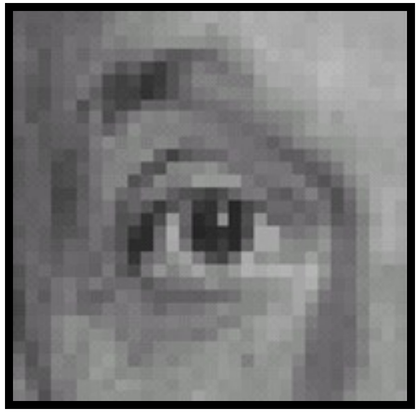
Original

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix}$$

?

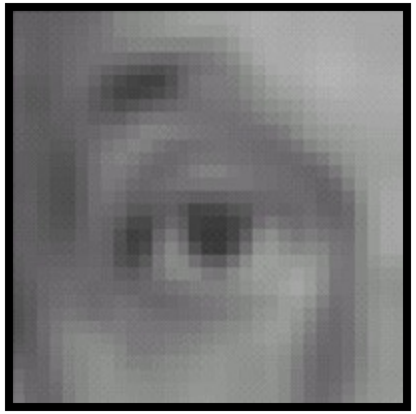
Source: D. Lowe

Practice with Linear Filters



Original

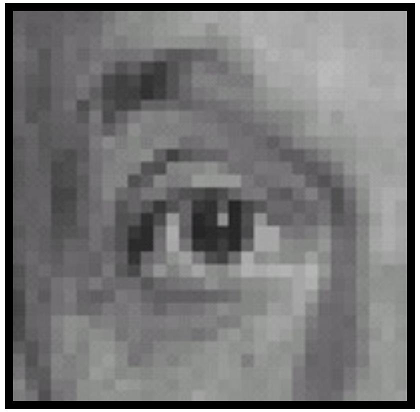
$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix}$$



Blur (with a box filter)

Source: D. Lowe

Practice with Linear Filters



Original

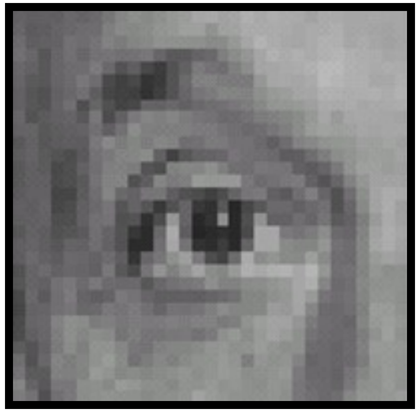
$$\begin{bmatrix}0 & 0 & 0\\ 0 & 2 & 0\\ 0 & 0 & 0\end{bmatrix} - \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

(Note that filter sums to 1)

?

Source: D. Lowe

Practice with Linear Filters



Original

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Sharpening filter

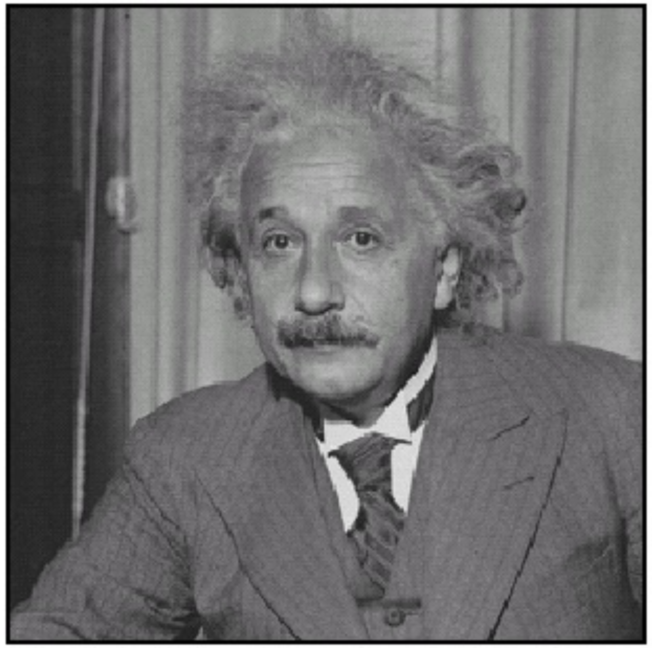
- Accentuates differences with local average

