Computer Vision

CSE/ECE 344/544

Summary of 2D transformations

Translation: $P' = P + T$

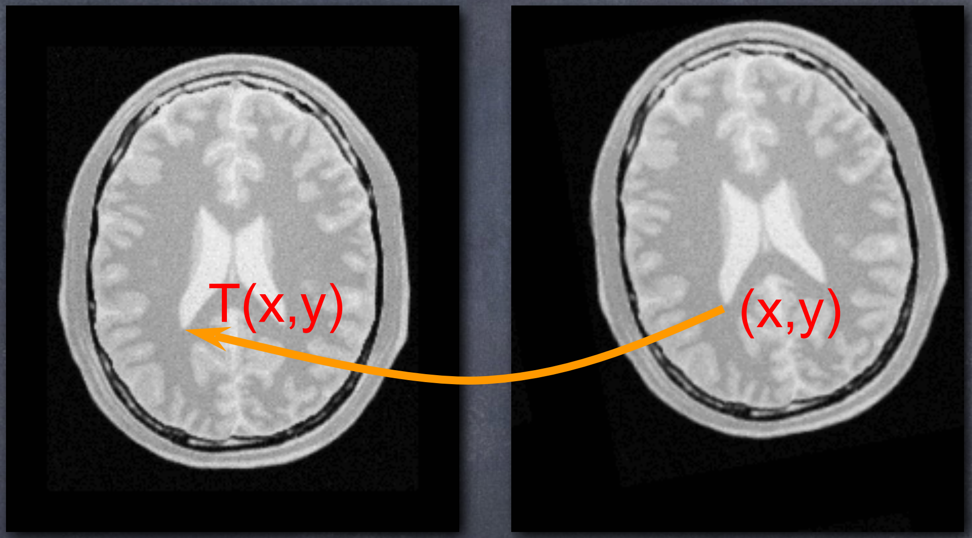
Scale: $P' = S\ P$

Rotation: $P' = R\ P$

Affine Transform

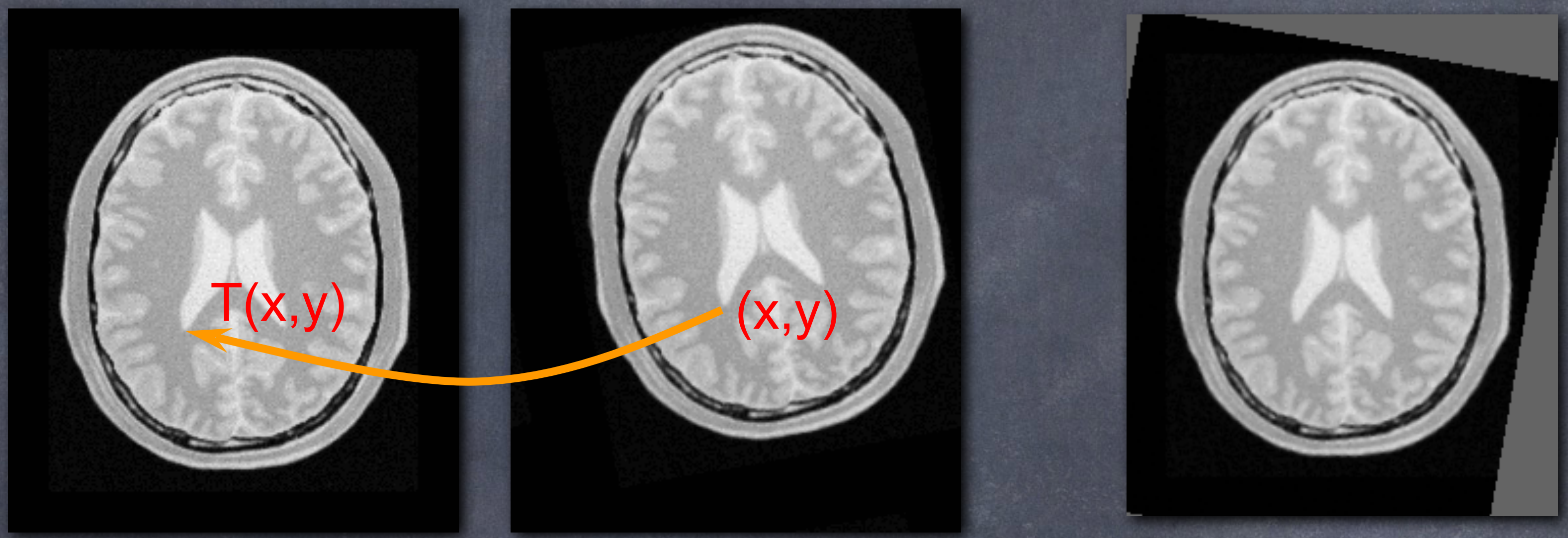
- What is affine transformation?

Let us taken an example



Fixed Image Moving Image

Let us taken an example



$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = A \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + B$$

0th order coefficients

1st order coefficients

Revisiting R-T-S

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = A \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + B$$

What happens if:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

Revisiting R-T-S

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = A \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + B$$

What happens if:

$$A = \begin{bmatrix} a_{11} & 0 \\ 0 & a_{22} \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$A = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Put it all together

- Translation: $\begin{vmatrix} x' \\ y' \end{vmatrix} = \begin{vmatrix} x \\ y \end{vmatrix} + \begin{vmatrix} t_x \\ t_y \end{vmatrix}$

- Rotation: $\begin{vmatrix} x' \\ y' \end{vmatrix} = \begin{vmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{vmatrix} \cdot \begin{vmatrix} t_x \\ t_y \end{vmatrix}$

- Scaling: $\begin{vmatrix} x' \\ y' \end{vmatrix} = \begin{vmatrix} s_x & 0 \\ 0 & s_y \end{vmatrix} \cdot \begin{vmatrix} t_x \\ t_y \end{vmatrix}$

Or, 3x3 Matrix Representations

- Translation:

$$\begin{vmatrix} x' \\ y' \\ 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & tx \\ 0 & 1 & ty \\ 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

- Rotation:

$$\begin{vmatrix} x' \\ y' \\ 1 \end{vmatrix} = \begin{vmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

$$\begin{vmatrix} x' \\ y' \\ 1 \end{vmatrix} = \begin{vmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

- Scaling:

Affine Transformation

- Translation, Scaling, Rotation, Shearing are all affine transformation
- Affine transformation – transformed point (x',y') is a linear combination of the original point (x,y), i.e.

$$\begin{vmatrix} x' \\ y' \\ 1 \end{vmatrix} = \begin{vmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ 0 & 0 & 1 \end{vmatrix} \cdot \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

- Any 2D affine transformation can be decomposed into a rotation, followed by a scaling, followed by a shearing, and followed by a translation.

