BME I5000: Biomedical Imaging

Lecture 3
Intensity Manipulations

Lucas C. Parra, parra@ccny.cuny.edu

some slides inspired by lecture notes of Andreas H. Hilscher at Columbia University.

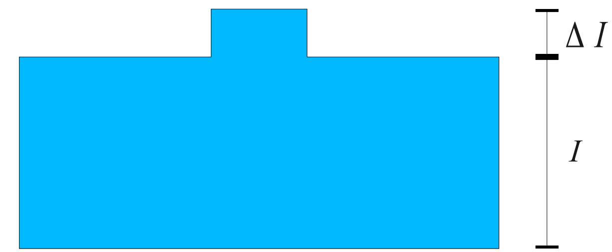
Schedule

1. Introduction, Spatial Resolution, Intensity Resolution, Noise
2. X-Ray Imaging, Mammography, Angiography, Fluoroscopy
3. Intensity manipulations: Contrast Enhancement, Histogram Equalization
4. Computed Tomography
5. Image Reconstruction, Radon Transform, Filtered Back Projection
6. Positron Emission Tomography
7. Maximum Likelihood Reconstruction
8. Magnetic Resonance Imaging
9. Fourier reconstruction, k-space, frequency and phase encoding
10. Optical imaging, Fluorescence, Microscopy, Confocal Imaging
11. Enhancement: Point Spread Function, Filtering, Sharpening, Wiener filter
12. Segmentation: Thresholding, Matched filter, Morphological operations
13. Pattern Recognition: Feature extraction, PCA, Wavelets
14. Pattern Recognition: Bayesian Inference, Linear classification

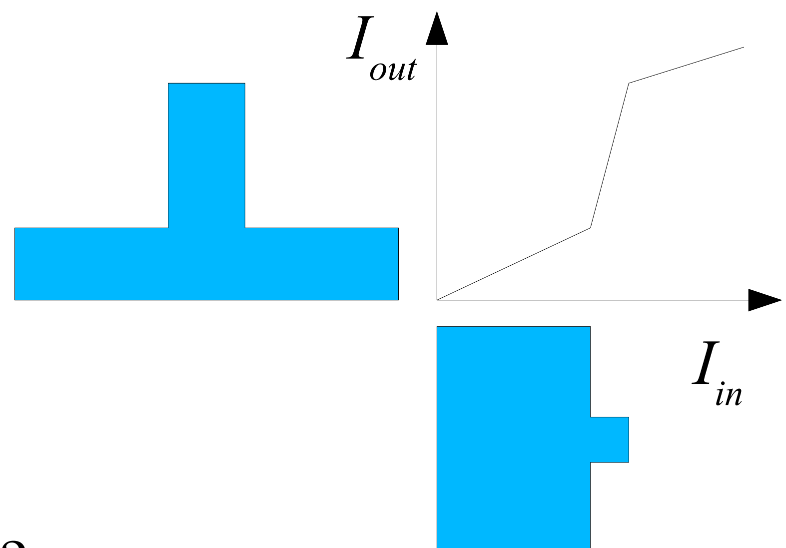
Contrast

Defined as:

$$C = \frac{\Delta I}{I}$$



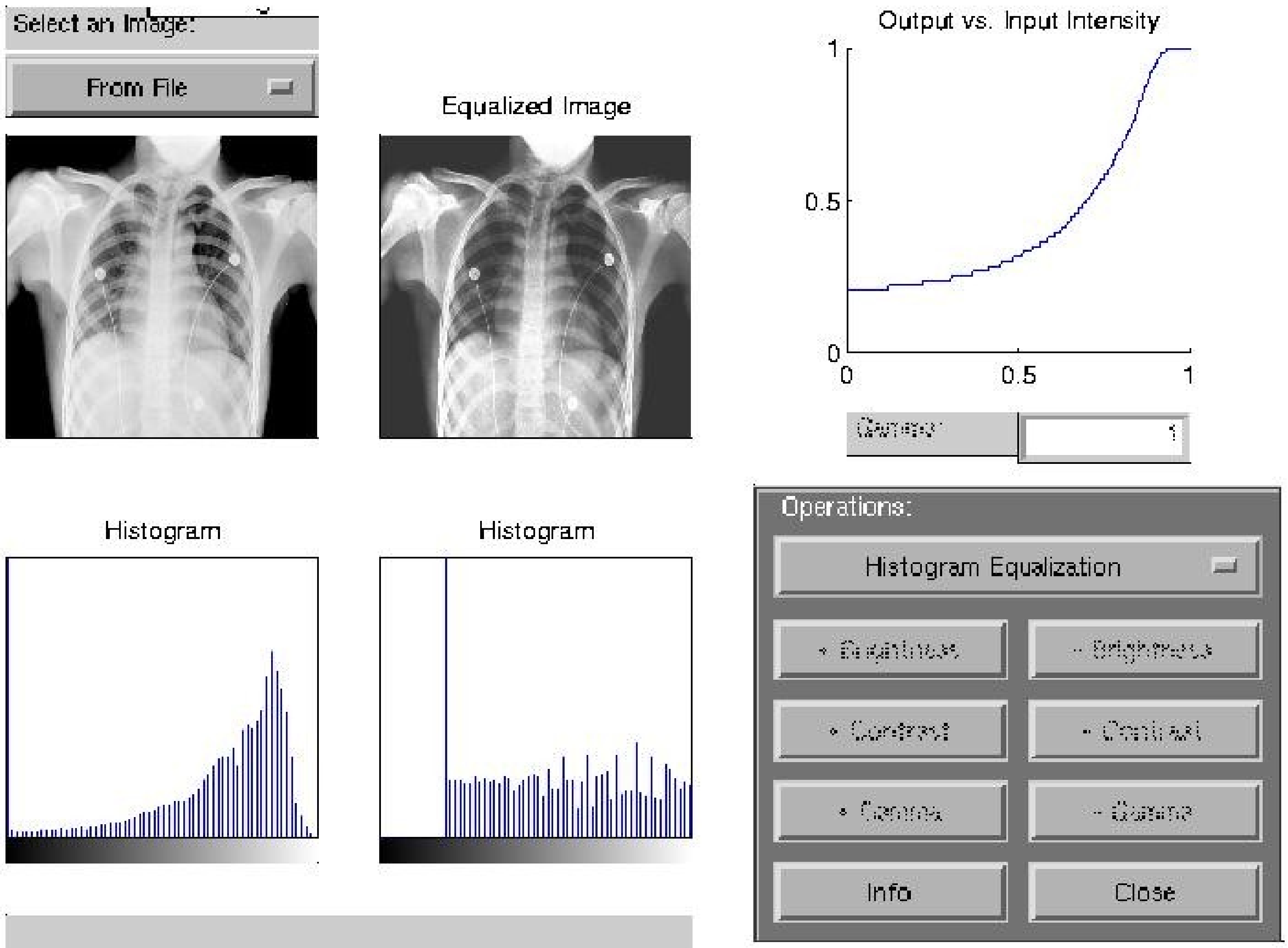
Enhancement by applying non-linearity to each pixel intensity $I_{out} = f(I_{in})$. Example here Contrast Stretching:



Question is:

- *Which* non-linearity?
- Should it be location dependent?

Intensity Resolution

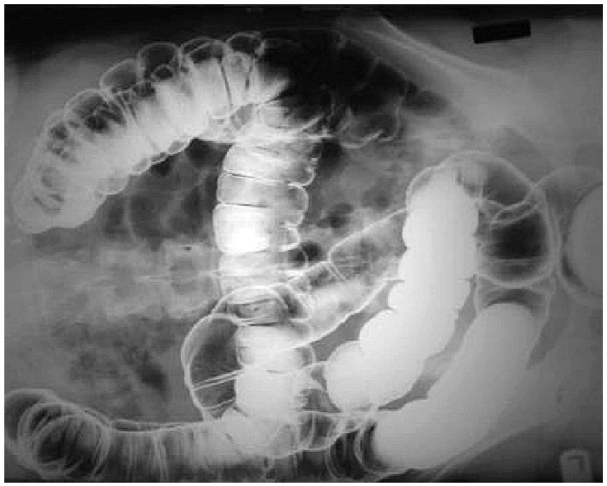



```
>> myimadjdemo
```

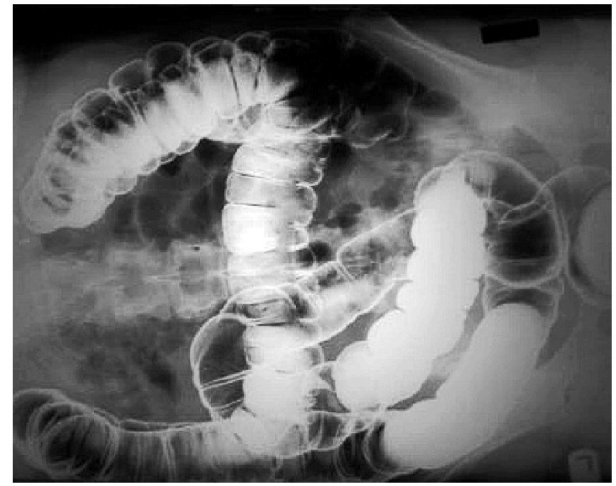
Contrast Stretching

Expand intensities that carry more information

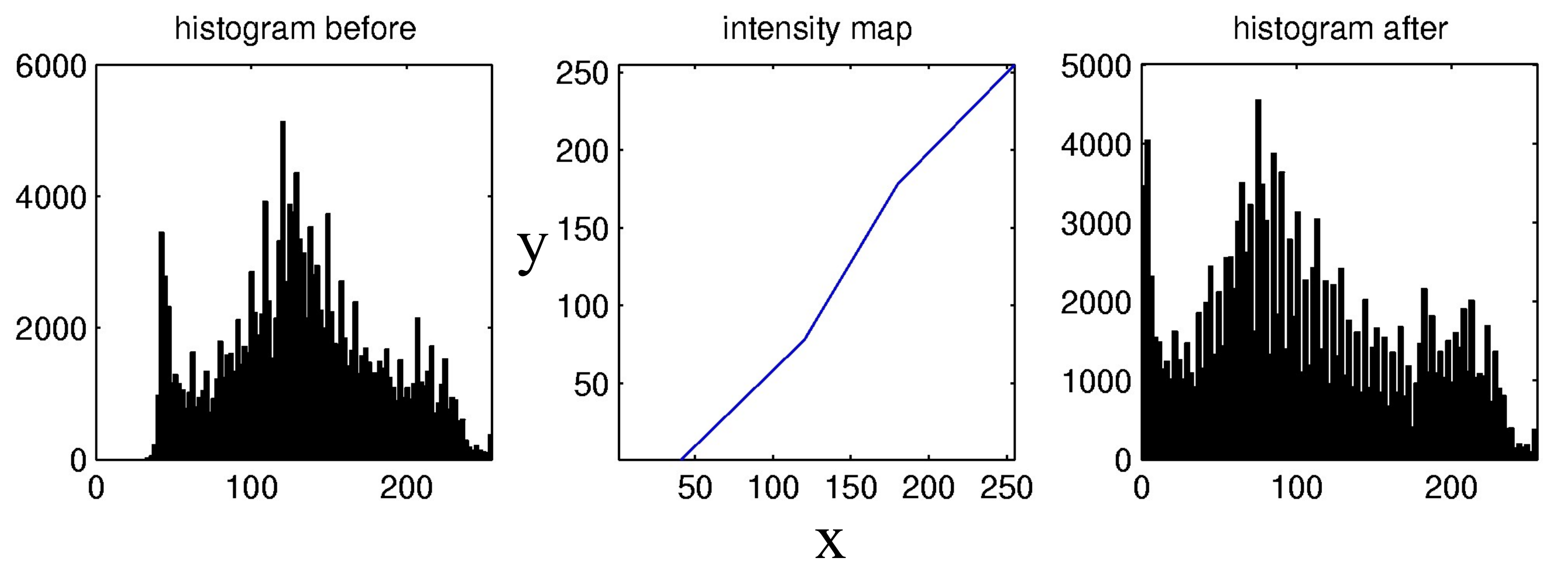
before



after

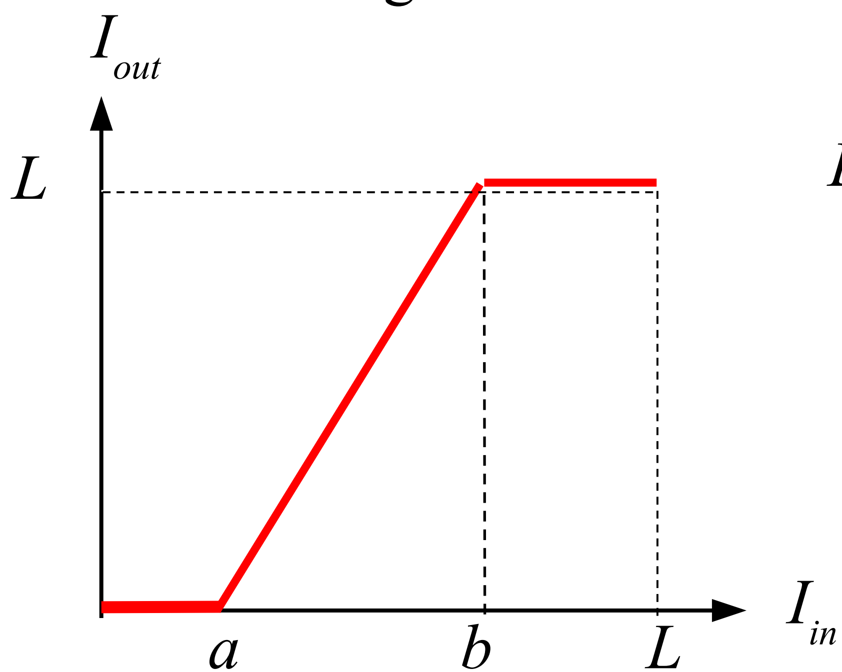


```
>> img = interp1(x,y,img);
```