Source: D. Lowe

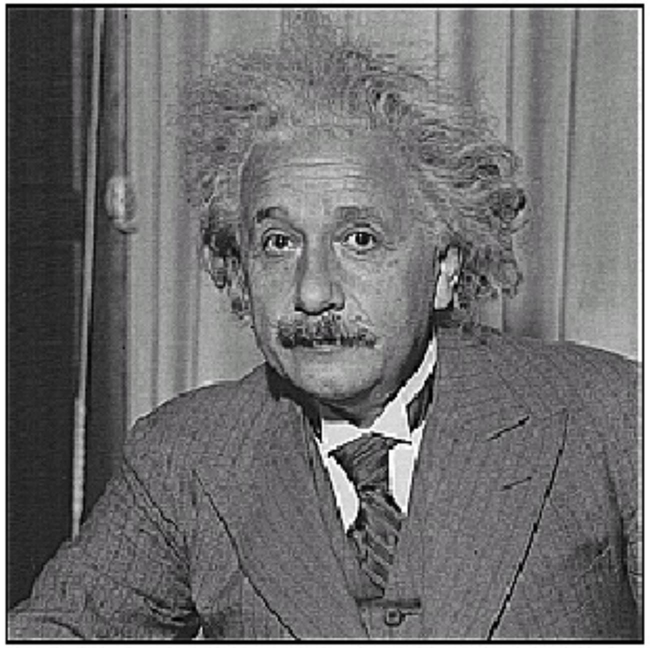
Filtering

Sharpening and Smoothing

Sharpening



before



after

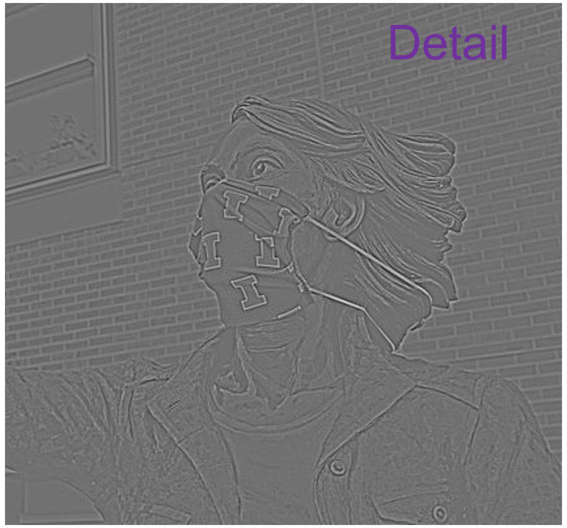
Source: D. Lowe

Sharpening

What does blurring take away?

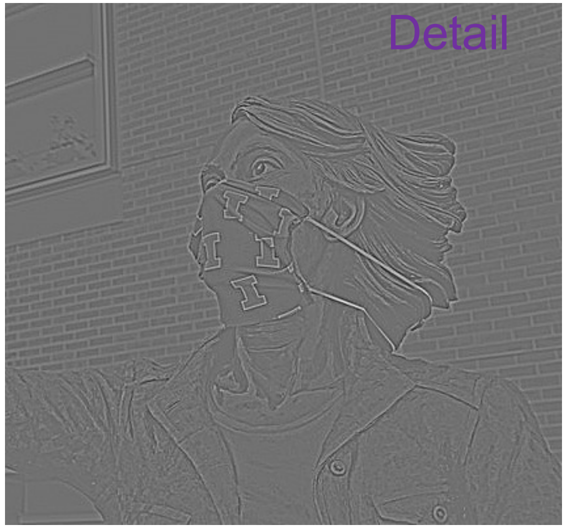
–

=



Let’s add it back in.

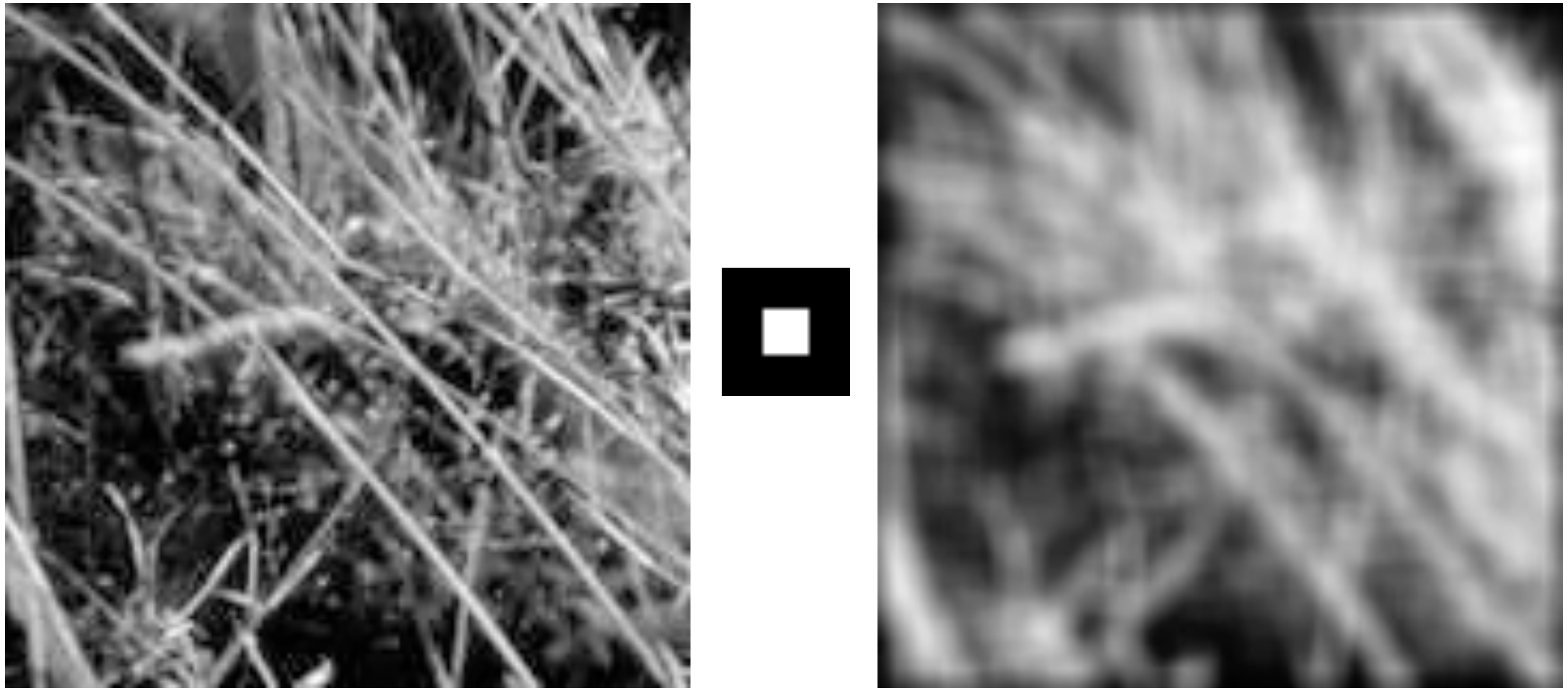
$+\alpha$



=

Revisit: Smoothing with Box Filter

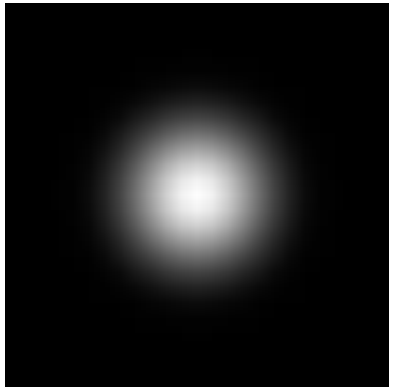
- What’s wrong with this picture?
- What’s the solution?



