CSC589 Introduction to Computer Vision
Lecture 3

Gaussian Filter, Histogram Equalization

Bei Xiao

Last lecture

- Image can represented as a matrix
- Point process
- Linear filter: convolution

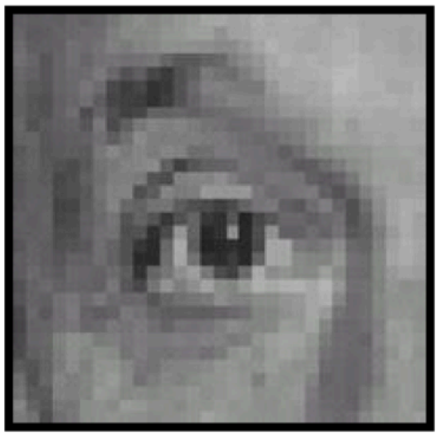
Take-home assignments

- Chapter 3.2 on linear filtering
- Image Histogram Equalization (pdf will be uploaded in blackboard)
- Chapter 1 of Solem (Computer vision with Python). Many useful examples
- Homework will be out this weekend and due a week.

Today's lecture

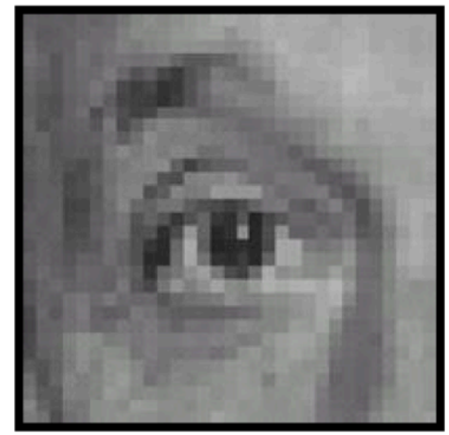
- More on linear filter, Image Sharpening
- Gaussian Filter (introduction)
- Image Histogram Equalization
- Basic image processing tutorial

Practice with linear filters



Original

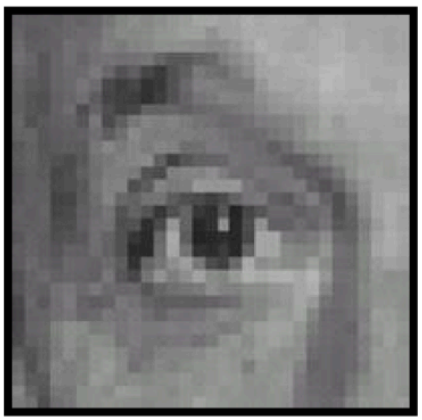
0	0	0
0	1	0
0	0	0



Filtered
(no change)

Source: D. Lowe

Practice with linear filters



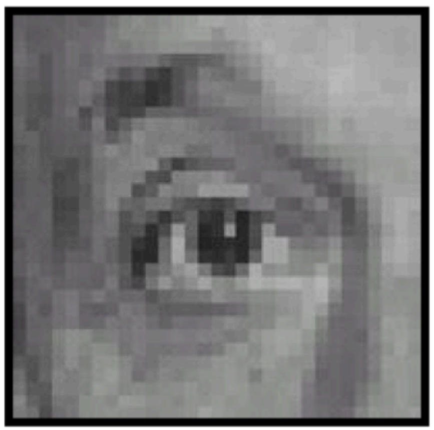
Original

0	0	0
0	0	1
0	0	0

?

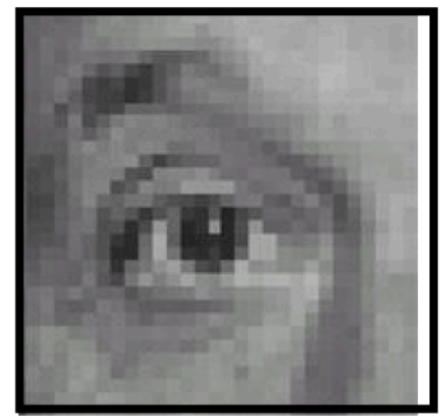
Source: D. Lowe

Practice with linear filters



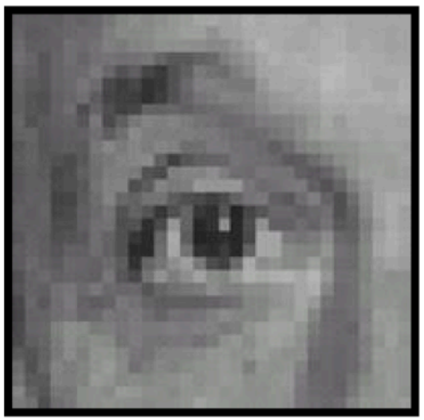
Original

0	0	0
0	0	1
0	0	0



Shifted left
By 1 pixel

Practice with linear filters



Original

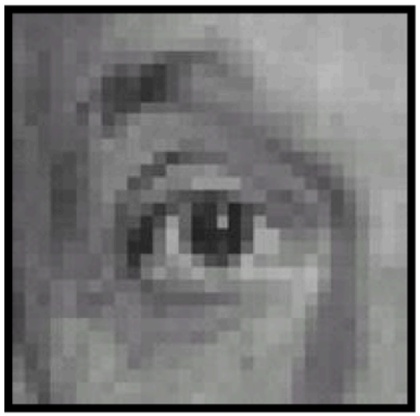
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

?

