Smoothing

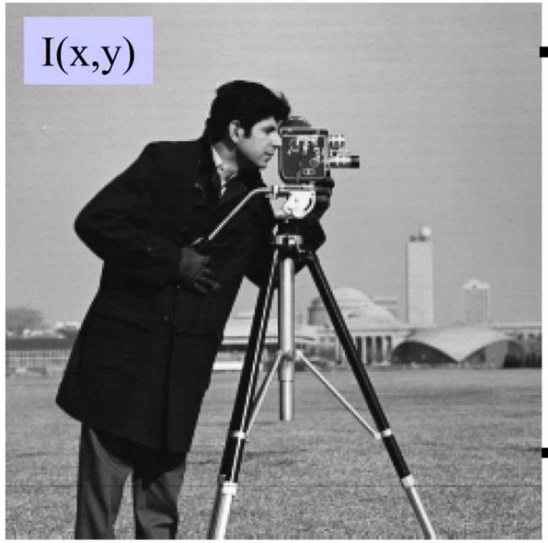
CS 4391 Introduction Computer Vision

Professor Yu Xiang

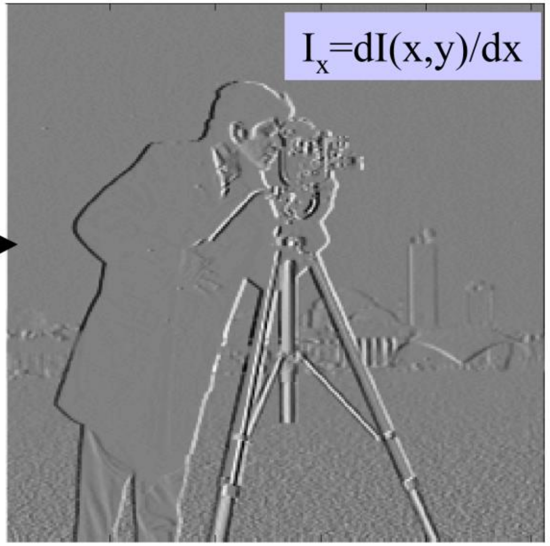
The University of Texas at Dallas

Recall Image Gradient

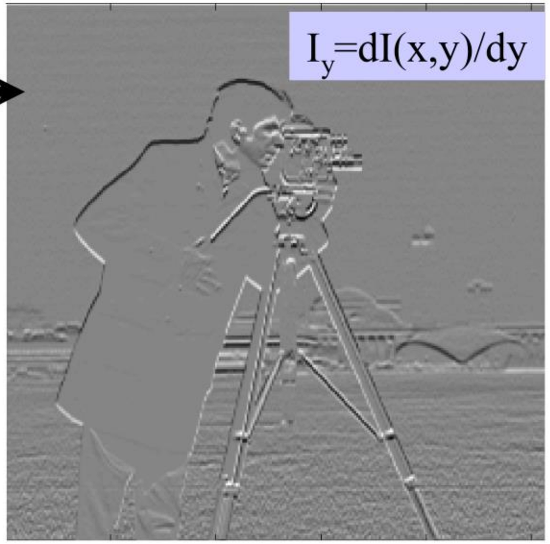
$$I_x = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} * I$$



Partial derivative wrt x



Partial derivative wrt y



$$I_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} * I$$

Note that there is a difference between convolving with a 1xn row filter and an nx1 col filter.