Source: D. Forsyth

Revisit: Smoothing with Box Filter

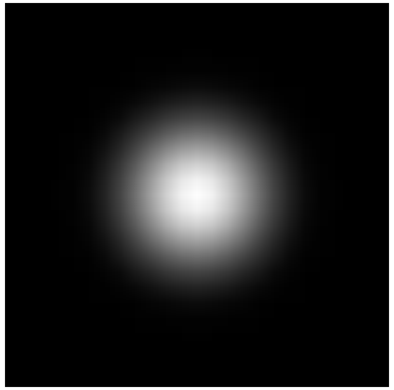
- What’s wrong with this picture?
- What’s the solution?
    - To eliminate edge effects, weight contribution of neighborhood pixels according to their closeness to the center



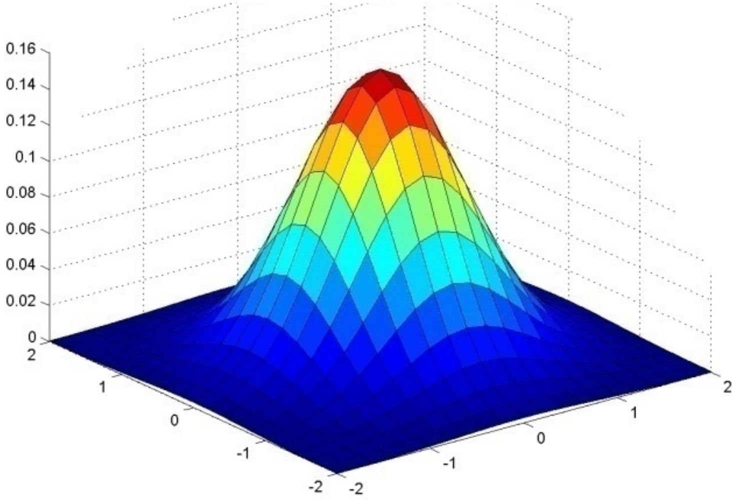
“fuzzy blob”

Credit to Lana Lazebnik

Gaussian Kernel



$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$



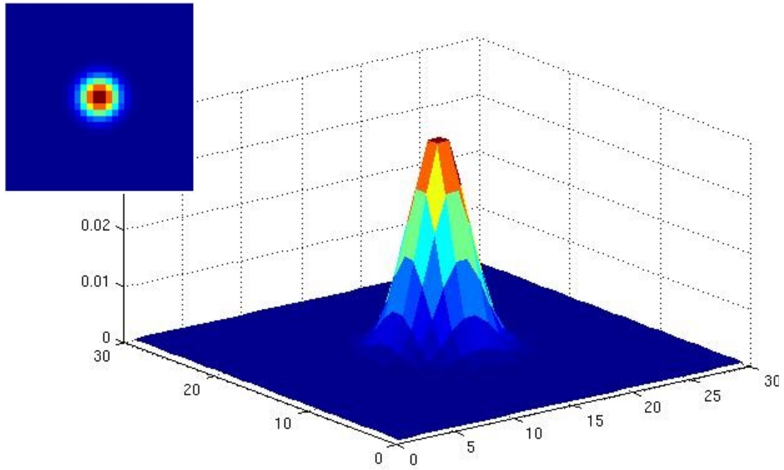
0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, σ = 1