(Note that filter sums to 1)

Source: D. Lowe

Practice with linear filters



Original

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

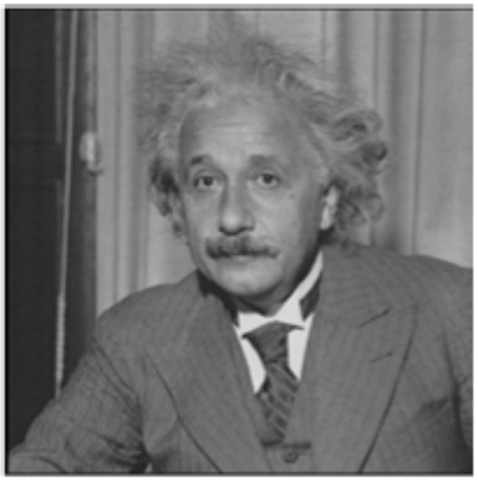
Sharpening filter

- Accentuates differences with local average

Source: D. Lowe

Image Sharpening

- Take this image and blur it



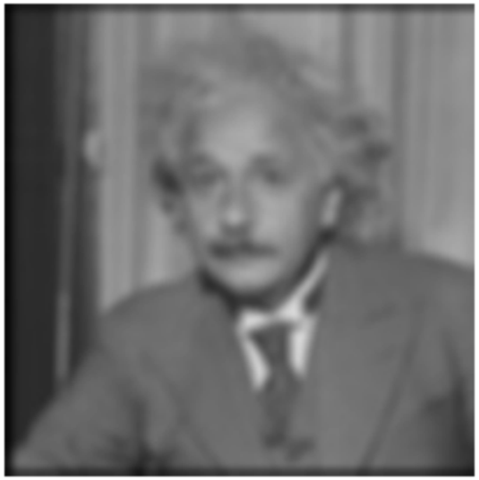
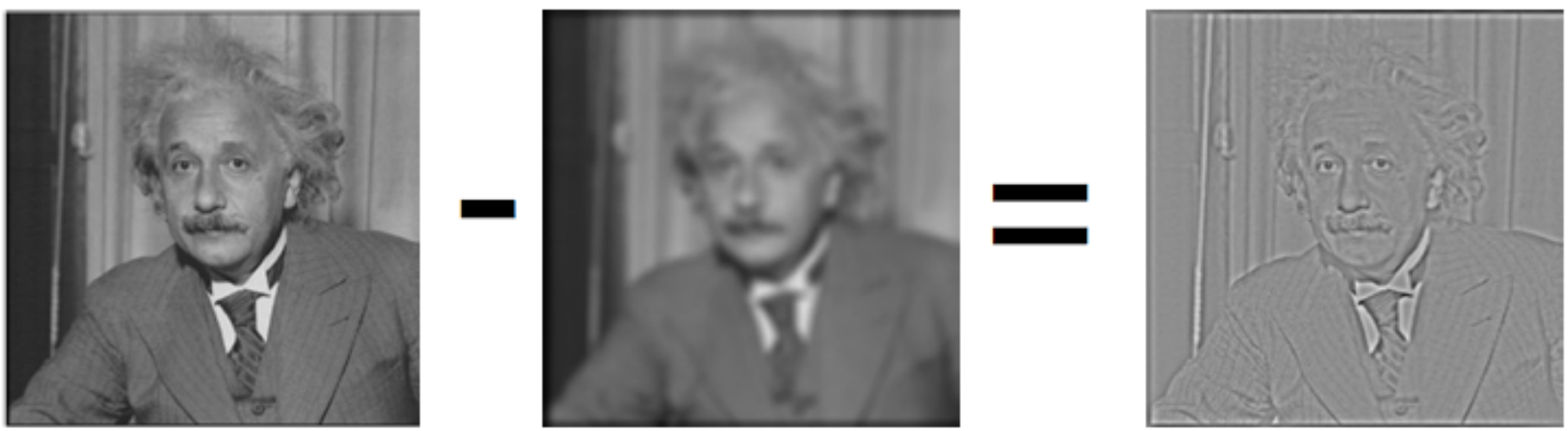


Image Sharpening

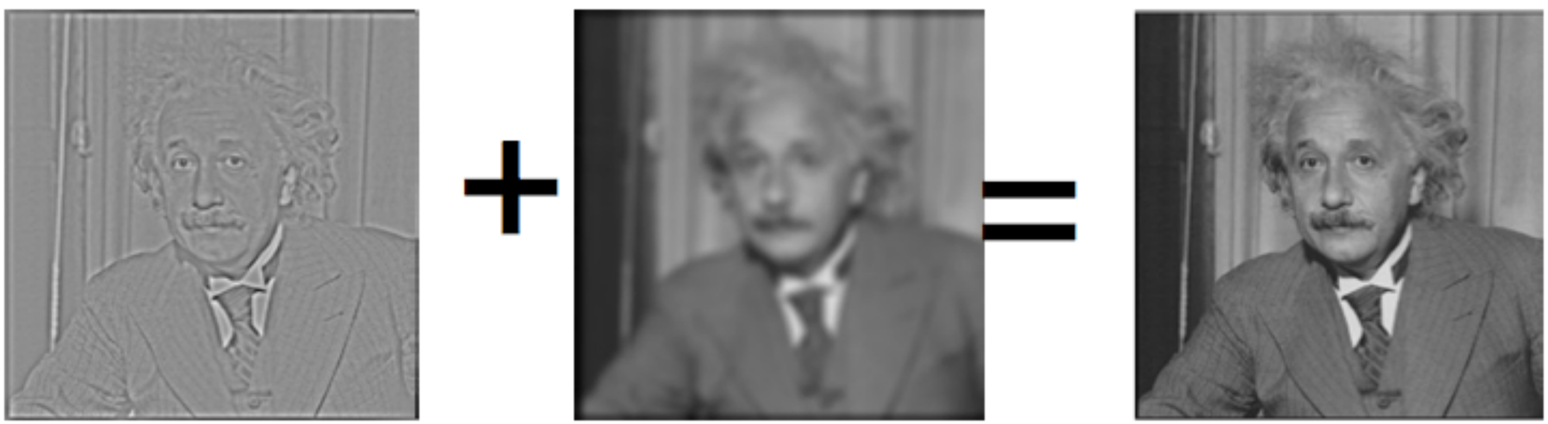
- What do we get if we subtract this two?