Image Gradient and Noise

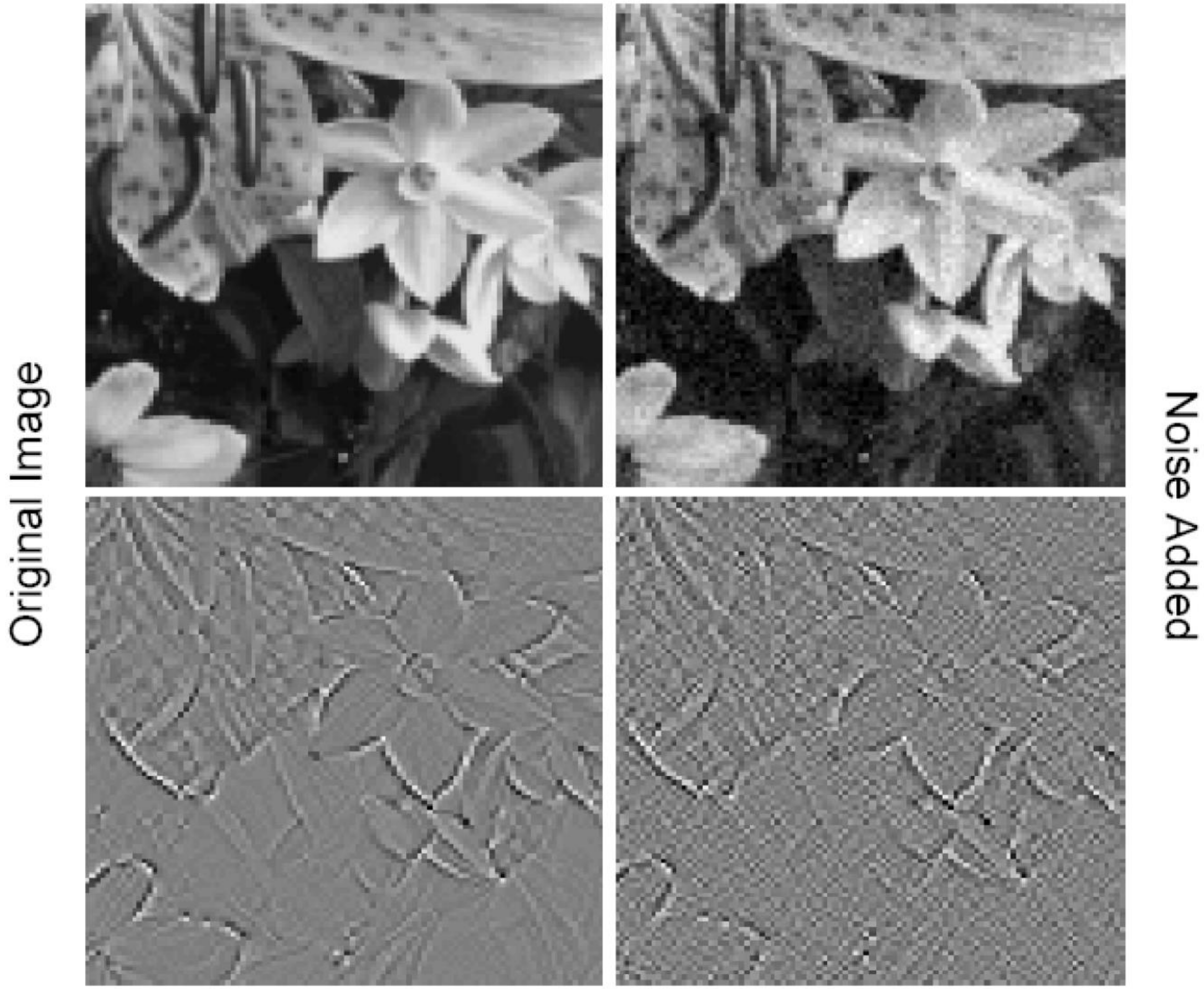
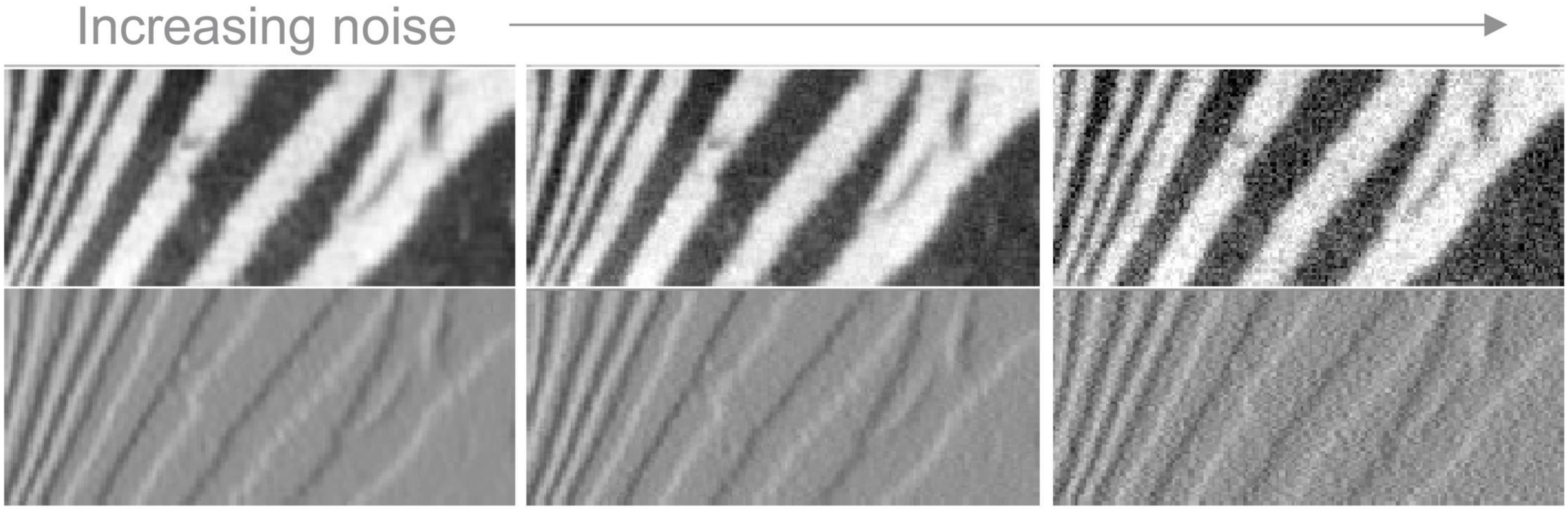


Image Gradient and Noise

- First derivative operator is affected by noise



Numerical derivatives can amplify noise

Image Noise

- Fact: images are noises

- Examples:
  - Light fluctuations
  - Sensor noise
  - Quantization effect
  - Finite precision