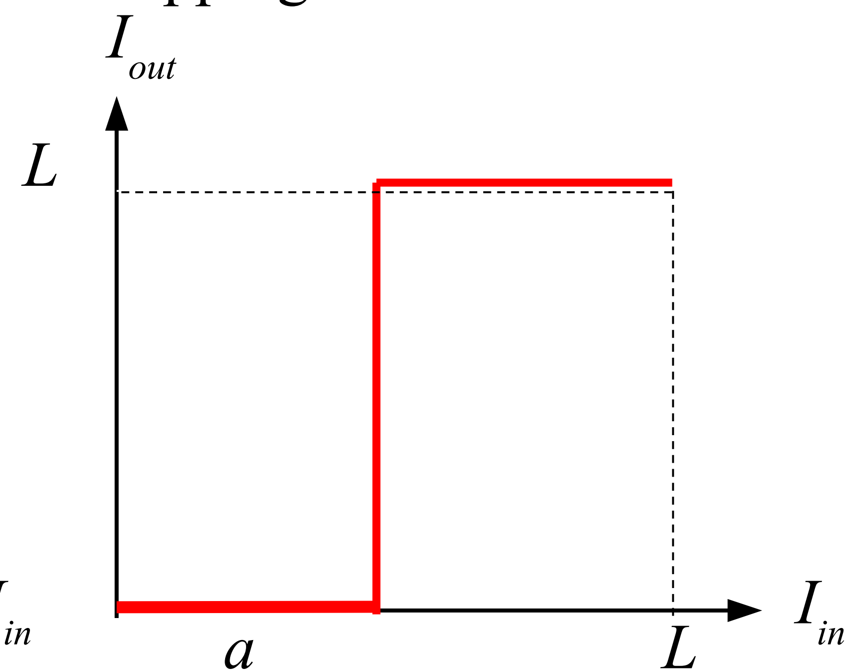
Clipping and Thresholding

Clipping is a special case of contrast stretching.



Useful for noise reduction when signal is known to lie in [a,b]. **However**, clipping in practice is often a problem as contrast is lost for high and low intensities that fall out of the dynamic range.

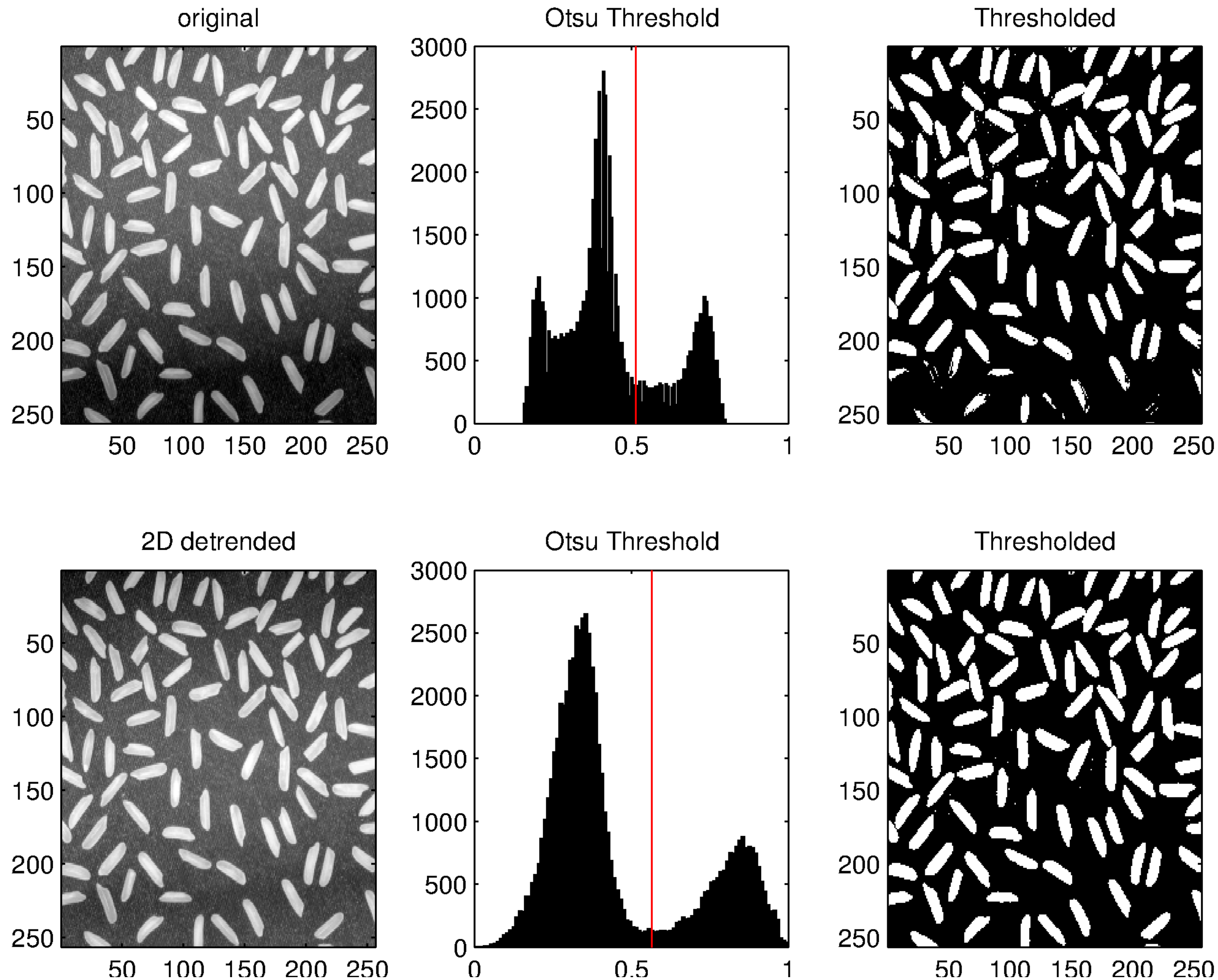
Thresholding is a special case of clipping a=b.