  Affine matrix = translation x shearing x scaling x rotation

Composing Transformation

- Composing Transformation - the process of applying several transformation in succession to form one overall transformation
- If we apply transforming a point P using M1 matrix first, and then transforming using M2, and then M3, then we have:

  (M3 x (M2 x (M1 x P)))

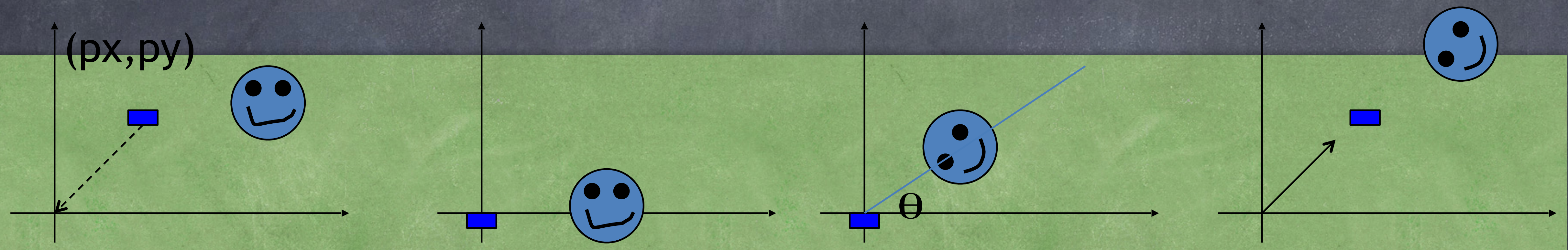
Composing Transformation

- Composing Transformation - the process of applying several transformation in succession to form one overall transformation
- If we apply transforming a point P using M1 matrix first, and then transforming using M2, and then M3, then we have:

$$(M3 \times (M2 \times (M1 \times P))) = \underbrace{M3 \times M2 \times M1}_{M} \times P$$

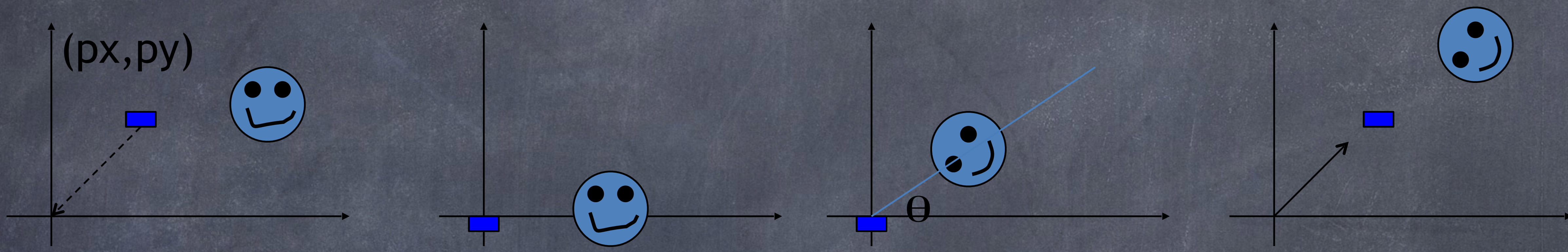
(pre-multiply)

Arbitrary Rotation Center



$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & px \\ 0 & 1 & py \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -px \\ 0 & 1 & -py \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Arbitrary Rotation Center



$$\begin{vmatrix} x' \\ y' \\ 1 \end{vmatrix} = \underbrace{\begin{vmatrix} 1 & 0 & px \\ 0 & 1 & py \\ 0 & 0 & 1 \end{vmatrix}}_{M_3} \underbrace{\begin{vmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{vmatrix}}_{M_2} \underbrace{\begin{vmatrix} 1 & 0 & -px \\ 0 & 1 & -py \\ 0 & 0 & 1 \end{vmatrix}}_{M_1} \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

$$M = M_3 * M_2 * M_1$$

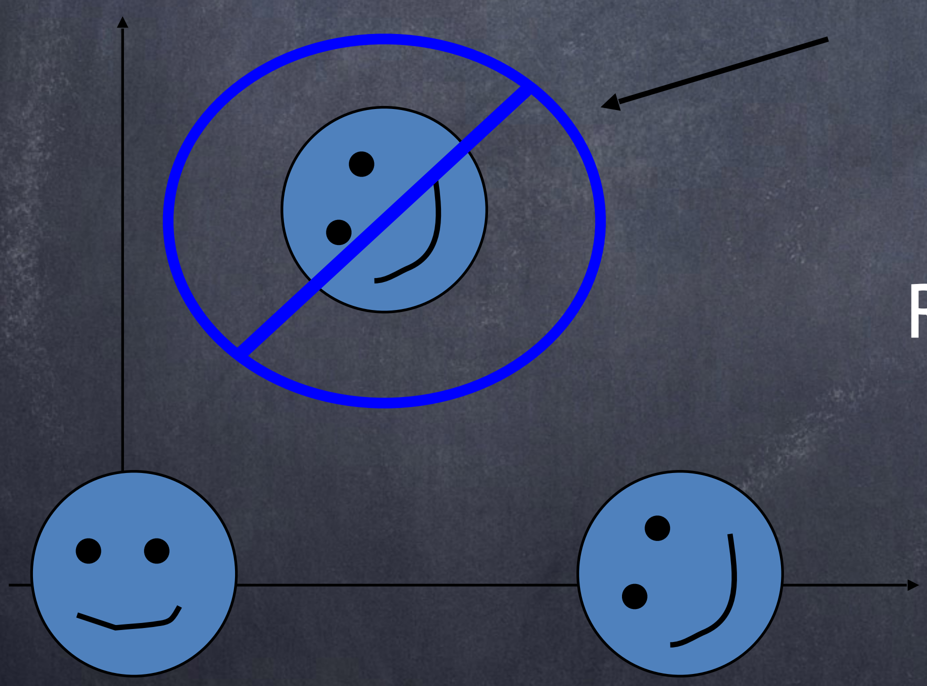
Composing Transformation

- Matrix multiplication is associative

  $M3 \times M2 \times M1 = (M3 \times M2) \times M1 = M3 \times (M2 \times M1)$
- Transformation products may not be commutative $A \times B \ != B \times A$

Transformation Order Matters!