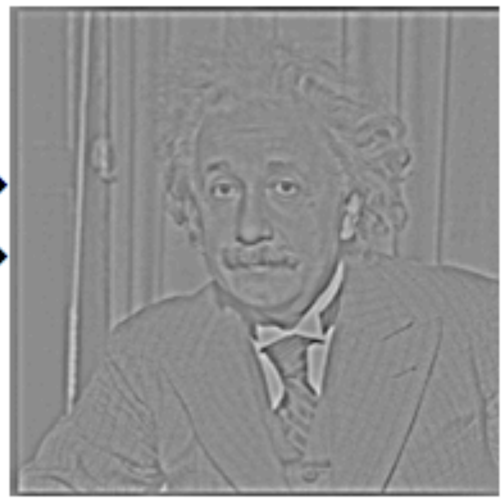
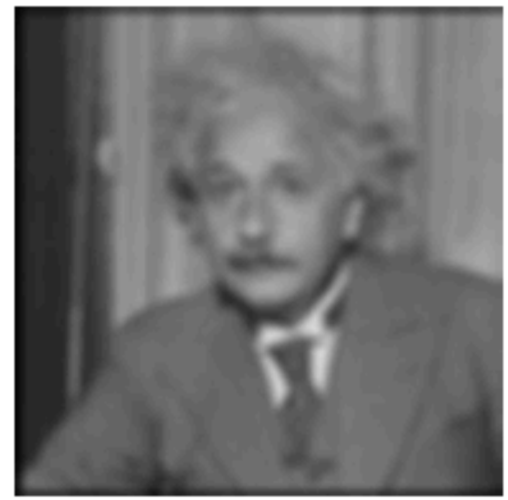
This is the left-over sharp stuff!

Let's make the image sharper?

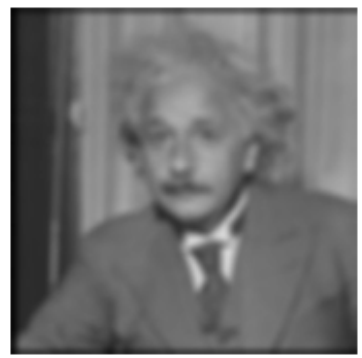
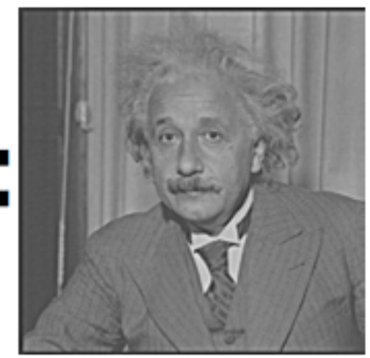
- We know: "sharp stuff + blurred = original"



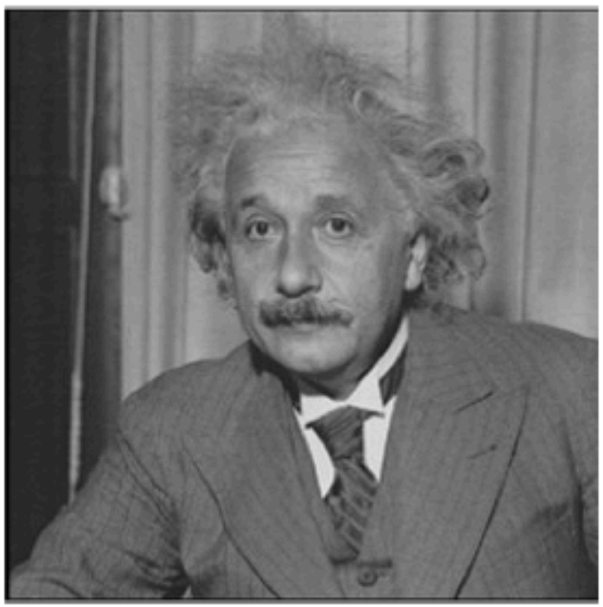
Let’s boost the sharp stuff a little

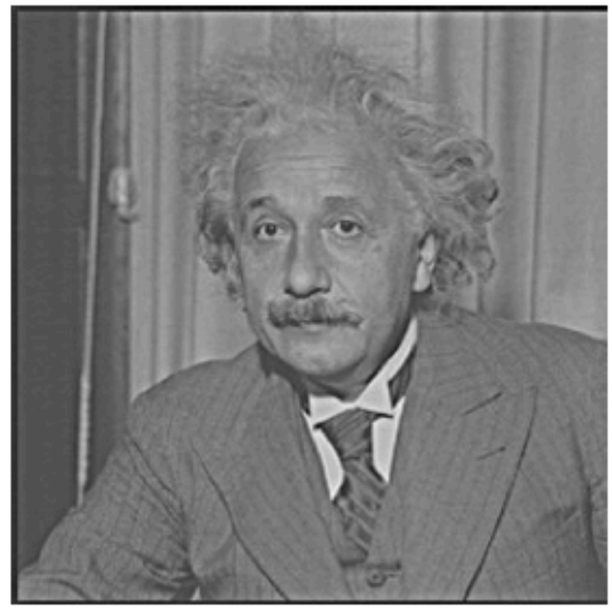
$2\times$  $+$ 

Let’s boost the sharp stuff a little

$2\times$  $+$  $=$ 

Side by Side





Now look at the computation

- Operations
  - 1 convolution
  - 1 subtraction over the whole image
- As an equation:

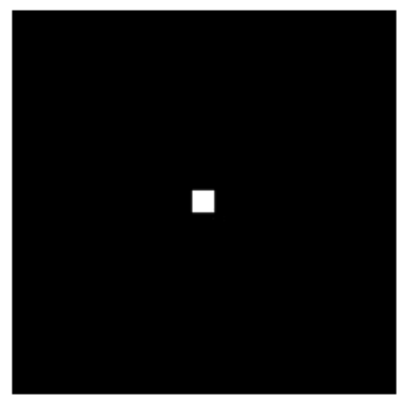
$$\mathcal{I} * f + 2\left(\mathcal{I} - \mathcal{I} * f\right)$$

Rewrite this

$$\mathcal{I} * f + 2\left(\mathcal{I} - \mathcal{I} * f\right)$$

$$\mathcal{I} * f + 2\left(\mathcal{I} * \delta - \mathcal{I} * f\right)$$

This is an identity filter or unit impulse



Rewrite this

$$\mathcal{I} * f + 2\left(\mathcal{I} - \mathcal{I} * f\right)$$

$$\mathcal{I} * f + 2\left(\mathcal{I} * \delta - \mathcal{I} * f\right)$$

$$\mathcal{I} * \left(f + 2\delta - 2f\right)$$

Now look at the computation

$$\mathcal{I} * (f + 2\delta - 2f)$$

- Can pre-compute new filter
- Operations

  1 convolution

Photoshop: Unsharp Masknig

- http://en.wikipedia.org/wiki/Unsharp_masking

Source image (left) and the sharpened image (right).

Unsharp Masking (Scipy)

- alpha = 30
- im_blur = filters.gaussian_filter(im, 3)
- im_blur2 = filters.gaussian_filter(im_blur,1)
- im_sharpened = im_blur + alpha * (im_blur - im_blur2)

Unsharp Masking (Scikit)

```
from skimage import filter
from skimage import img_as_float
import matplotlib.pyplot as plt


unsharp_strength = 0.8
blur_size = 8  # Standard deviation in pixels.

# Convert to float so that negatives don't cause problems
image = img_as_float(data.camera())
blurred = filter.gaussian_filter(image, blur_size)
highpass = image - unsharp_strength * blurred
sharp = image + highpass


fig, axes = plt.subplots(ncols=2)
axes[0].imshow(image, vmin=0, vmax=1)
axes[1].imshow(sharp, vmin=0, vmax=1)
```

Unsharp Masking (MATLAB)

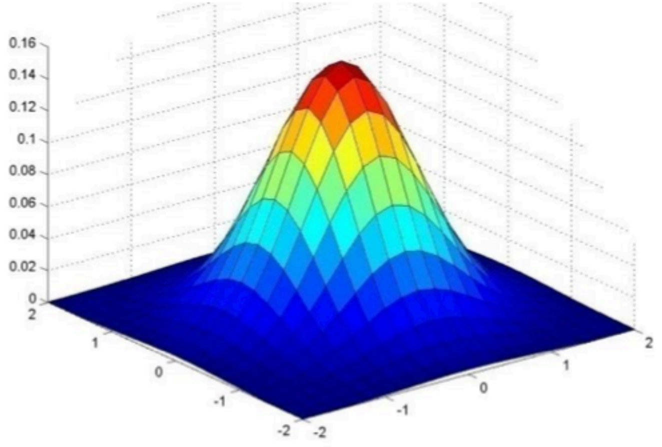
- a = imread(‘rice.png’)
- imshow(a)
- b = imsharpen(a,’Radius’,2, ‘Amount’,1);
- Imshow(b)

Your Convolution filter toolbox

- 90% of the filtering that you will do will be either
- Smoothing (or Blurring)
- High-Pass Filtering (will explain later)
- Most common filters:
- Smoothing: Gaussian
- High Pass Filtering: Derivative of Gaussian

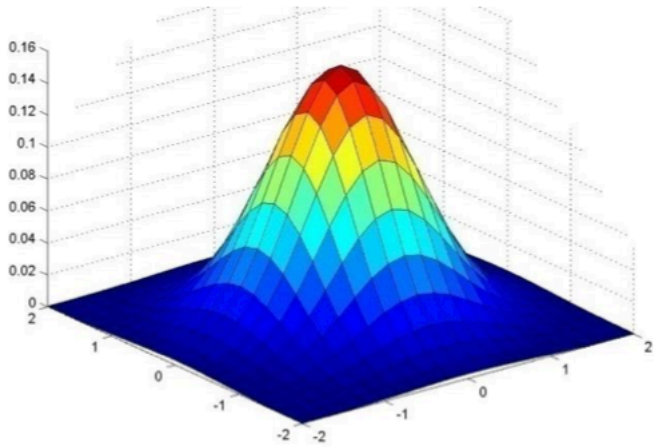
Gaussian Filter

- Gaussian = normal distribution function
-



Gaussian Filter

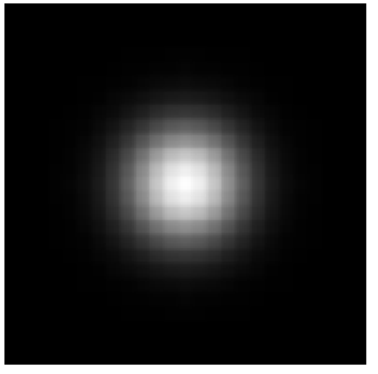
- Gaussian = normal distribution function
-