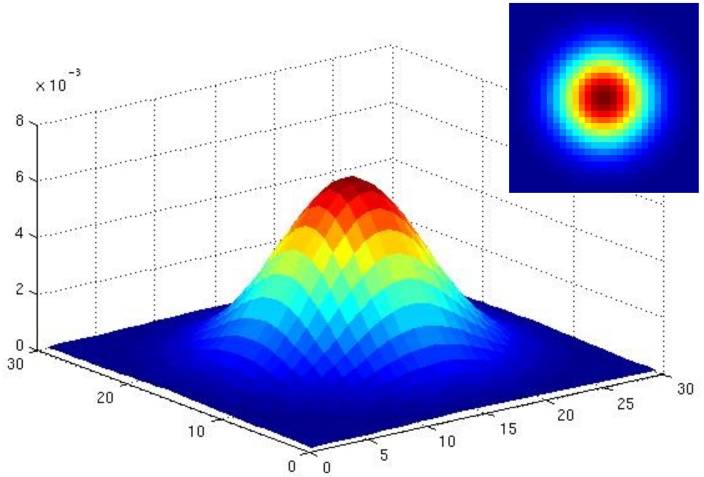
Source: C. Rasmussen

Gaussian Kernel

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$



σ = 2 with 30 x 30 kernel



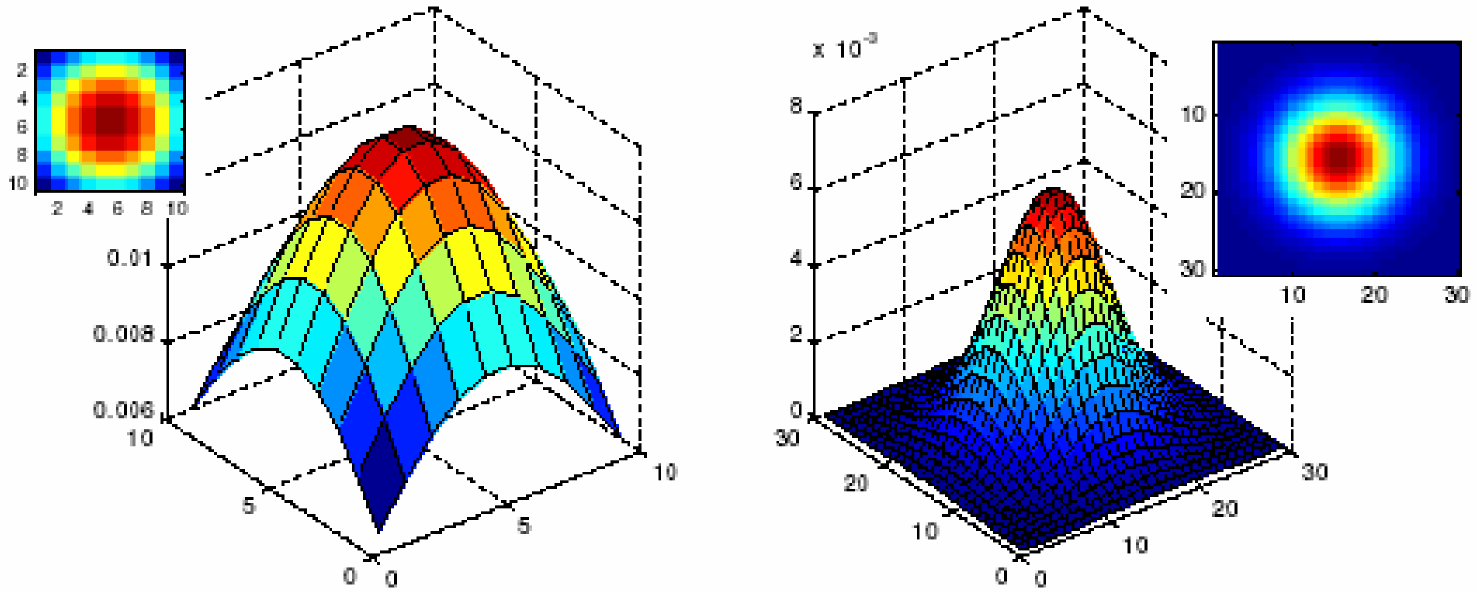
σ = 5 with 30 x 30 kernel

Standard deviation σ: determines extent of smoothing

Source: K. Grauman

Choosing Kernel Width

- The Gaussian function has infinite support, but discrete filters use finite kernels