- Example: rotation and translation are not commutative



Translate (5,0) and then Rotate 60 degree

OR

Rotate 60 degree and then translate (5,0)??

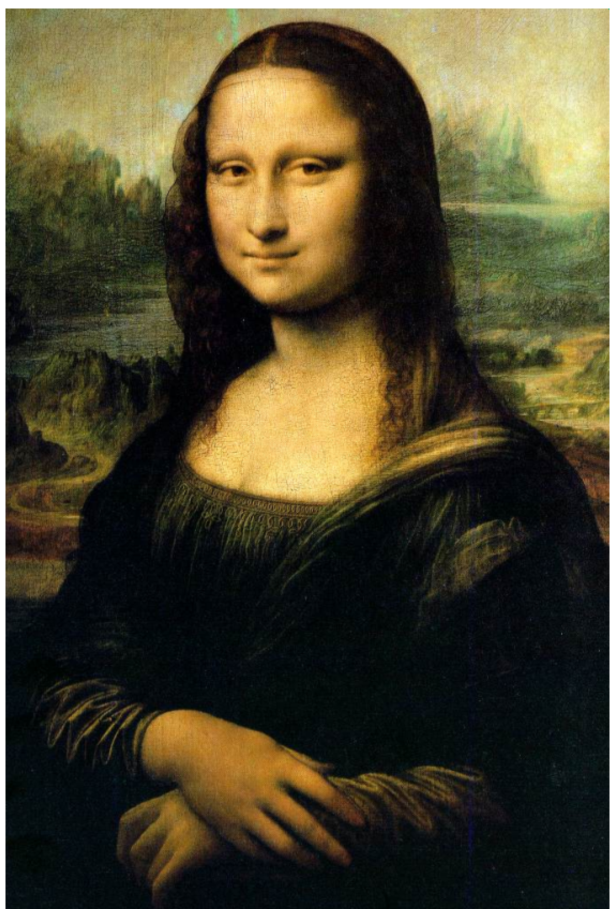
Rotate and then translate !!