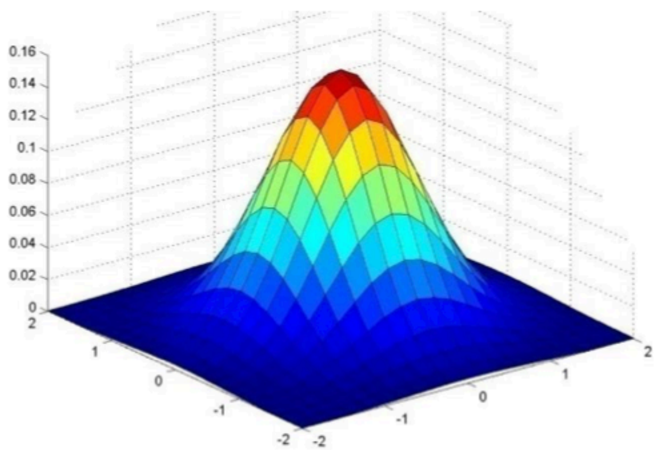


$$K(i,j) = \frac{1}{Z}\exp\left(-\frac{i^2+j^2}{2\sigma^2}\right)$$

Gaussian Filter

- Gaussian = normal distribution function
-

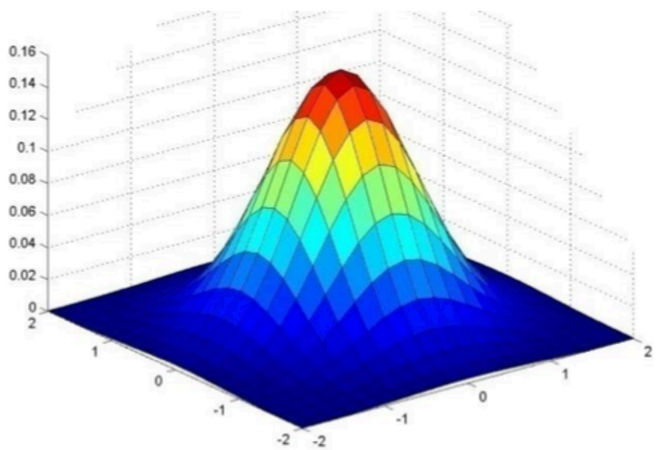




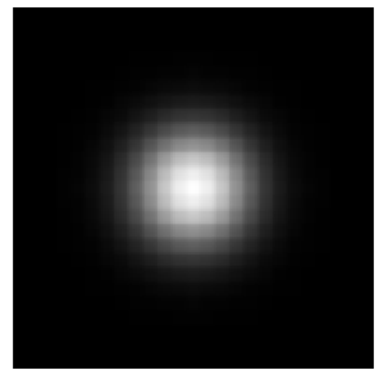
$$K(i,j) = \frac{1}{Z}\exp\left(-\frac{i^2+j^2}{2\sigma^2}\right)$$

Gaussian Filter

- Gaussian = normal distribution function
-



$$K(i,j) = \frac{1}{Z} \exp\left(-\frac{i^2+j^2}{2\sigma^2}\right)$$



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

Gaussian filter with different σ

Original

$\sigma = 3$

$\sigma = 5$

$\sigma = 7$

Gaussian Filter

- Remove “high-frequency” components from image (low-pass filter)
  - Images become more smooth
- Convolution with self is another Gaussian
  - So can smooth with small-width kernel, repeat and get same result as larger-width kernel would have.
  - Convolving two times with kernel width σ is same as convolving once with kernel width σ/2 (Can you proof this?)

Separability of the Gaussian filter

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

- The 2D Gaussian can be expressed as the product of two functions, one a function of x and the other the function of y.
- In this case, the two functions are (identical) 1D Gaussian.

Separability example

2D convolution
(center location only)

$$\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix} * \begin{bmatrix}2&3&3\\3&5&5\\4&4&6\end{bmatrix}$$

The filter factors
into a product of 1D
filters:

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

Perform convolution
along rows:

$$\begin{bmatrix}1&2&1\end{bmatrix} * \begin{bmatrix}2&3&3\\3&5&5\\4&4&6\end{bmatrix} = \begin{bmatrix}\ &11&\ \\ \ &18&\ \\ \ &18&\ \end{bmatrix}$$

Followed by convolution
along the remaining column:

Source: K. Grauman

Why is separability useful?

- The process of performing a convolution requires K^2 operations per pixel
- Suppose v and h are horizontal and vertical kernels.
- K = vh^T
- 2K operations per pixel!