Modeling Image Noise

- Additive random noise

$$I(x, y) = s(x, y) + n_i$$

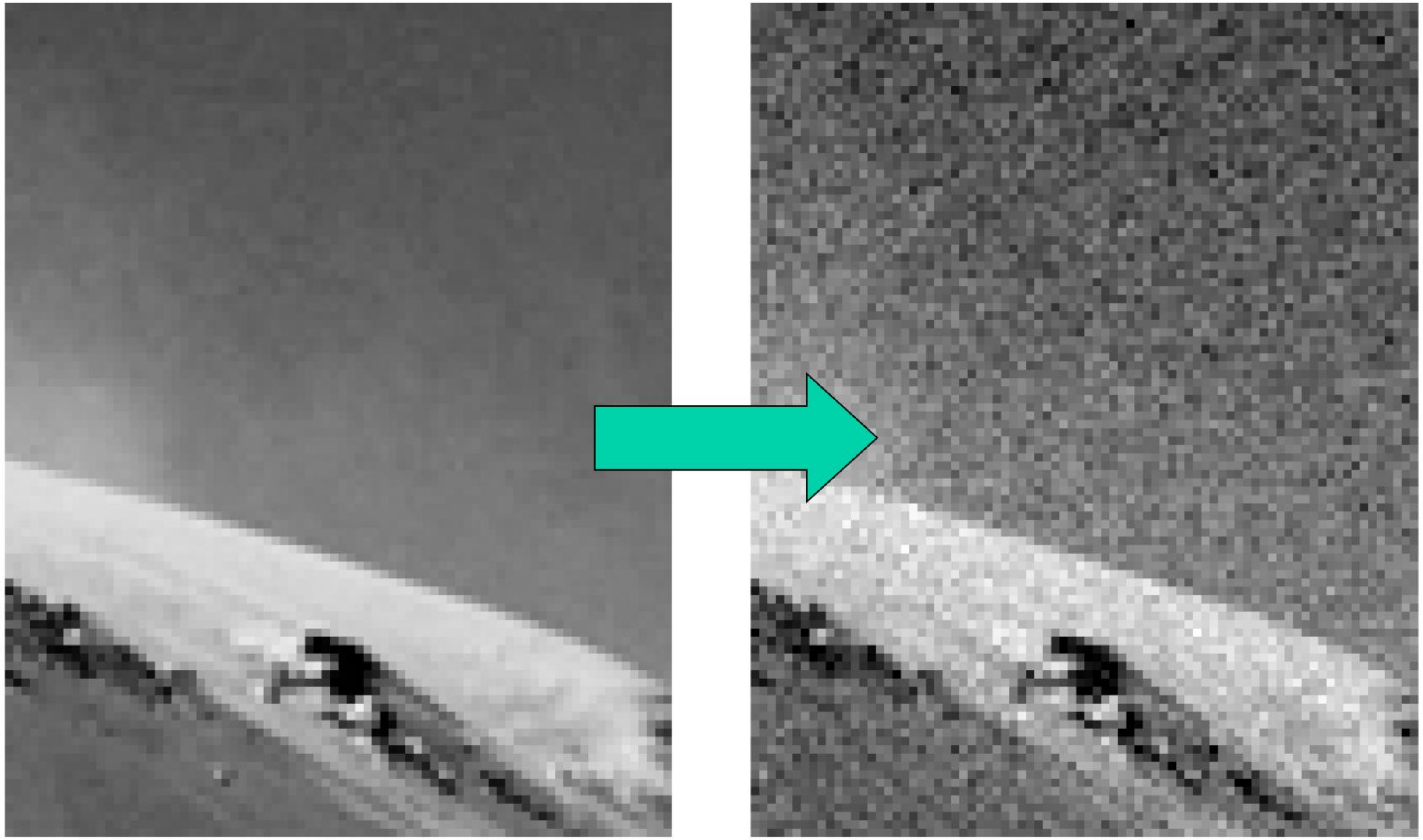
- n_i is i.i.d (independent and identically distributed)