Finding Affine Transformations

- How many points determines affine transformation

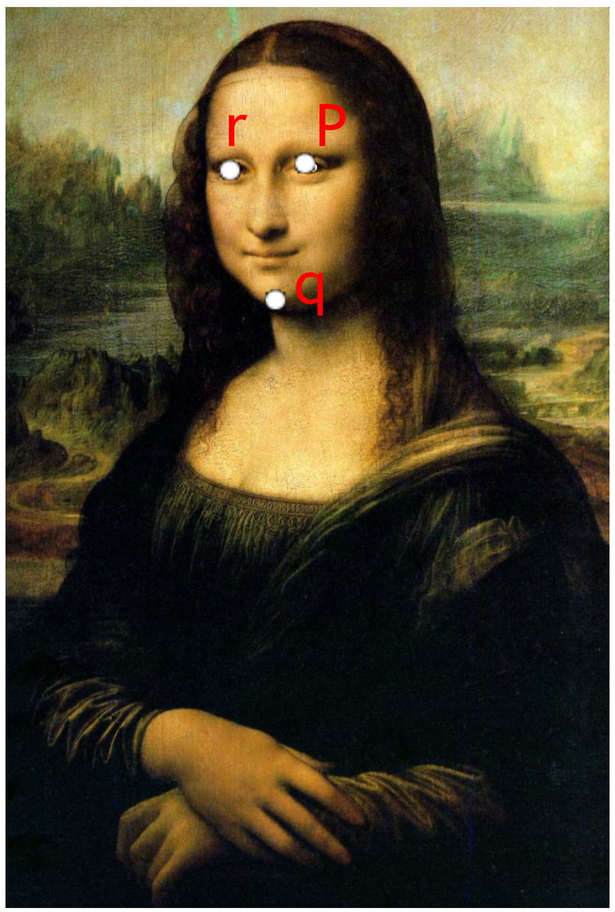
Finding Affine Transformations

- How many points determines affine transformation



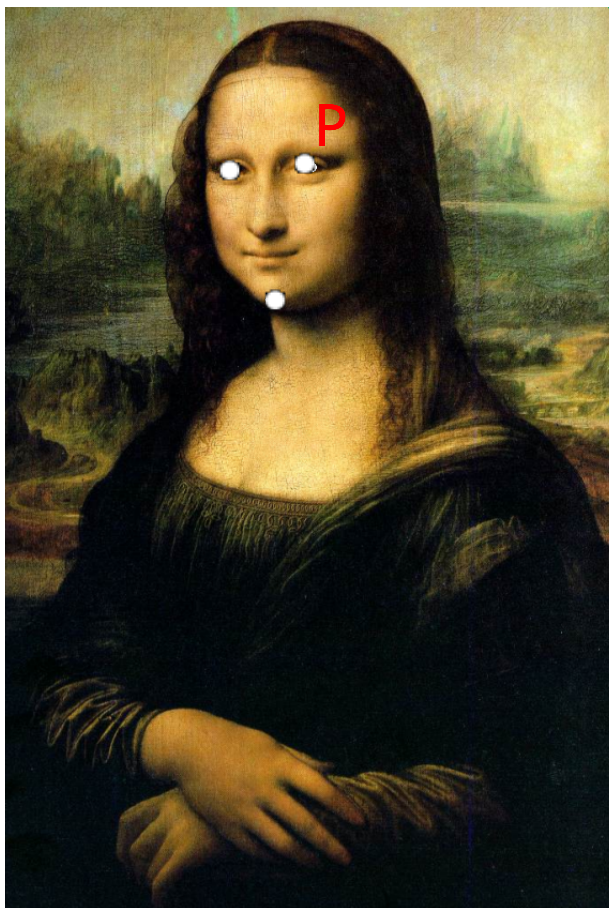
Finding Affine Transformations

- Image of 3 points determines affine transformation



Finding Affine Transformations

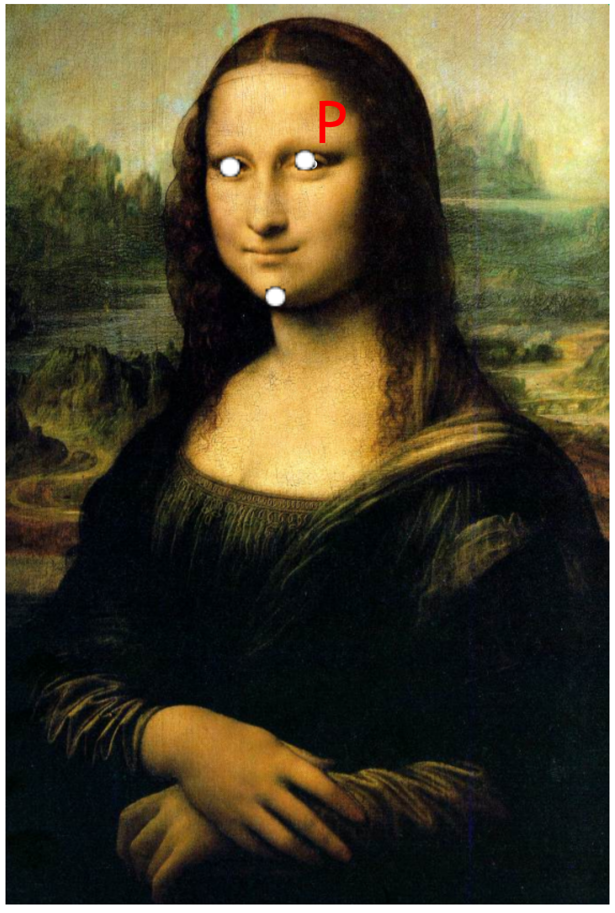
- Image of 3 points determines affine transformation



$$\begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix} = \begin{pmatrix} p'_x \\ p'_y \\ 1 \end{pmatrix}$$

Finding Affine Transformations

- Image of 3 points determines affine transformation

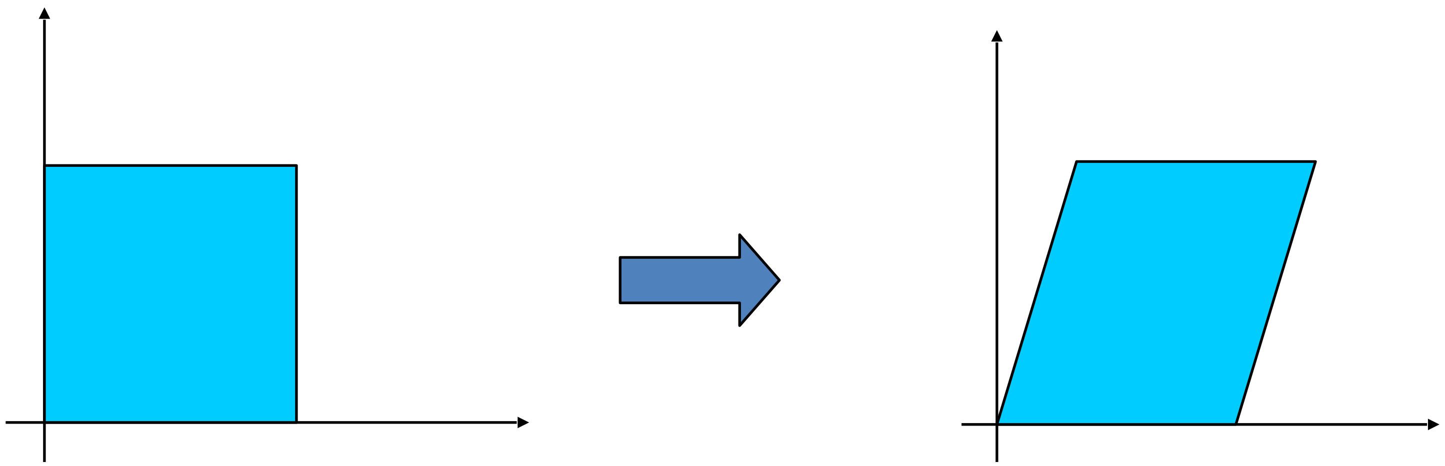


- Each pair gives us 2 linear equations on 6 unknowns!

- In total, 6 unknowns 6 linear equations.

$$\begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ 1 \end{pmatrix} = \begin{pmatrix} p'_x \\ p'_y \\ 1 \end{pmatrix}$$