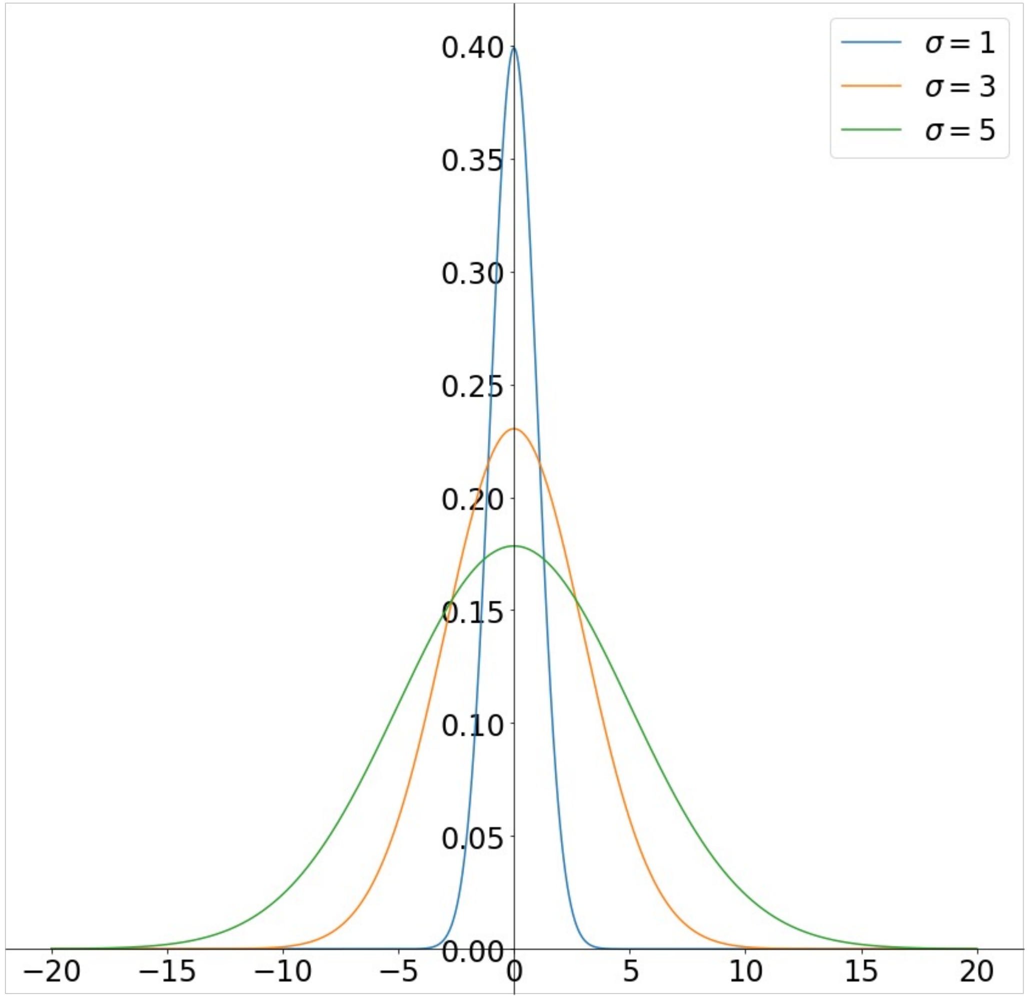
$\sigma = 5$ with 10x10 kernel

$\sigma = 5$ with 30x30 kernel

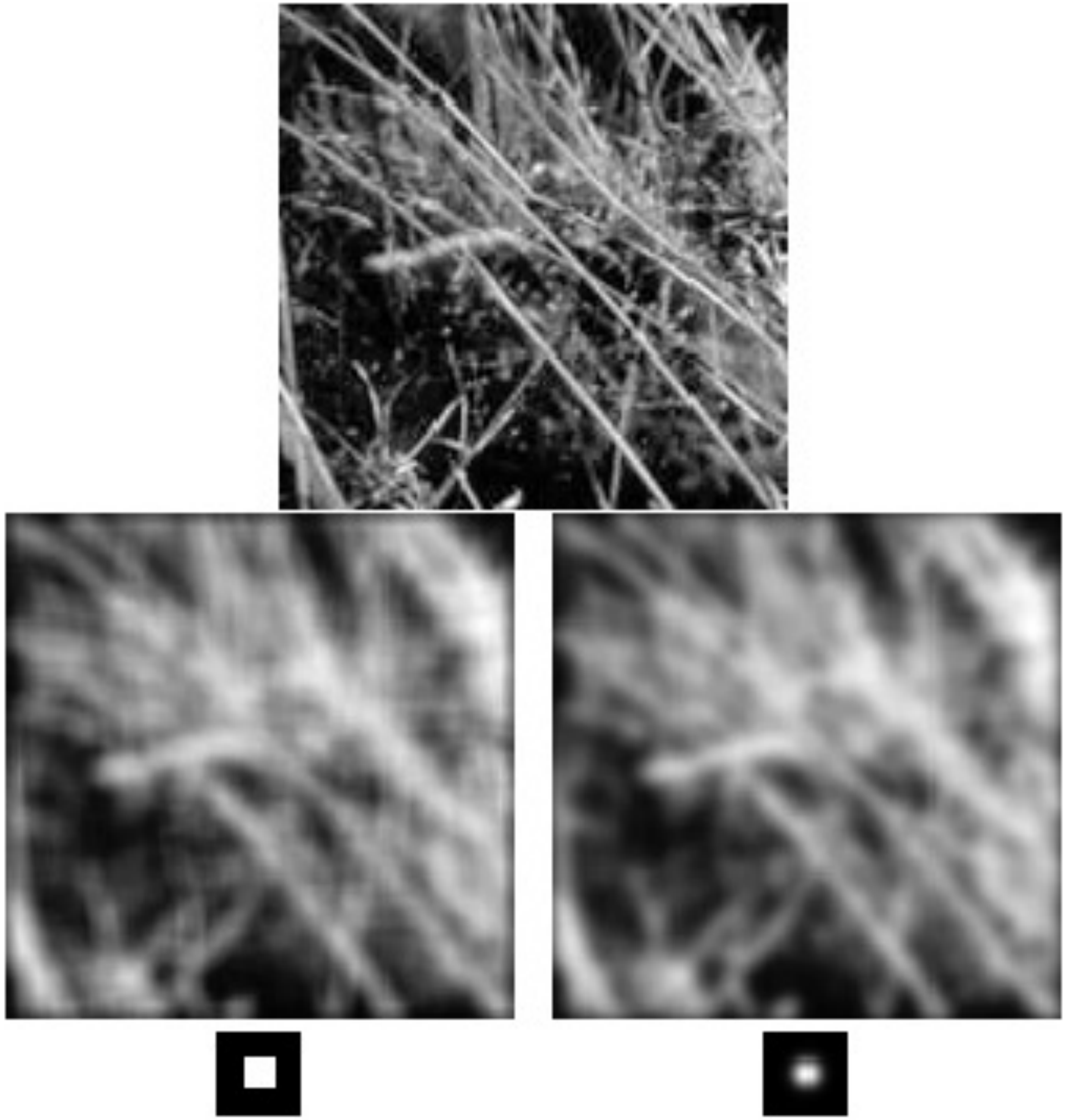
Source: K. Grauman

Choosing Kernel width

- Rule of thumb: set filter half-width to about 3σ



Gaussian vs. Box Filtering



Slide from Lana Lazebnik

Gaussian Filters

- Remove high-frequency components from the image (*low-pass filter*)
- Convolution with self is another Gaussian
  - So can smooth with small-σ kernel, repeat, and get same result as larger-σ kernel would have
  - Convolving two times with Gaussian kernel with std. dev. σ is same as convolving once with kernel with std. dev. $\sqrt{2}\sigma$
- *Separable* kernel
  - Factors into product of two 1D Gaussians
  - Discrete example:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

Source: K. Grauman

Separability of the Gaussian filter

- The 2D Gaussian can be expressed as the product of two functions, one a function of x and the other a function of y.