Image Histogram Equalization

- We have a low-contrast image:

Image Histogram Equalization

- We would like to increase the contrast

What is a histogram

$$p_n = \frac{\text{number of pixels with intensity } n}{\text{total number of pixels}} \qquad n = 0, 1, ..., L-1.$$

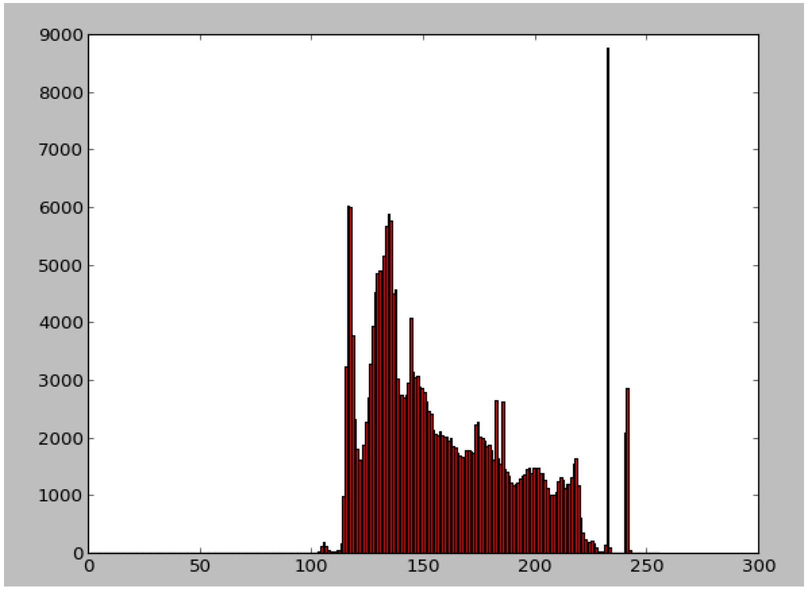
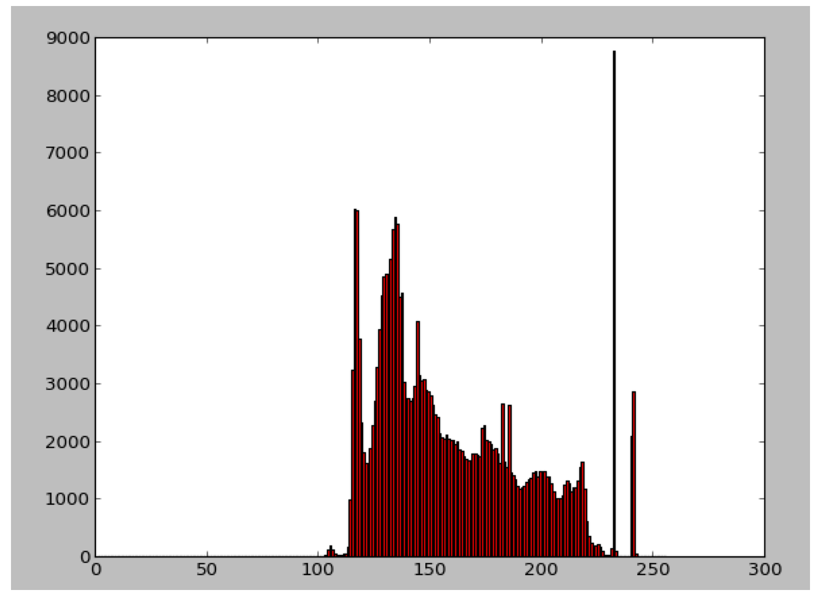


Image Histogram Equalization

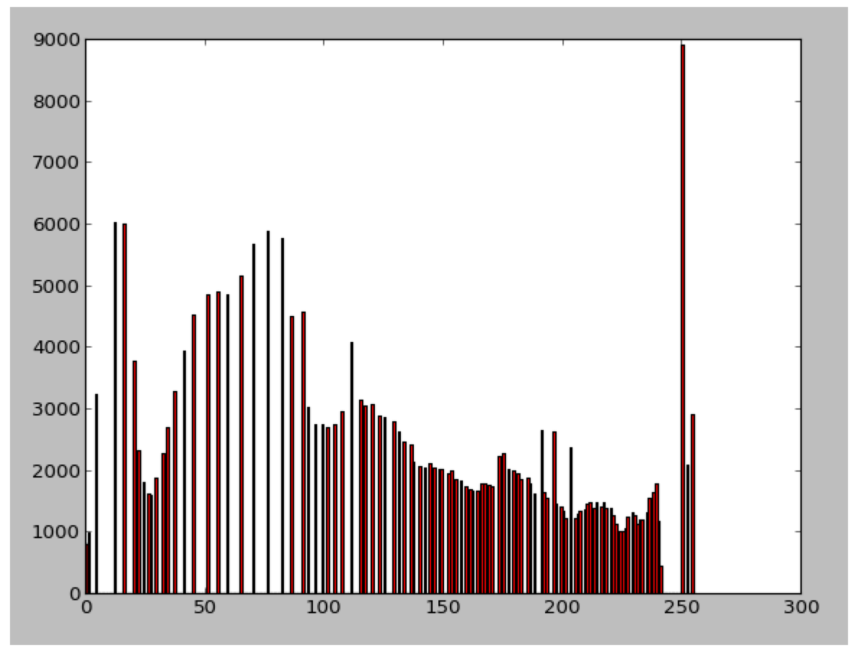
- We have a low-contrast image:



Limited image range

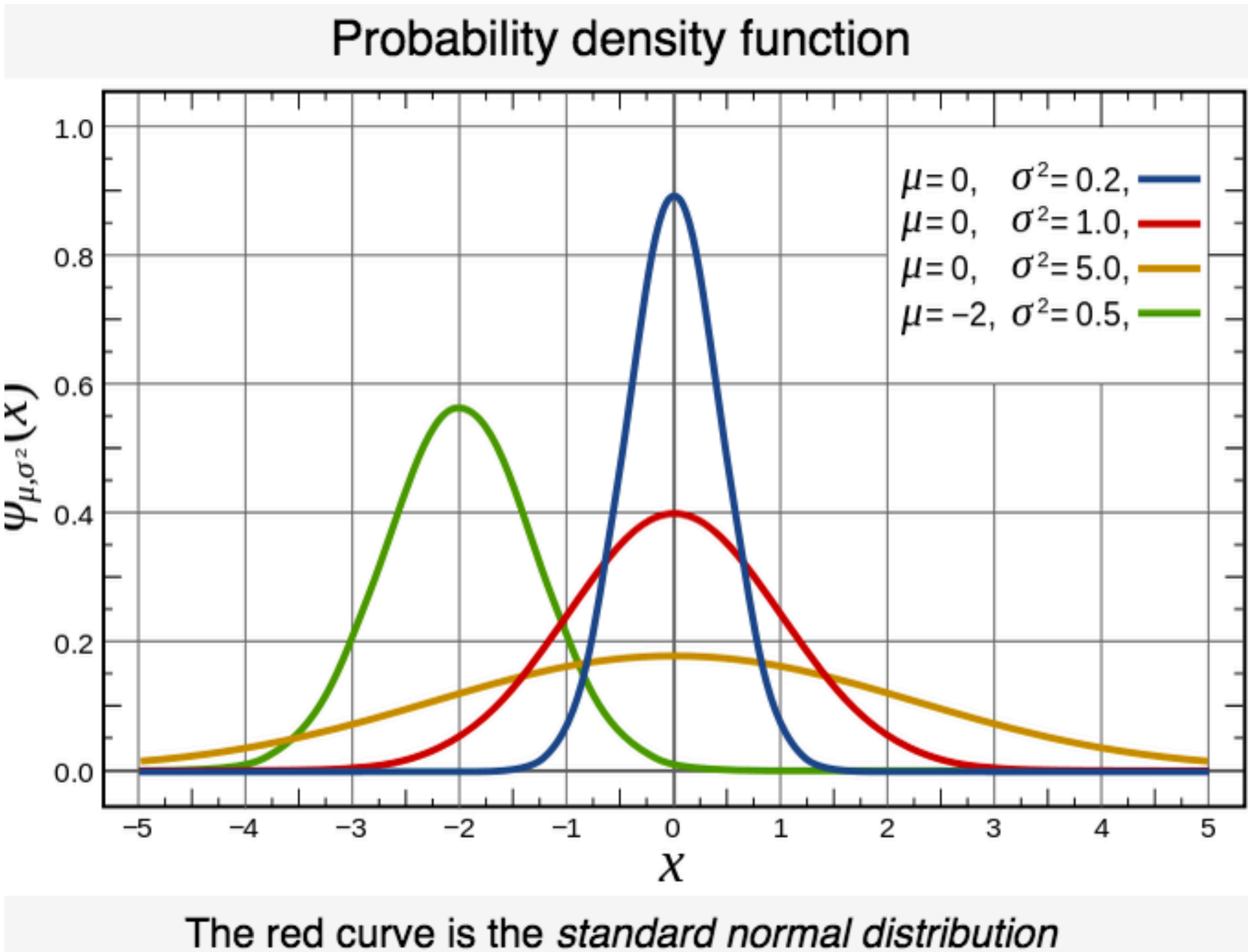
Image Histogram Equalization

- What we want is a histogram that covers the whole range of [0,255], but the shape must be preserved!