- Zero-mean gaussian noise $\mathcal{N}(0, \sigma^2)$

Gaussian distribution
(normal distribution)

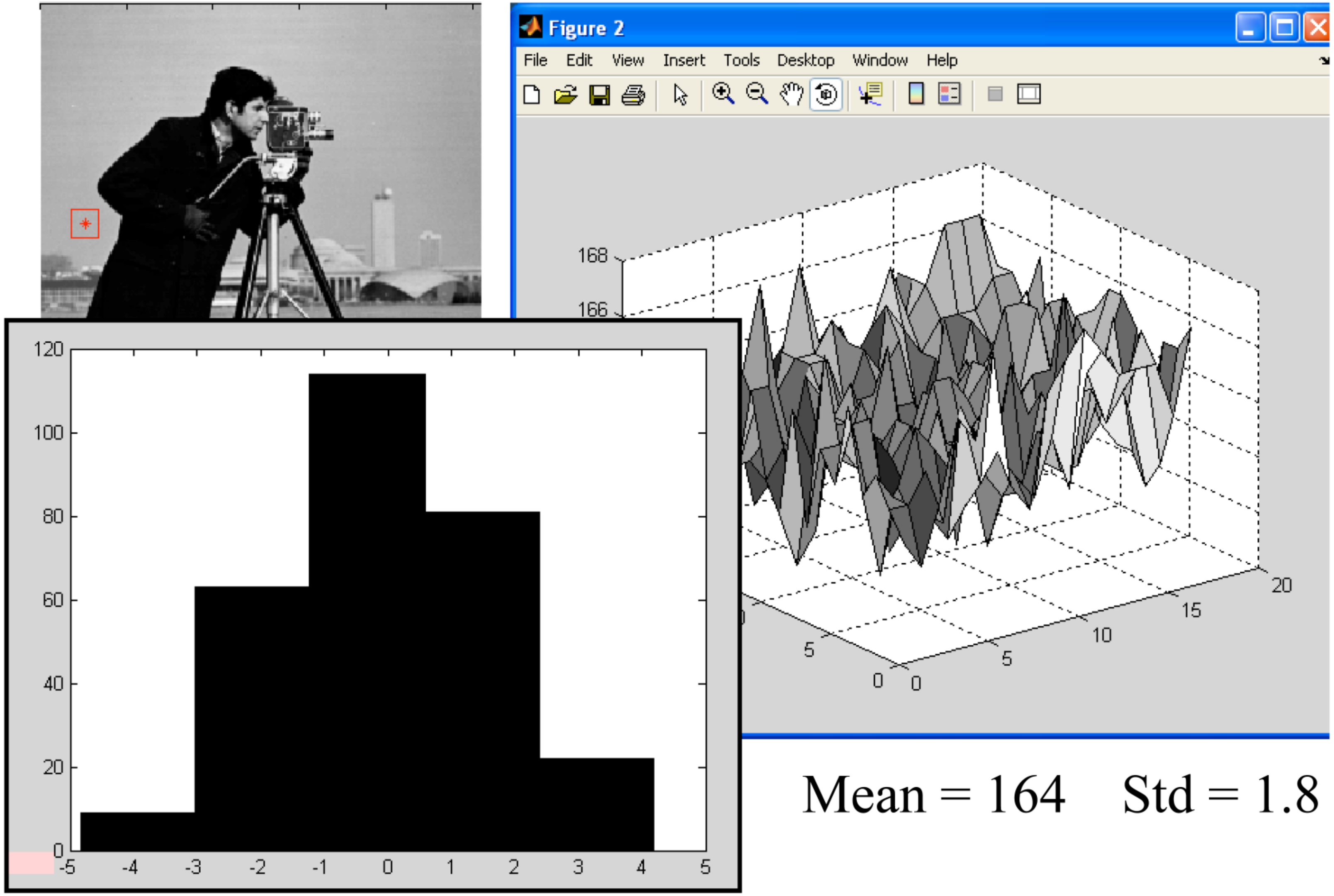
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Modeling Image Noise