- In this case the two functions are the (identical) 1D Gaussian.

$$
\begin{aligned}
G_\sigma(x,y) &= \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \\
&= \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}\right)\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{y^2}{2\sigma^2}}\right)
\end{aligned}
$$

Source: D. Lowe

Why is separability useful?

- Separability means that a 2D convolution can be reduced to two 1D convolutions (one along rows and one along columns)

- What is the complexity of filtering an $n \times n$ image with an $m \times m$ kernel?
  - $O(n^2 m^2)$

- What if the kernel is separable?
  - $O(n^2 m)$

Credit to Lana Lazebnik

Filtering

Denoising

Different Types of Noise

Original

Salt and pepper noise

Impulse noise

Gaussian noise

- **Salt and pepper noise**: contains random occurrences of black and white pixels

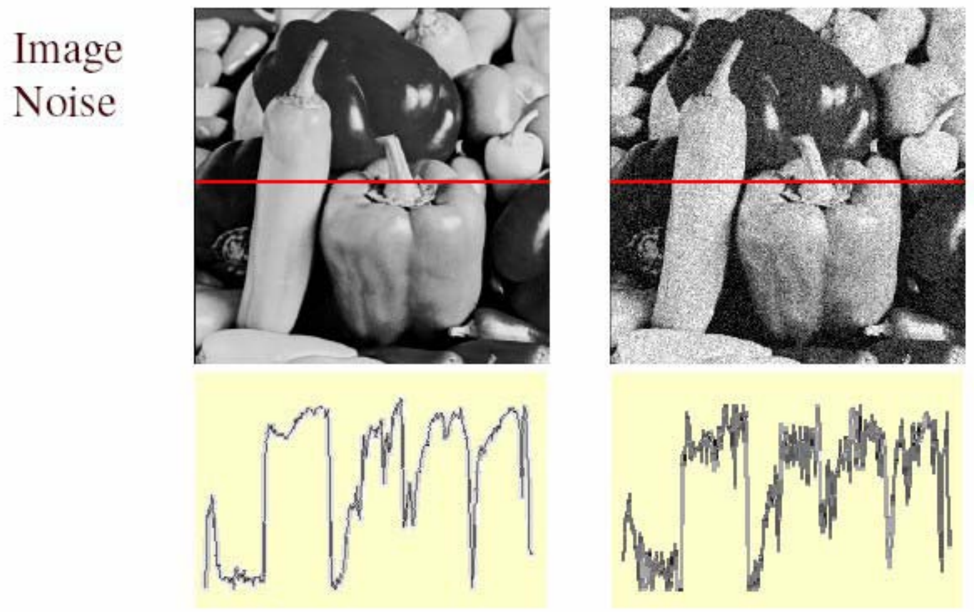
- **Impulse noise:** contains random occurrences of white pixels

- **Gaussian noise**: variations in intensity drawn from a Gaussian normal distribution

Source: S. Seitz

Gaussian Noise

- Mathematical model: sum of many independent factors
- Good for small standard deviations
- Assumption: independent, zero-mean noise