Shearing

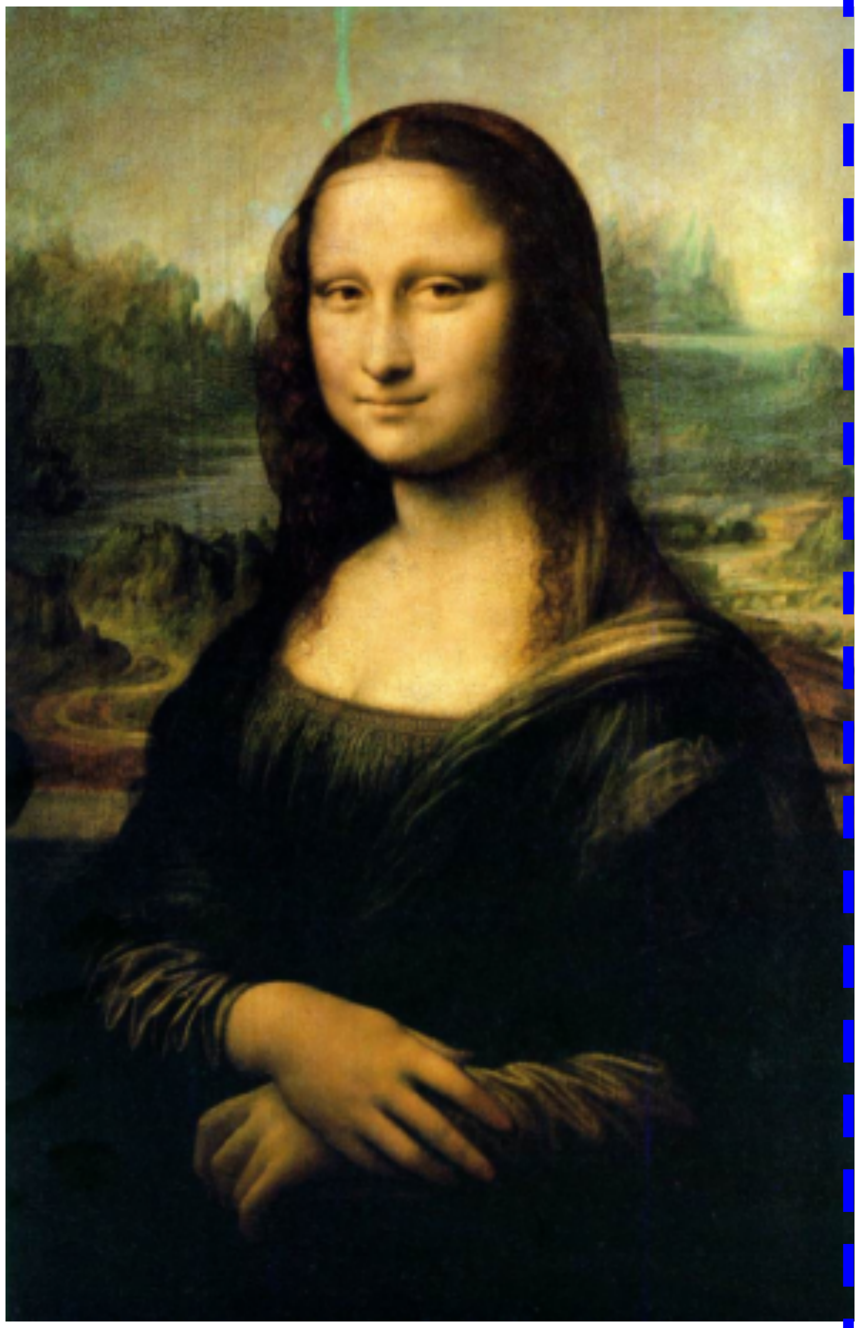


- Y coordinates are unaffected, but x coordinates are translated linearly with y
- That is:
  - y' = y
  - x' = x + y * h

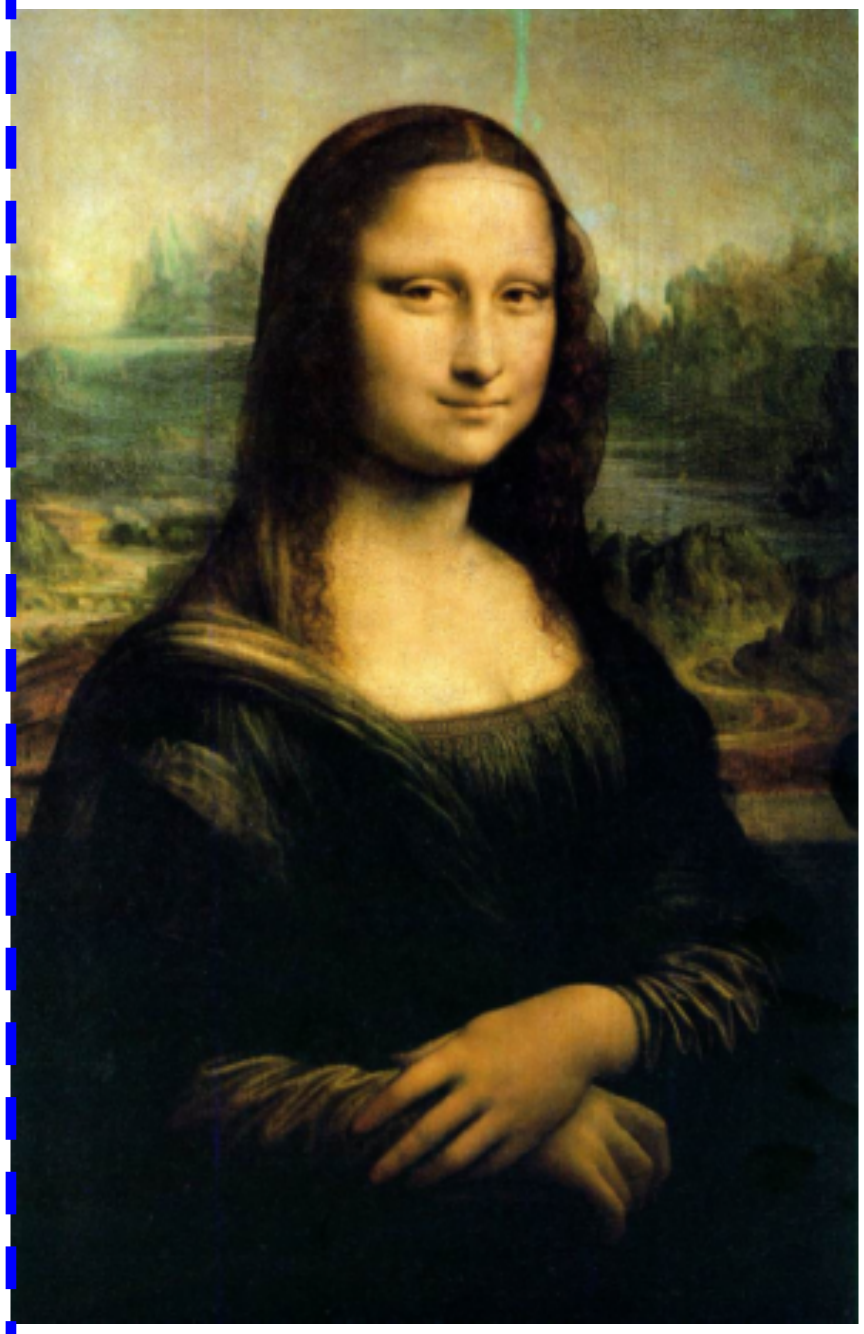
$$\begin{vmatrix} x \\ y \\ 1 \end{vmatrix} = \begin{vmatrix} 1 & h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