mean 0, sigma = 16



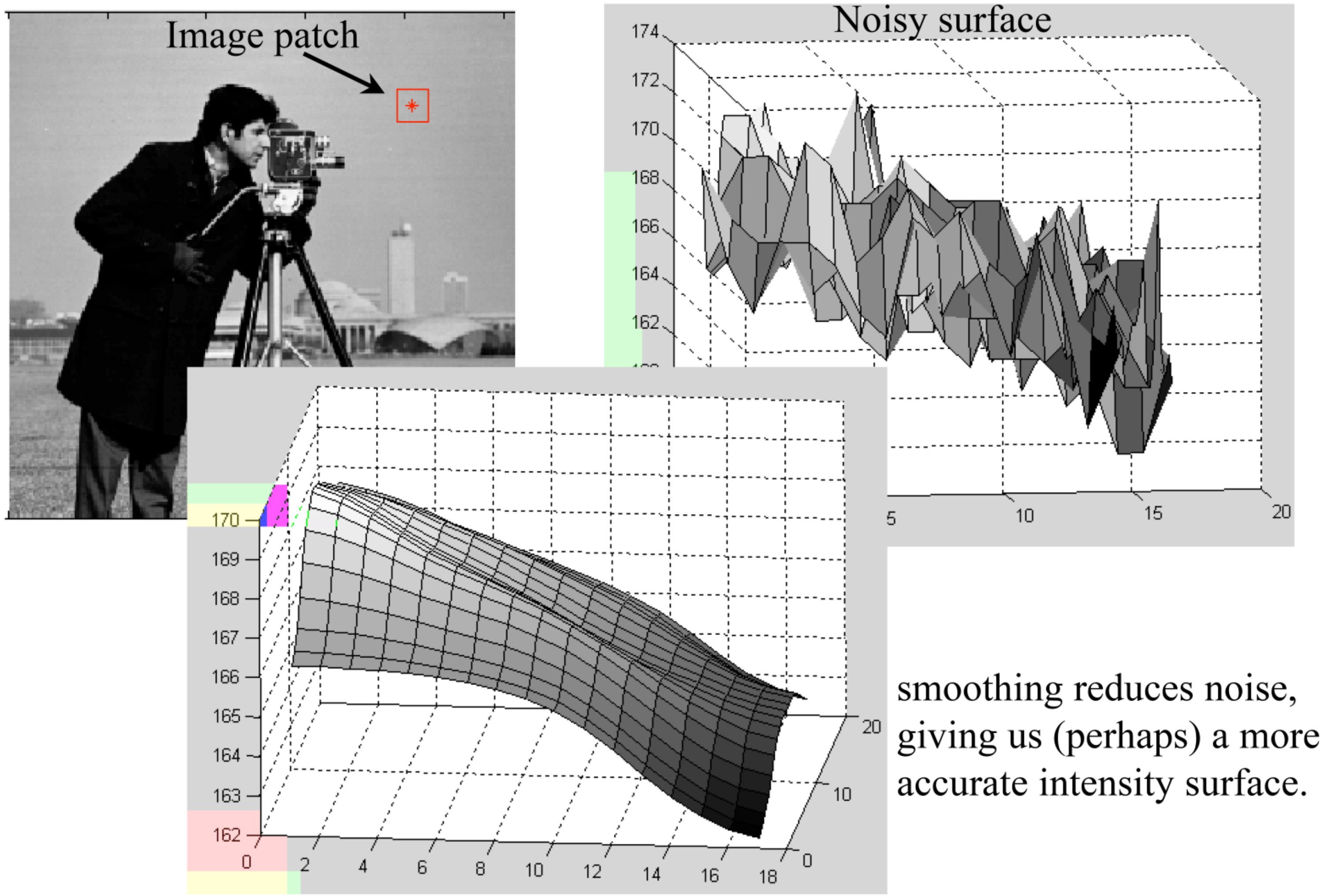
Empirical Evidence



Mean = 164 Std = 1.8

Smoothing

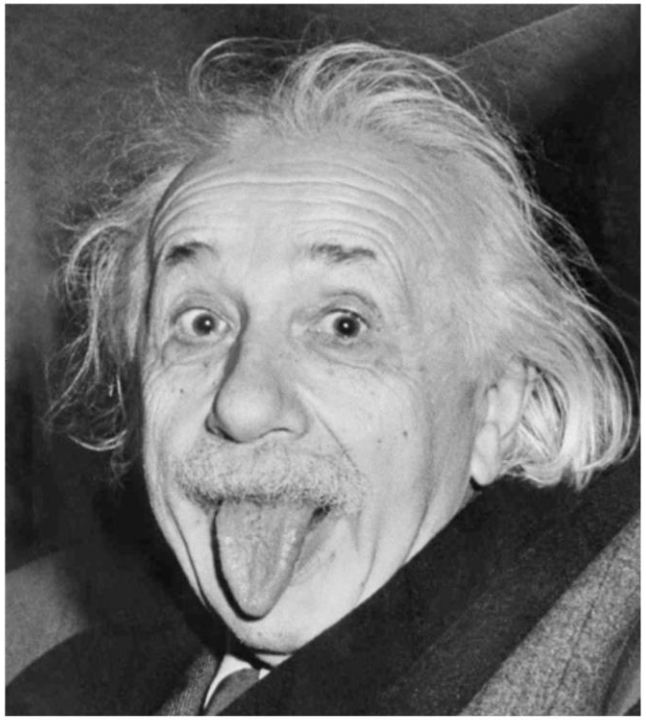
- Reduce noise

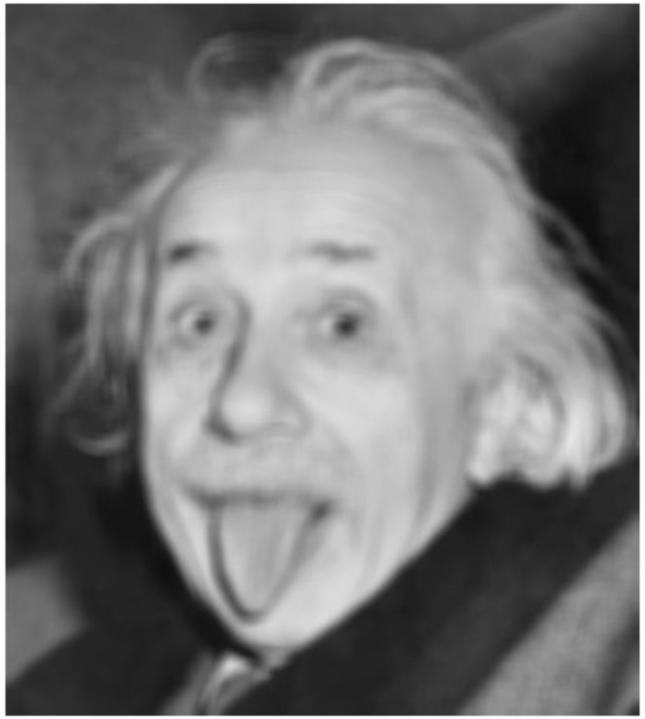


Box Filter

- Replace a pixel with a local average (smoothing)

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





Why Average Reduces Noise

- Intuitive explanation: variance of noise in the average is smaller than variance of the pixel noise (assuming zero-mean Gaussian noise).