Useful to generate binary output.

Thresholding + detrending

- Otsu's method selects threshold to minimizes the intraclass variance.
- Thresholding does not work well with non-uniform lighting or background. Solution is to apply different thresholds in different areas, and interpolate between them, e.g. with 2D detrending:



Thresholding + detrending

Otsu's method in matlab:

```
>> img = img > graythresh(img);
```

2D detrending + Otsu's method:

```
>> img = img - myplanefit(img); % detrend
>> img = img > graythresh(img); % thresholding
```

Use this plane fitting to find the “2D trend”:

```
function Zest = myplanefit(Z)
% Zest = myplanefit(Z) will fit a plane such that values
% in matrix Z are approximated by Zest=mx*X+my*Y+c in a
% least squares sense with X,Y representing the column and
% row index of each element in the matrix.
myerror=@(Z,p) sum(sum((Z - myplane(p,size(Z))).^2));
poptimal = fminsearch(@(p) myerror(Z,p),[mean(Z(:)) 0 0]);
Zest = myplane(poptimal,size(Z));

function Zest = myplane(p,dims);
[X,Y] = meshgrid(1:dims(2),1:dims(1));
Zest = p(1)+p(2)*X+p(3)*Y;
```

It should be obvious how to extend this to fit a parabola if that should be useful for your case.

Thresholding + detrending in python

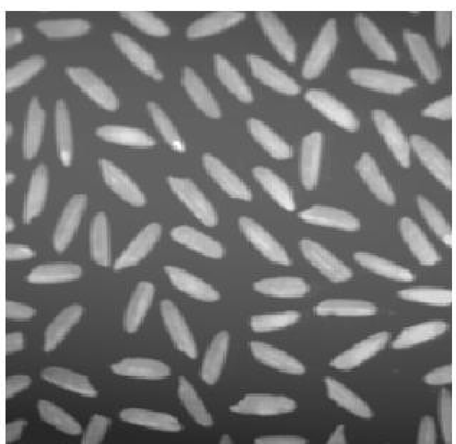
```
import skimage as ski
import numpy as np
from scipy.optimize import fmin

def myplane(p,dims):
   # Takes parameters (p) and image shape (dims) as input
   # Returns a plane of the same dimension transformed by parameters p
   [X,Y] = np.meshgrid(np.linspace(0,1, dims[0]), np.linspace(0,1, dims[1]))
   return p[0] + p[1]*X + p[2]*Y

def myplanefit(Z):
   # Uses the fmin function from scipy.optimizme to find the optimal parameters
   # Returns the optimal plane according to the defined error
   myerror = lambda p,Z: np.sum((Z-myplane(p,np.shape(Z)))**2)
   poptimal = fmin(myerror, [np.mean(Z), 0, 0], args=(Z, ))
   return myplane(poptimal,np.shape(Z))

## 2D detrending + Otsu
img = img - myplanefit(img)
img = img > ski.filters.threshold_otsu(img)
```

Clipping and Thresholding

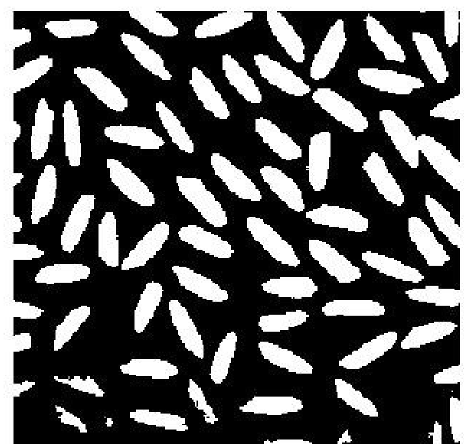
original

clipped

original

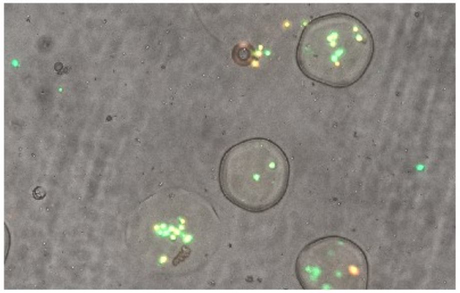


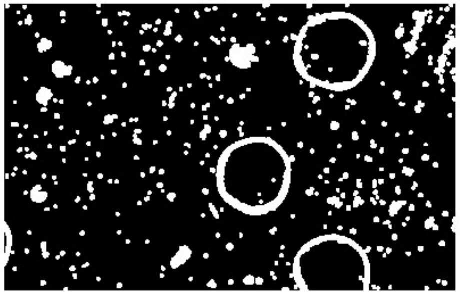
thresholded



Assignment 4: Measure the area and number of rice corns in slide 9. Be sure to subtract a quadratic model of the background intensity before thresholding. Use morphological operations to fill in “holes” and detach corrected rice corns.