Reflection

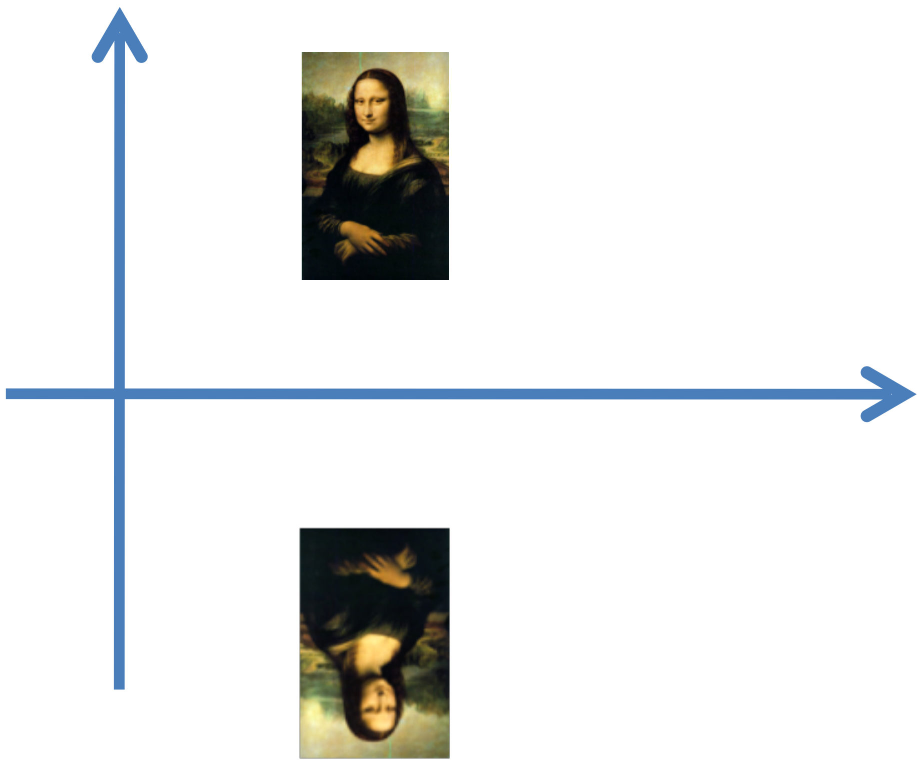
Reflection



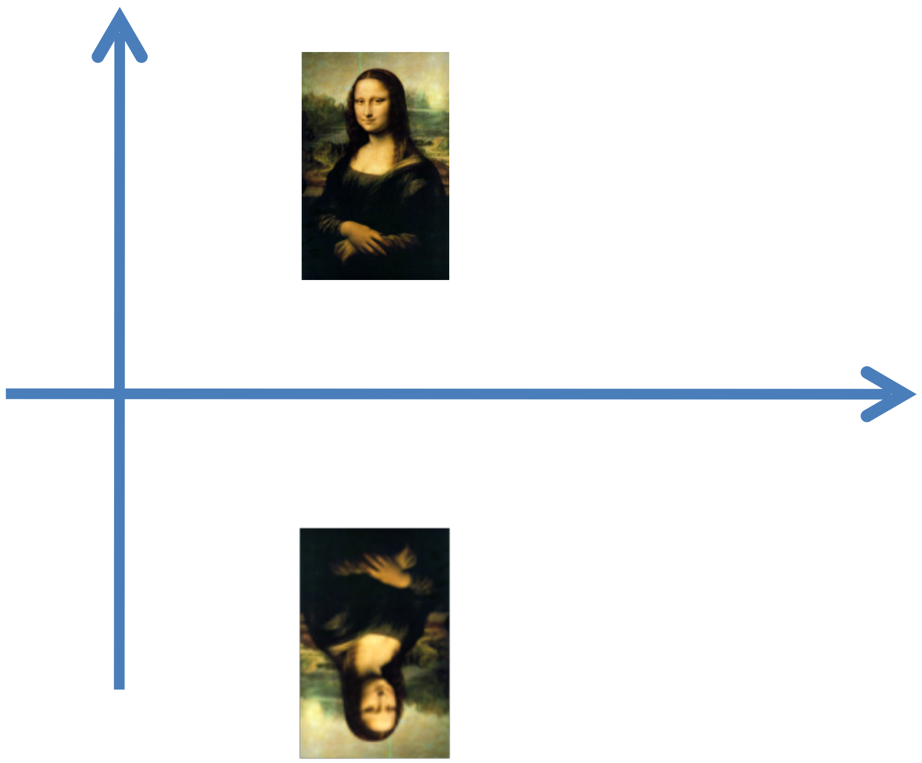
Reflection



Reflection about X-axis

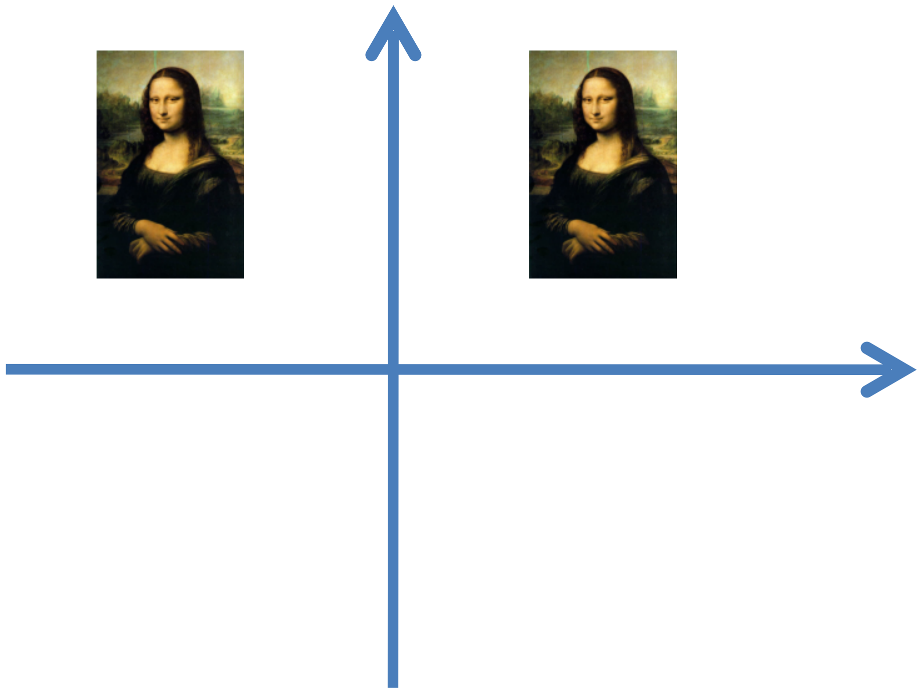


Reflection about X-axis

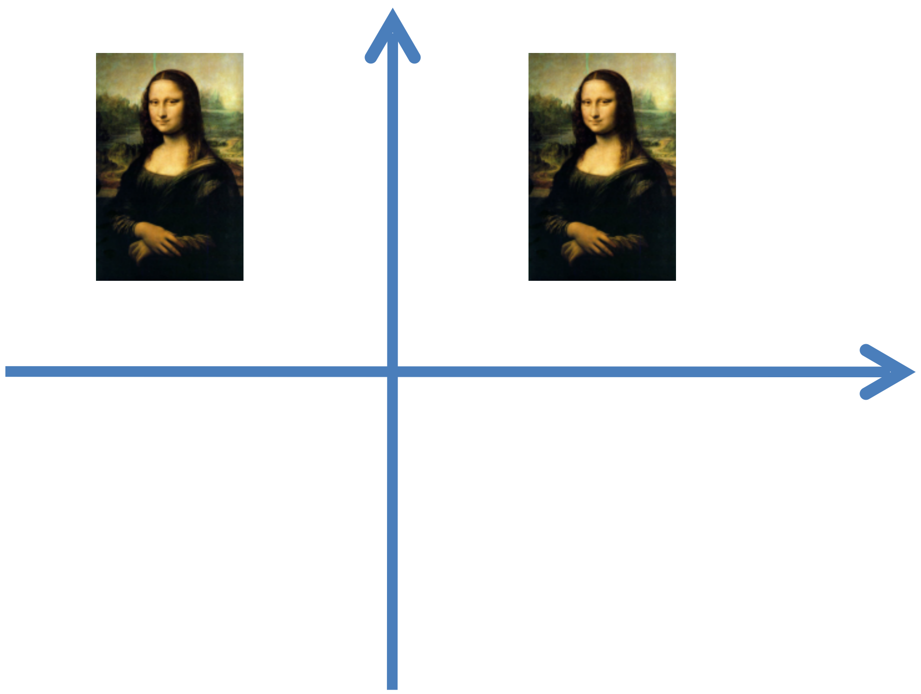


$$\begin{vmatrix} x \\ y \\ 1 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