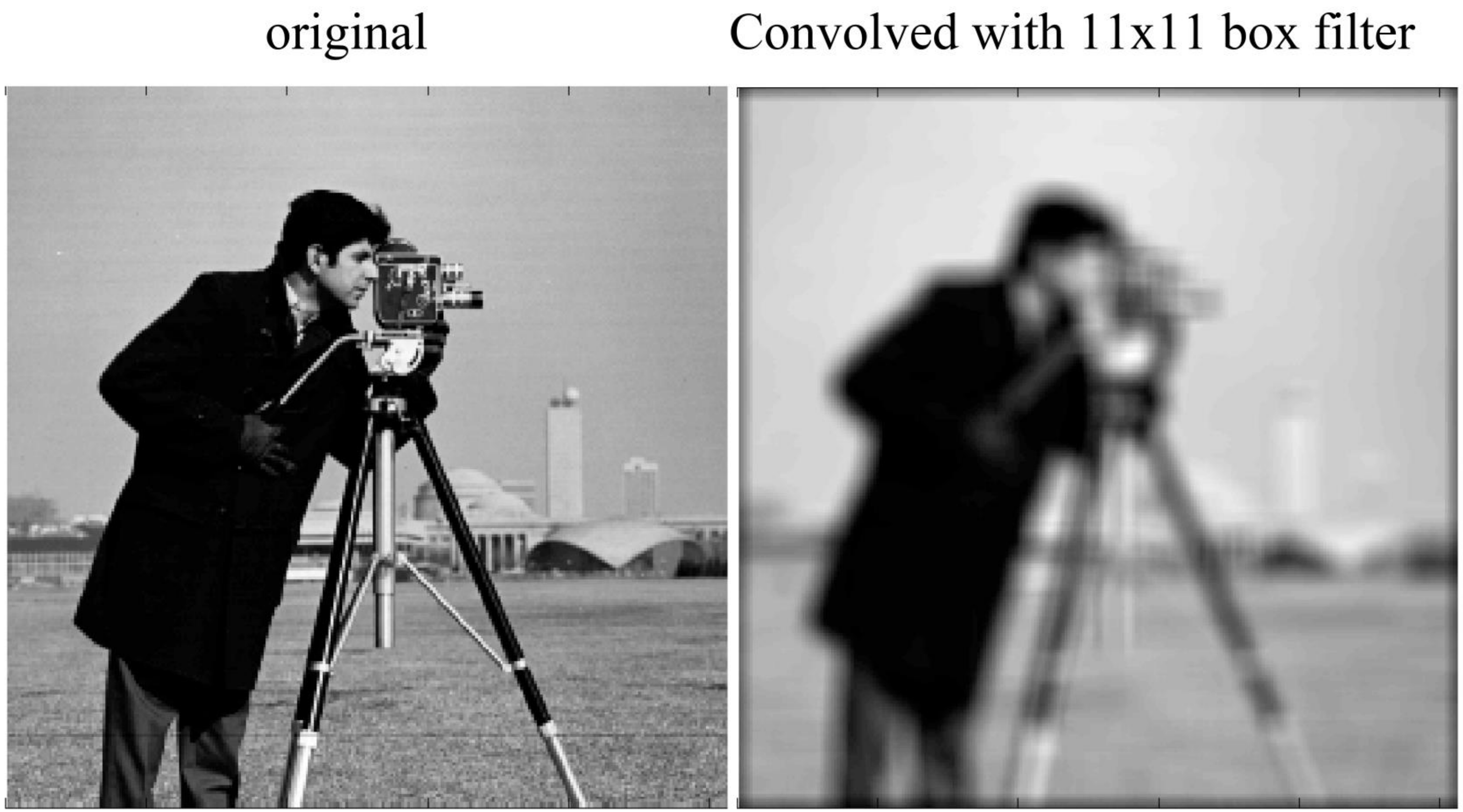
$$A = \frac{1}{m^2} \sum_{i=1}^{m^2} I_m$$

$I_m = s_m + n_m$ with n being i.i.d. $G(0, \sigma^2)$

$$E(A) = \frac{1}{m^2} \sum s_m$$

$$var(A) = E\left[(A - E(A))^2\right] = \frac{\sigma^2}{m}$$

Smoothing with Box Filter



Drawback: smoothing reduces fine image detail

Needs to balance smoothing and keep image gradient

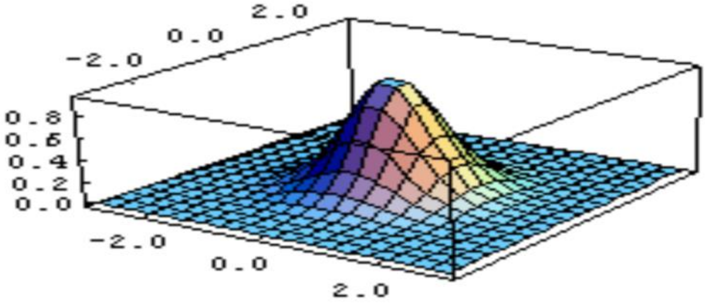
Gaussian Filter

- A case of weighted averaging

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- The weights are from a 2D Gaussian distribution