Alternate 4: use image homodimer_staining.tiff and measure the area of cells (hotspots) inside of the “bubbles”, one bubble at a time. Bubbles are microgels beads . You will likely use thresholing followed by morphological operations and masking, as well as bwlabel() to select individual areas. I advise to use also funciton imfill() to fill in the bubbles, after detecting their edge with thresholding, and filtering for size. Here how some of the intermediate images might look like …





Morphological operations

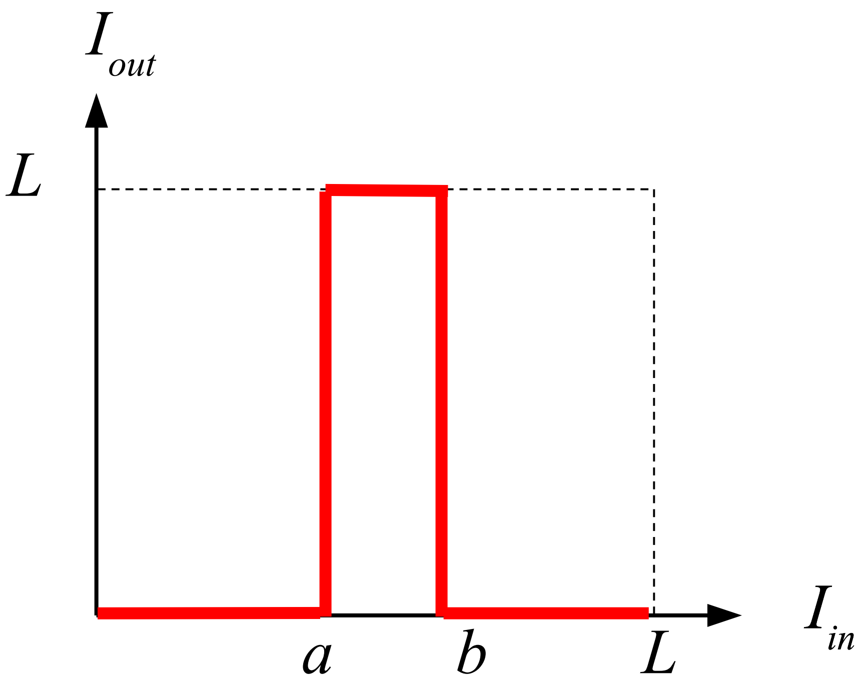
- Erosion
- Dilation
- Opening
- Closing

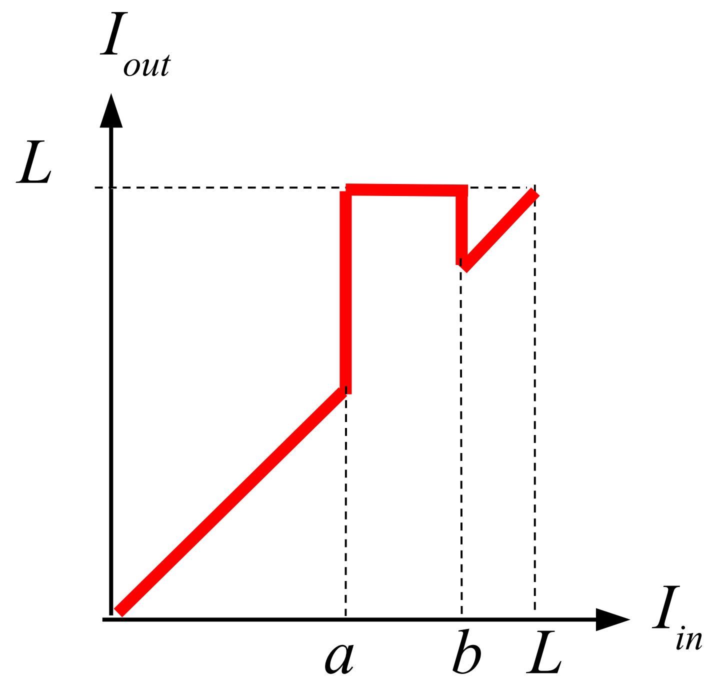
Intensity level slicing

Segmentation of certain gray level regions:

Without background

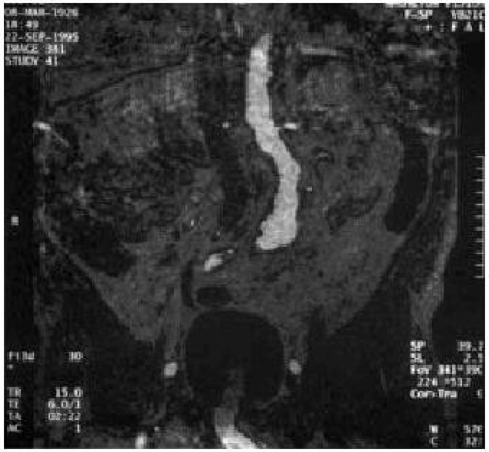
With background



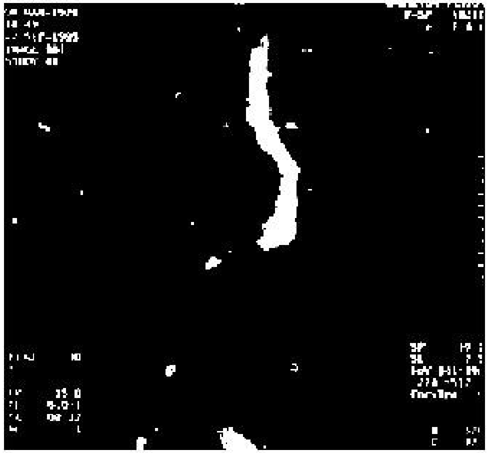


Intensity level slicing

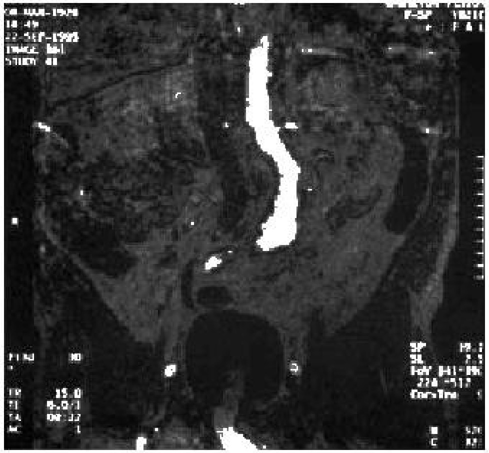
original figure



slicing without background



slicing with background



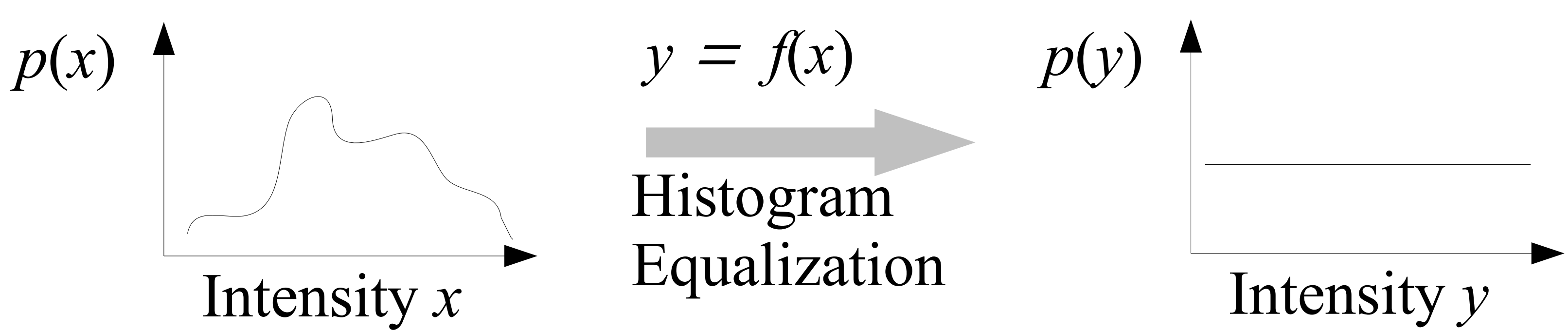
Histogram Equalization

A standard heuristic method for choosing the contrast enhancing non-linearity is to make the histogram of the intensities uniform.

This will:

- Stretch apart intensity values that are too close together.
- Represent all intensities equally often.
- Spend more of the dynamic range on intensities that occur more often.

Assumption: All intensity values carry the same amount of information.

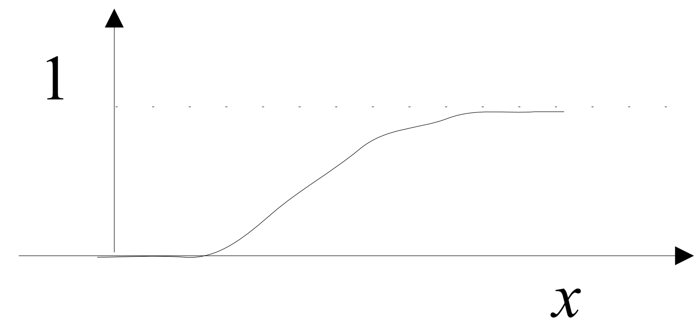


Random Variables - Probability density
(review)

Probability distribution defined as the probability that $X \leq x$

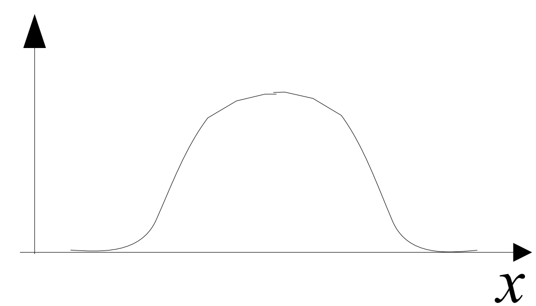
$$F_X(x) = Pr(X \leqslant x)$$

$$F_X(-\infty) = 0, \qquad F_X(\infty) = 1$$



Probability density function (**PDF**)

$$p_X(x) = \frac{\partial}{\partial x} F_X(x)$$



Alternatively, a **PDF** is a real valued function with

$$\int_{-\infty}^{\infty} dx\, p_X(x) = 1, \qquad p_X(x) \geqslant 0$$

Random Variables - Histogram

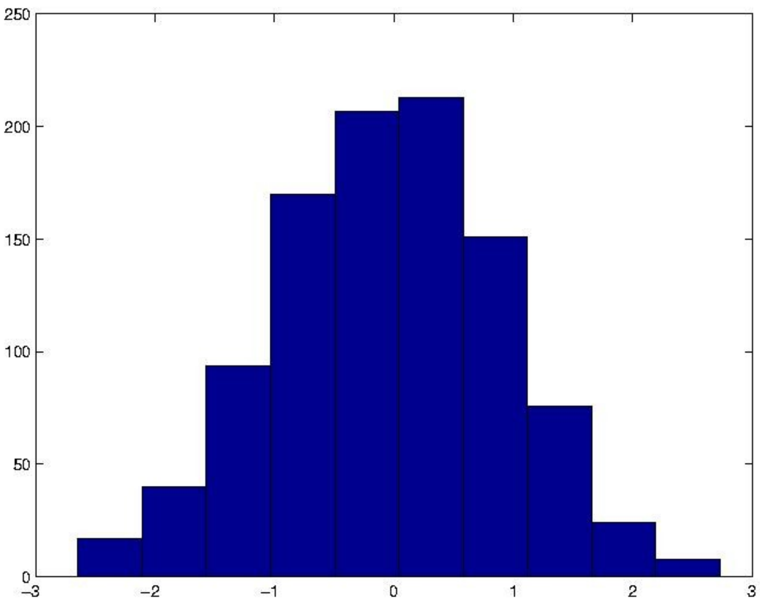
(review)

Estimate of probability density is the **histogram**

$$p_X(x) \approx \frac{F_X(x+\Delta x) - F_X(x)}{\Delta x} \propto Pr(x \leqslant X \leqslant x+\Delta x)$$