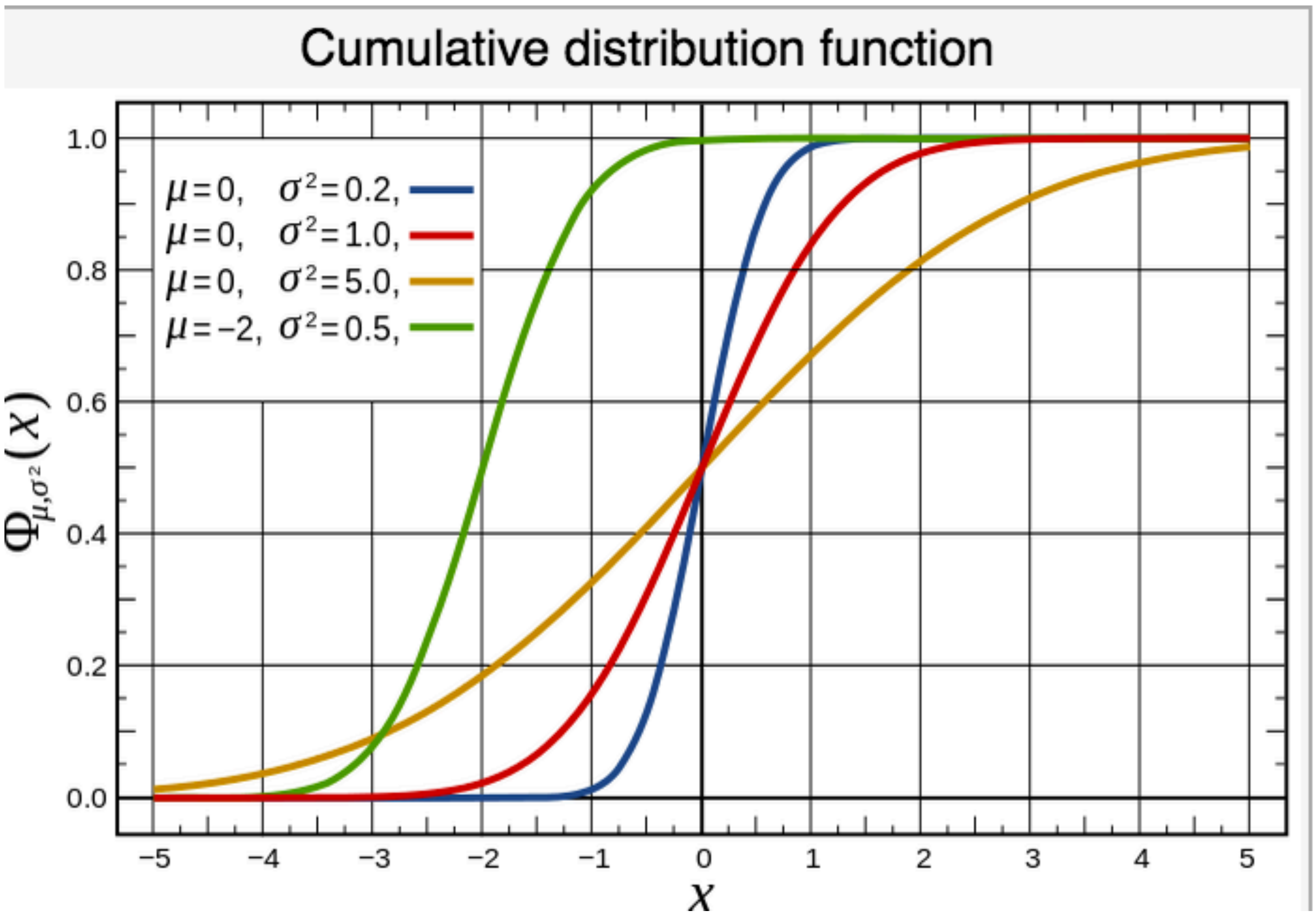
Ideal histogram

Mini detour of Probability Theory



PDF, density of a continuous random variable, is a function that describes the relative likelihood for this random variable to take on a given value.

Mini detour of Probability Theory

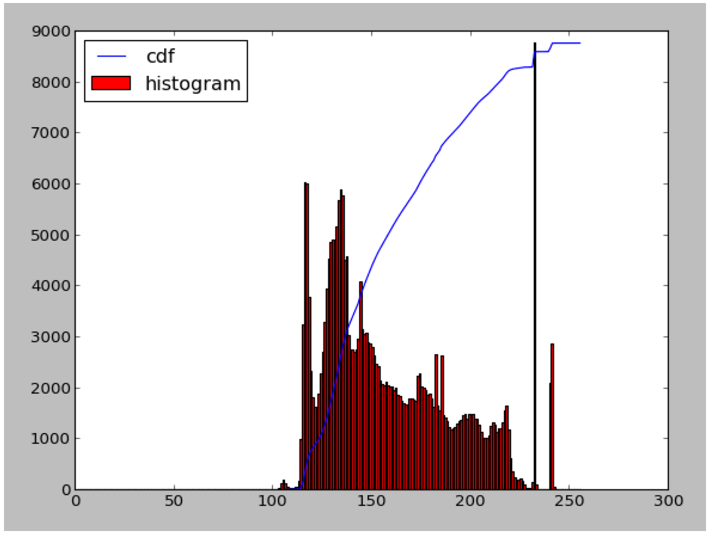


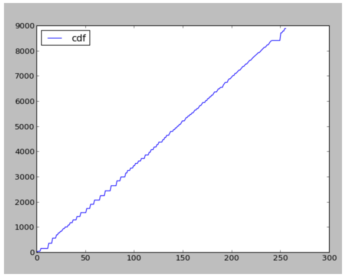
cumulative distribution function (CDF), describes the probability that a real-valued random variable X with a given probability distribution will be found to have a value less than or equal to x.

How does it work?

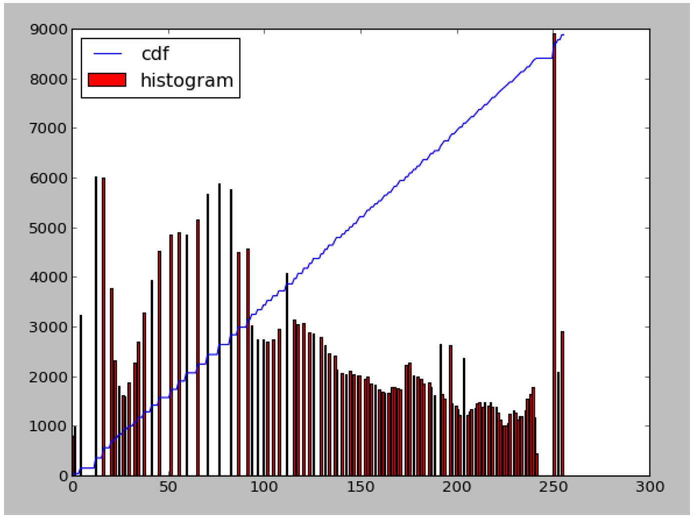
- Mapping one distribution to another distribution (a wider and more uniform of intensity values) so that the intensity values are spreading over the whole range
- The mapping should be the cumulative density function (CDF)

Stretching the CDF





Transform



Before

After

Next class

- Bordering effect
- Image Derivatives