Reflection about Y-axis

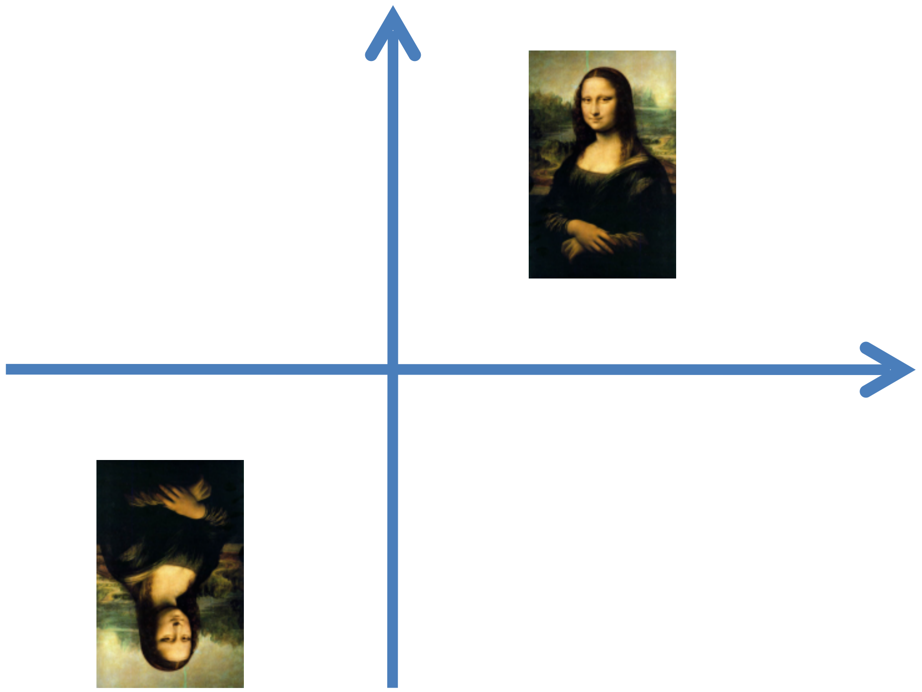


Reflection about Y-axis

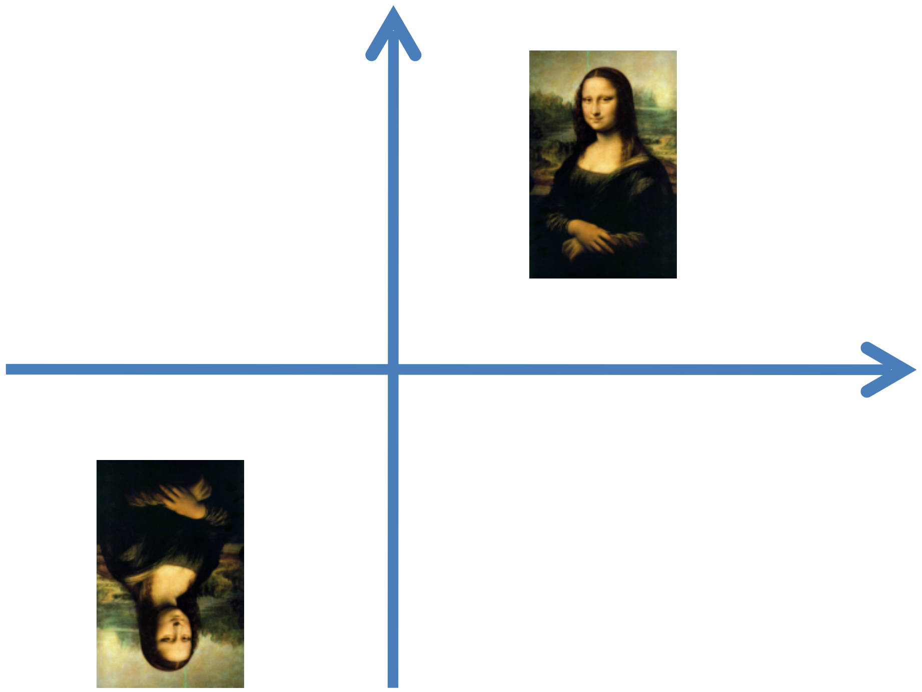


$$\begin{vmatrix} x \\ y \\ 1 \end{vmatrix} = \begin{vmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

What’s the Transformation Matrix?