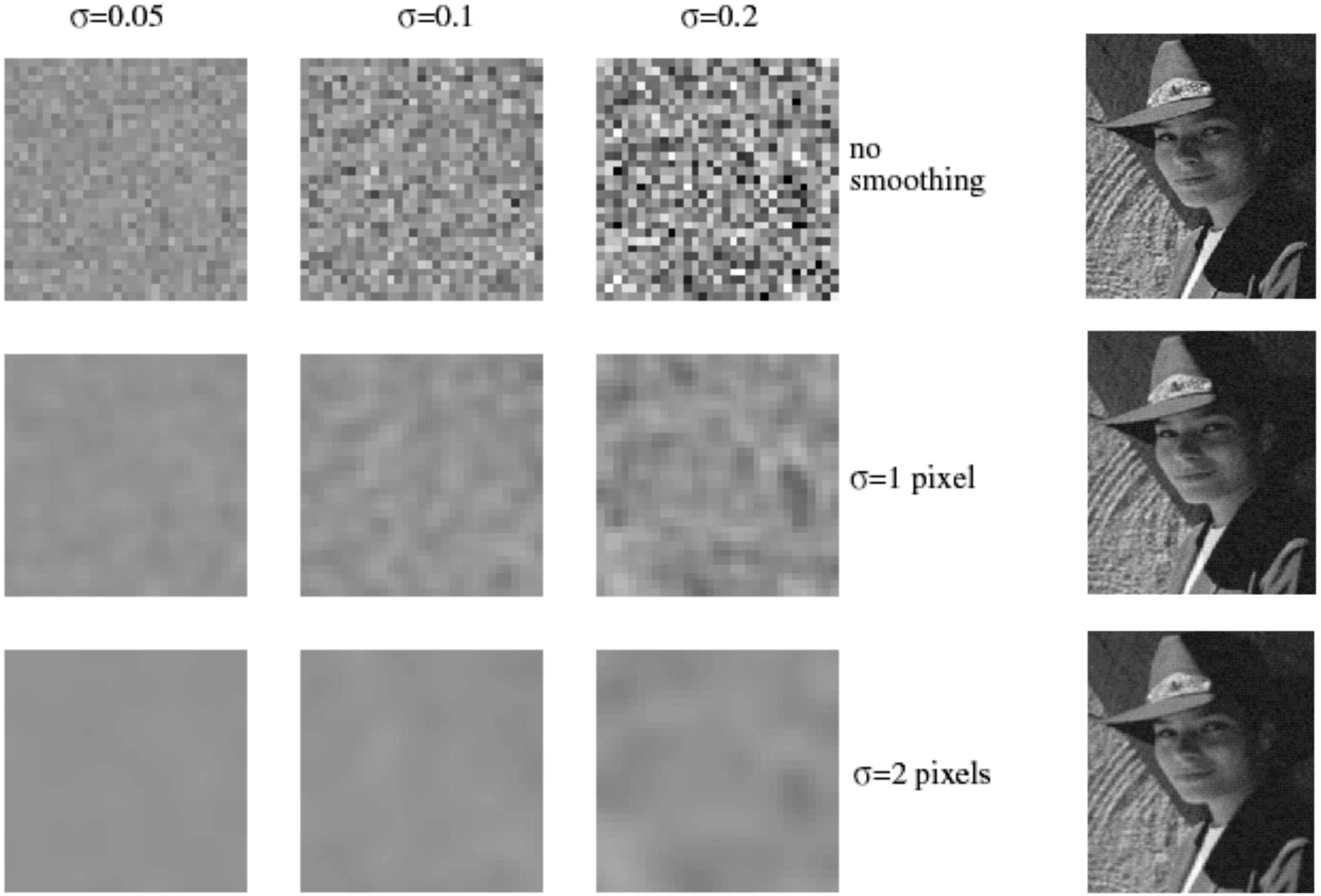


$$f(x,y) = \overbrace{\bar{f}(x,y)}^{\text{Ideal Image}} + \overbrace{\eta(x,y)}^{\text{Noise process}}$$

Gaussian i.i.d. ("white") noise:
$\eta(x,y) \sim \mathcal{N}(\mu, \sigma)$

Source: M. Hebert

Reducing Gaussian Noise



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

Credit to Lana Lazebnik

Reducing Salt-and-Pepper Noise

3x3

5x5

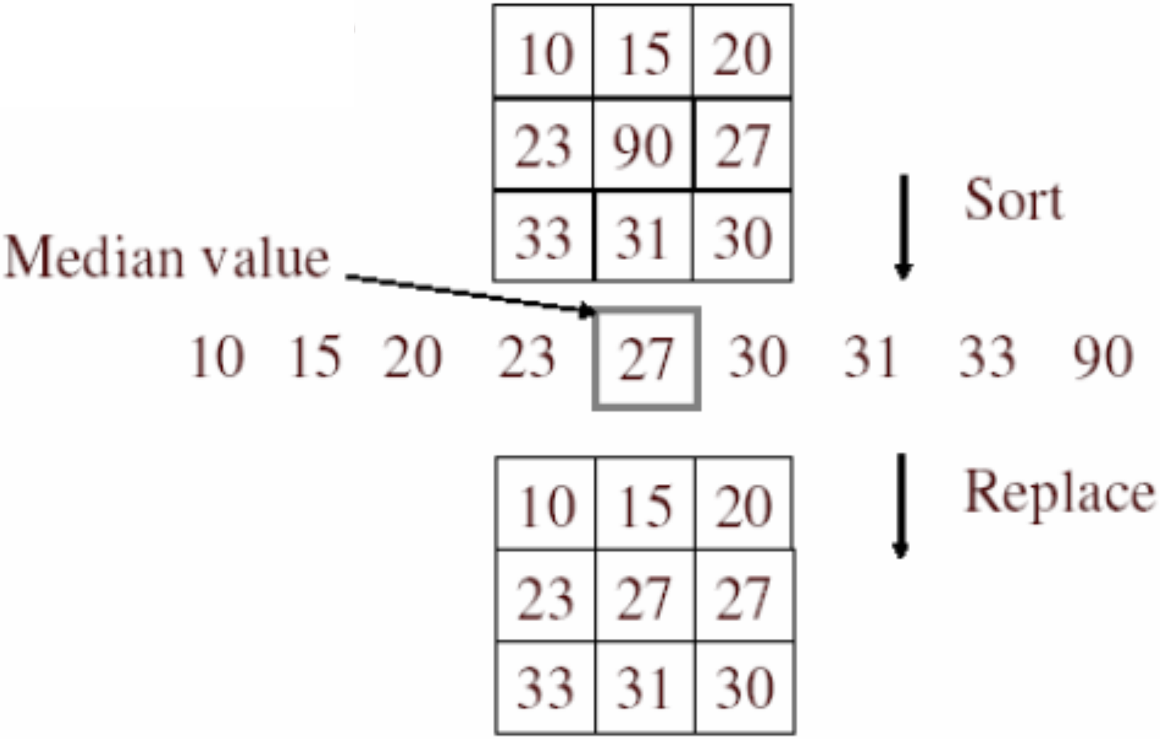
7x7

What’s wrong with the results?

Credit to Lana Lazebnik

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

Is median filtering linear?