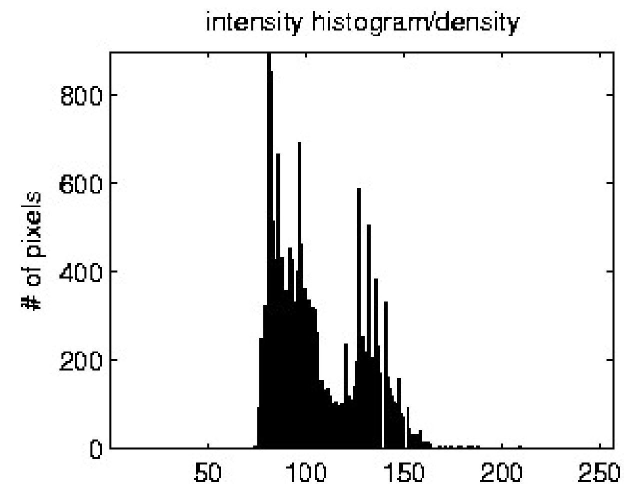
Where we measure the likelihood $Pr(x \leqslant X \leqslant x+\Delta x)$
by counting how many samples fall within x and $x + \Delta x$.

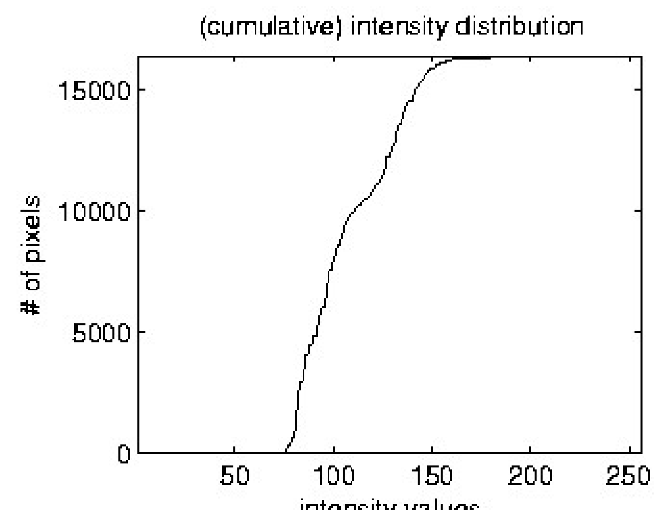
```
>> hist(randn(1000,1))
```



Histogram of an image

```
>> z = 1:256;
>> h=hist(img(:),z)
>> bar(z,h)
>> plot(z,cumsum(h))
```





Histogram Equalization

Goal: Given intensities x with pdf $p(x)$, find an intensity map $y = f(x)$, such that $p(y) = const.$

$$p(x) = \left|\frac{dy}{dx}\right| p(y) = \left|\frac{df(x)}{dx}\right| const.$$

$$\int_{-\infty}^{x} dx'\, p(x') = \int_{f(-\infty)}^{f(x)} df = f(x)$$

Result: The non-linear intensity transformation in histogram equalization is the cumulative histogram of the data.

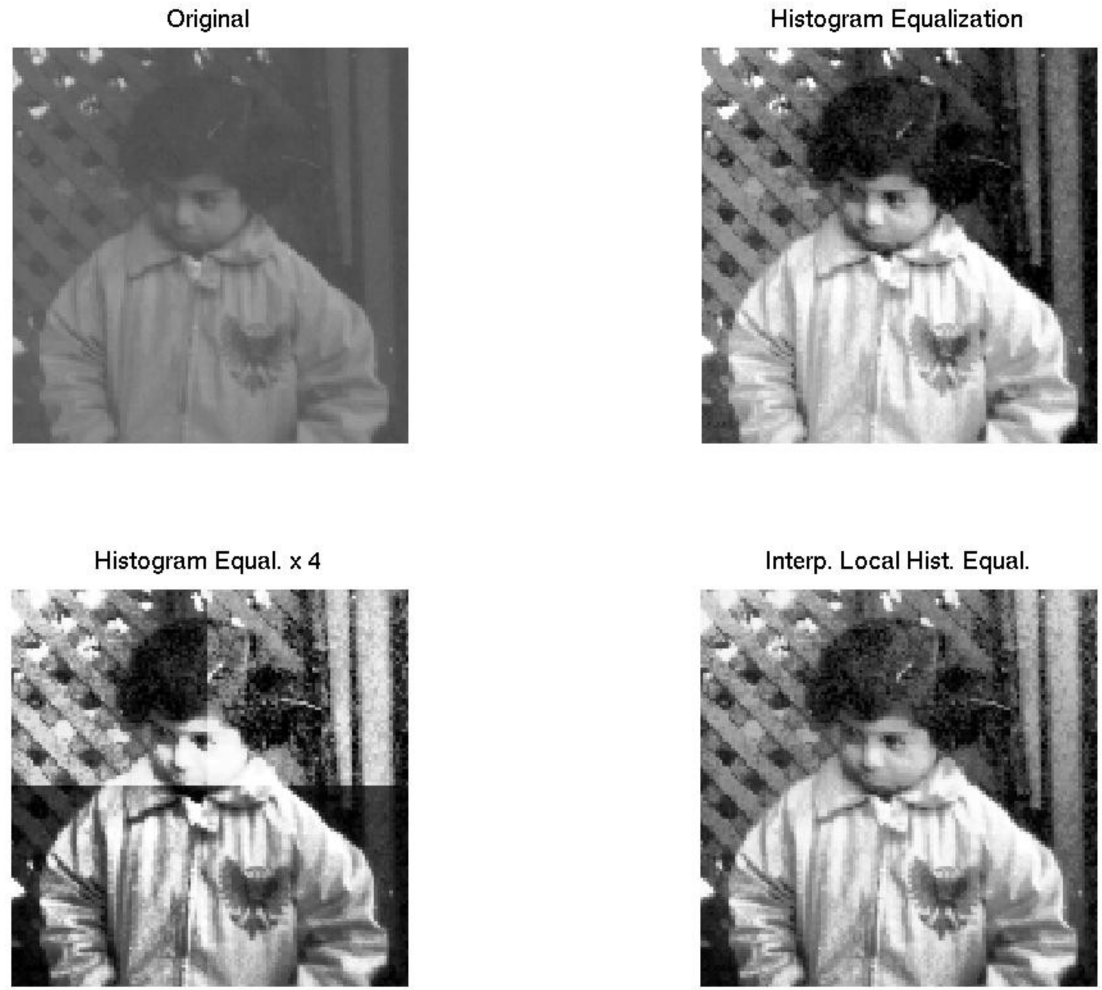
$$y = f(x) = \int_{-\infty}^{x} dx'\, p(x') \equiv F(x)$$