$$h(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{\sigma^2}}$$

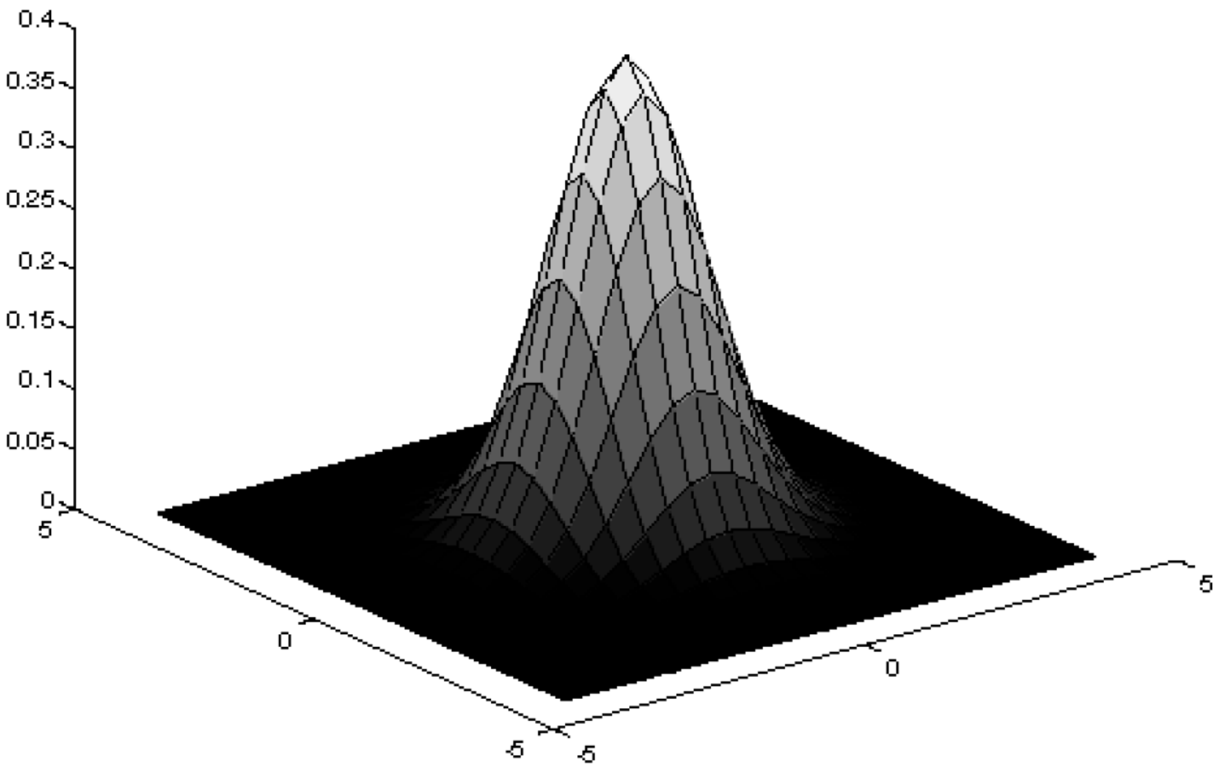


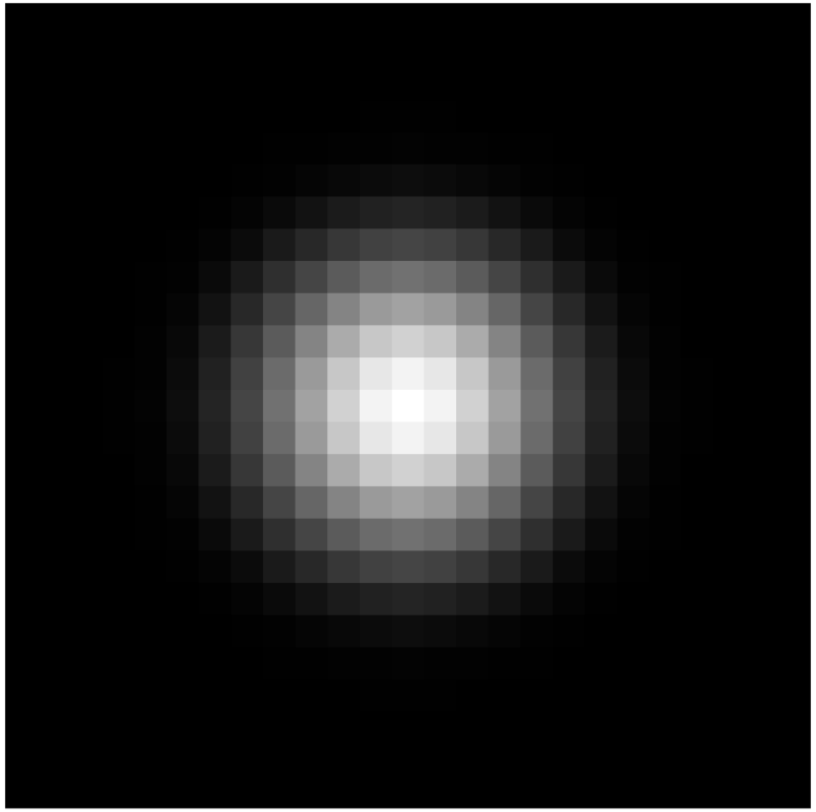
- Gives more weight at the central pixels and less weights to the neighbors