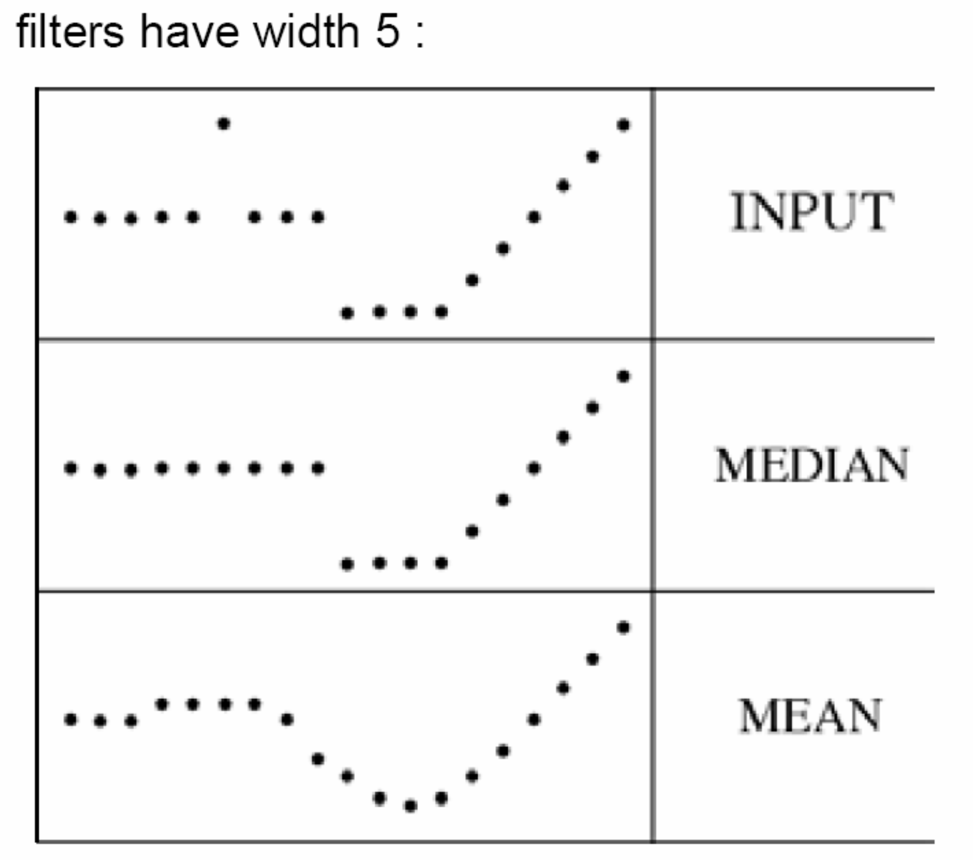
- Let's check linearity

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 2 & 2 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Credit to Lana Lazebnik

Median Filter

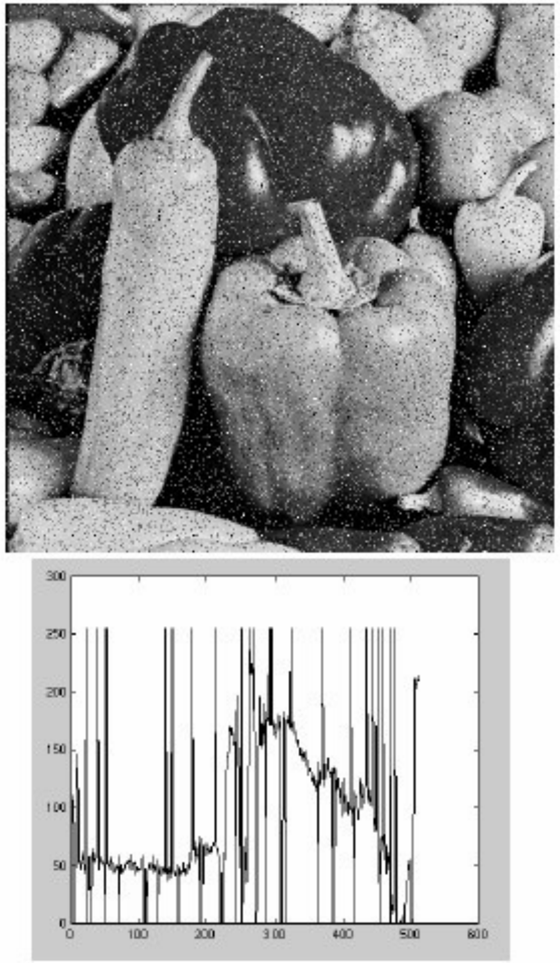
- What advantage does median filtering have over Gaussian filtering?
  - Robustness to outliers



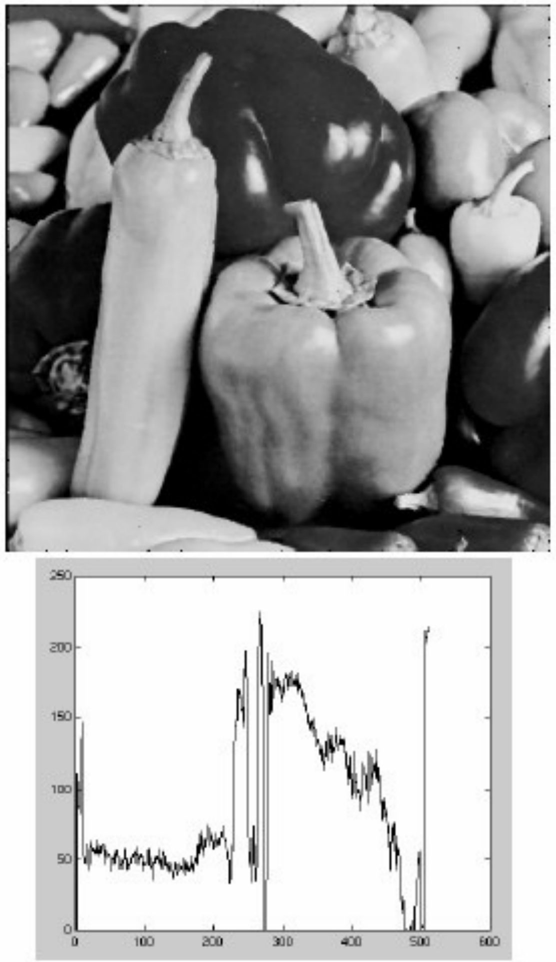
Source: K. Grauman

Median Filter

Salt-and-pepper noise



Median filtered



Source: M. Hebert

Gaussian vs. Median Filtering

3x3 5x5 7x7

Gaussian

Median

Credit to Lana Lazebnik