Implementation on image `I`:

```
>> N=prod(size(I)); z=0:255;
>> Ie = interp1(z,cumsum(hist(I(:),z))/N,I);
```

Histogram Equalization - Local

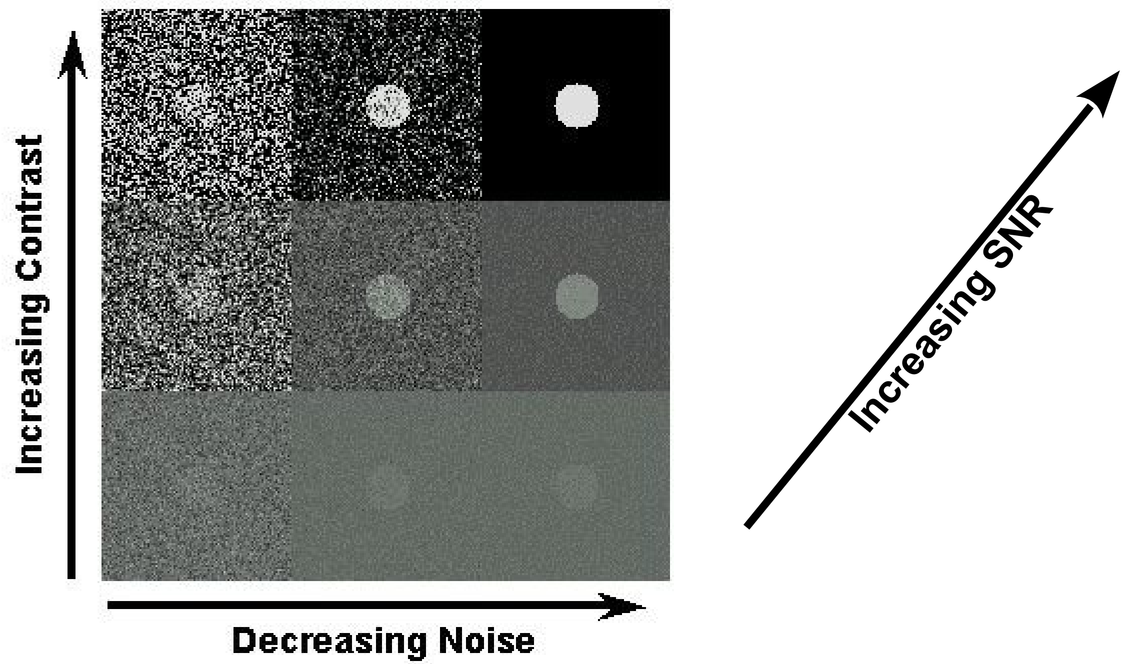
- Optimal intensity correction may not be the same everywhere in the image.
- Therefore one may need to do separate hist. equal. in different regions of the image.
- To avoid artificial contrast edges due to the different $F(x)$ these have to be interpolated based on the position of the pixel relative to the surrounding patches.



Contrast vs. Noise

Increasing contrast may also increase noise intensities.

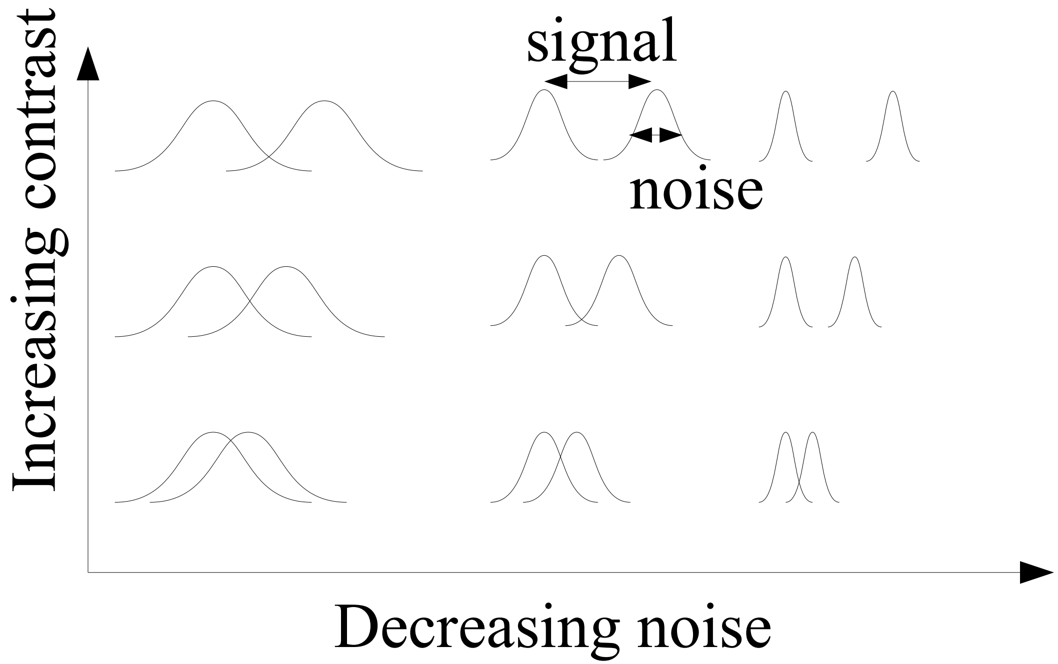
For the detection of small structures the important criteria is SNR and not contrast:



Note the importance of defining what is signal and what is noise.

Contrast vs. Noise

Detectability of intensity variation depends on the SNR:



$$d' = \frac{\Delta\mu}{\sigma}$$

The relevant value for detection is the SNR. Here it is the ratio between the intensity difference $\Delta\mu$ and the noise level σ.

This metric is particularly important in X-ray imaging and is sometimes used to quantify the quality of a system.