- The farther away the neighbors, the smaller the weight

Gaussian Filter

- An isotropic (circularly symmetric) Gaussian





Gaussian Smoothing Example

original

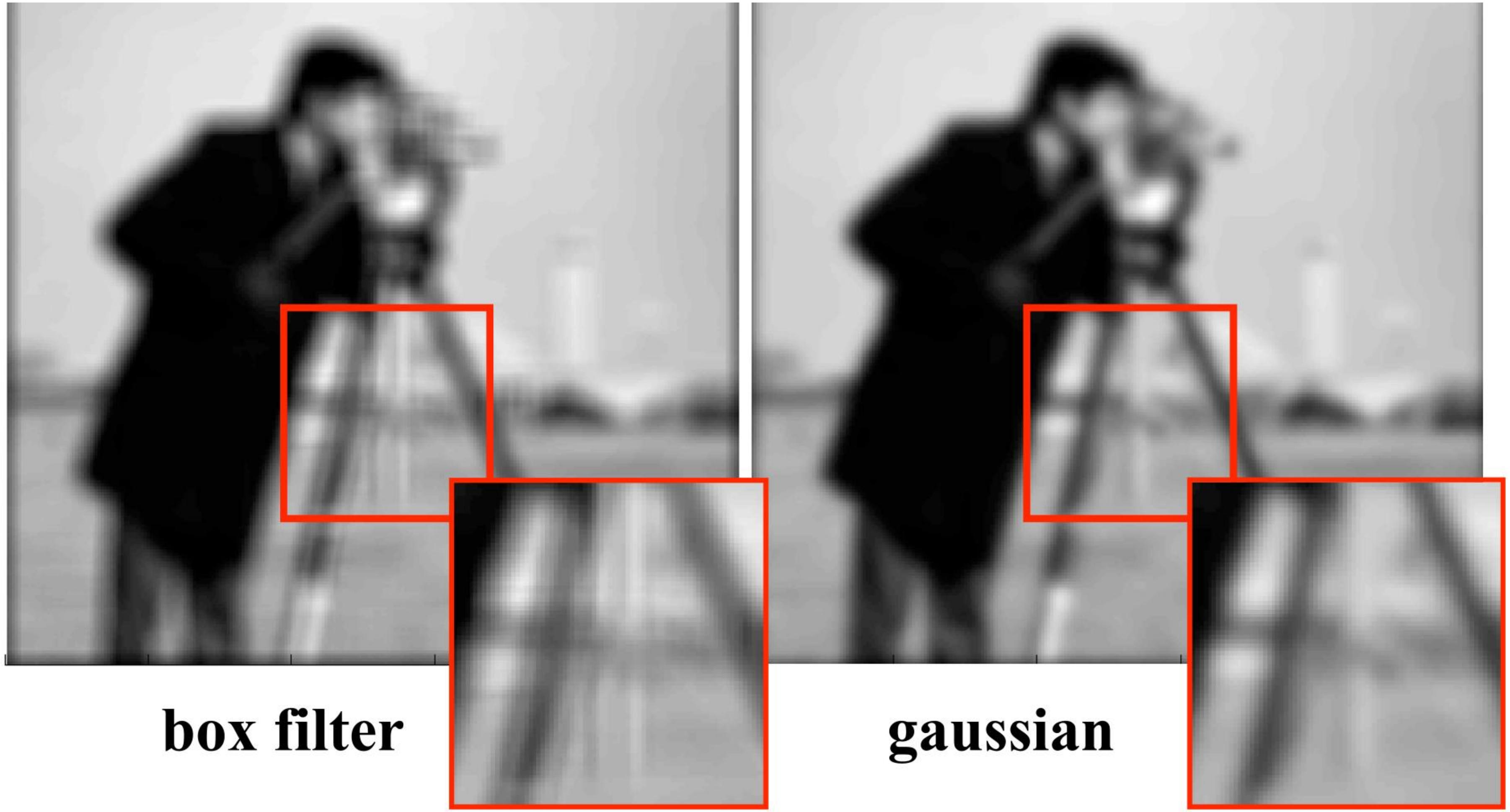
sigma = 3

Box vs. Gaussian

box filter

gaussian

Box vs. Gaussian



Note: Gaussian is a true low-pass filter, so won't cause high frequency artifacts

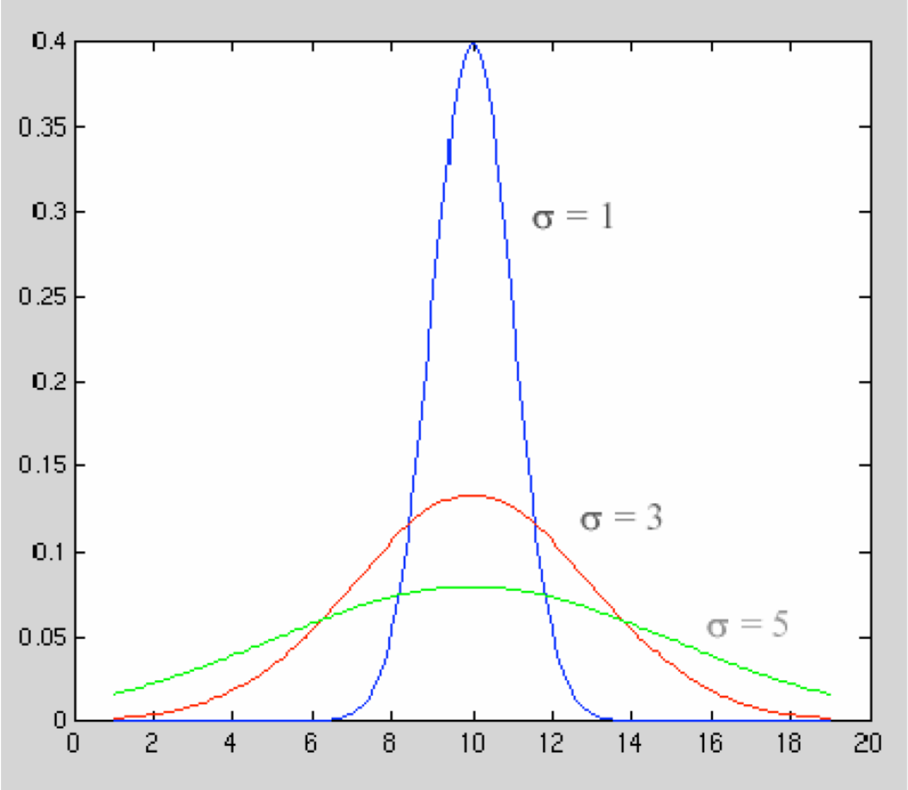
Gaussian Smoothing at Different Scales

- The std. dev of the Gaussian determines the amount of smoothing

- Gaussian theoretically has infinite support, but we need a filter of finite size.

- For a 98.76% of the area, we need +/-2.5 σ

- +/- 3σ covers over 99% of the area.



Gaussian Smoothing at Different Scales

Standard deviation σ

- Pixels at a distance of more than 3σ are small
- Typical filter dimension $\lceil 6\sigma \rceil \times \lceil 6\sigma \rceil$
- Large σ , large filter size

$$h(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{\sigma^2}}$$

Gaussian Smoothing at Different Scales

original

sigma = 1

Gaussian Smoothing at Different Scales

original

sigma = 3

Gaussian Smoothing at Different Scales

original

sigma = 10

Further Reading

- Chapter 3.2, 3.3, Richard Szeliski
- Multivariate normal distribution https://en.wikipedia.org/wiki/Multivariate_normal_distribution
- OpenCV image smoothing https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_filtering/py_filtering.html