What's the Transformation Matrix?



$$\begin{vmatrix} x \\ y \\ 1 \end{vmatrix} = \begin{vmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

Image interpolation

Resolution Enhancement

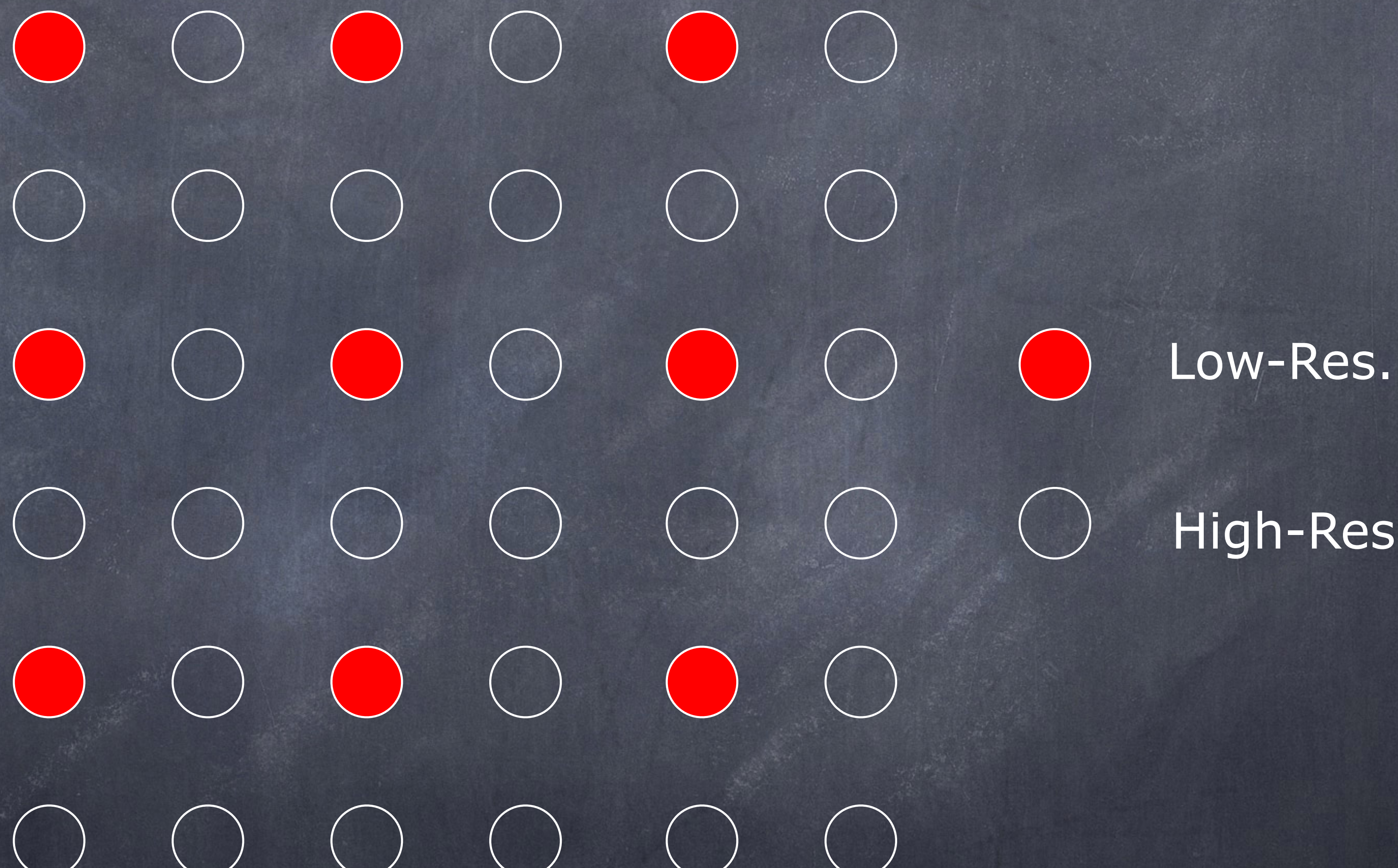
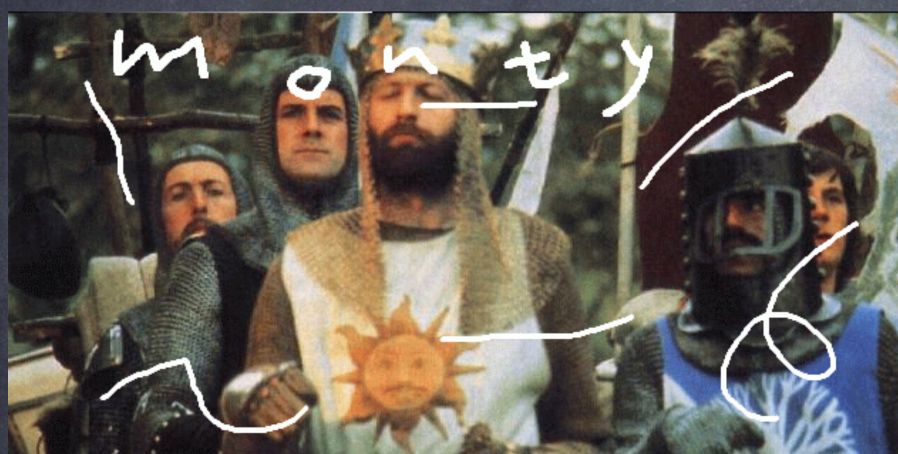
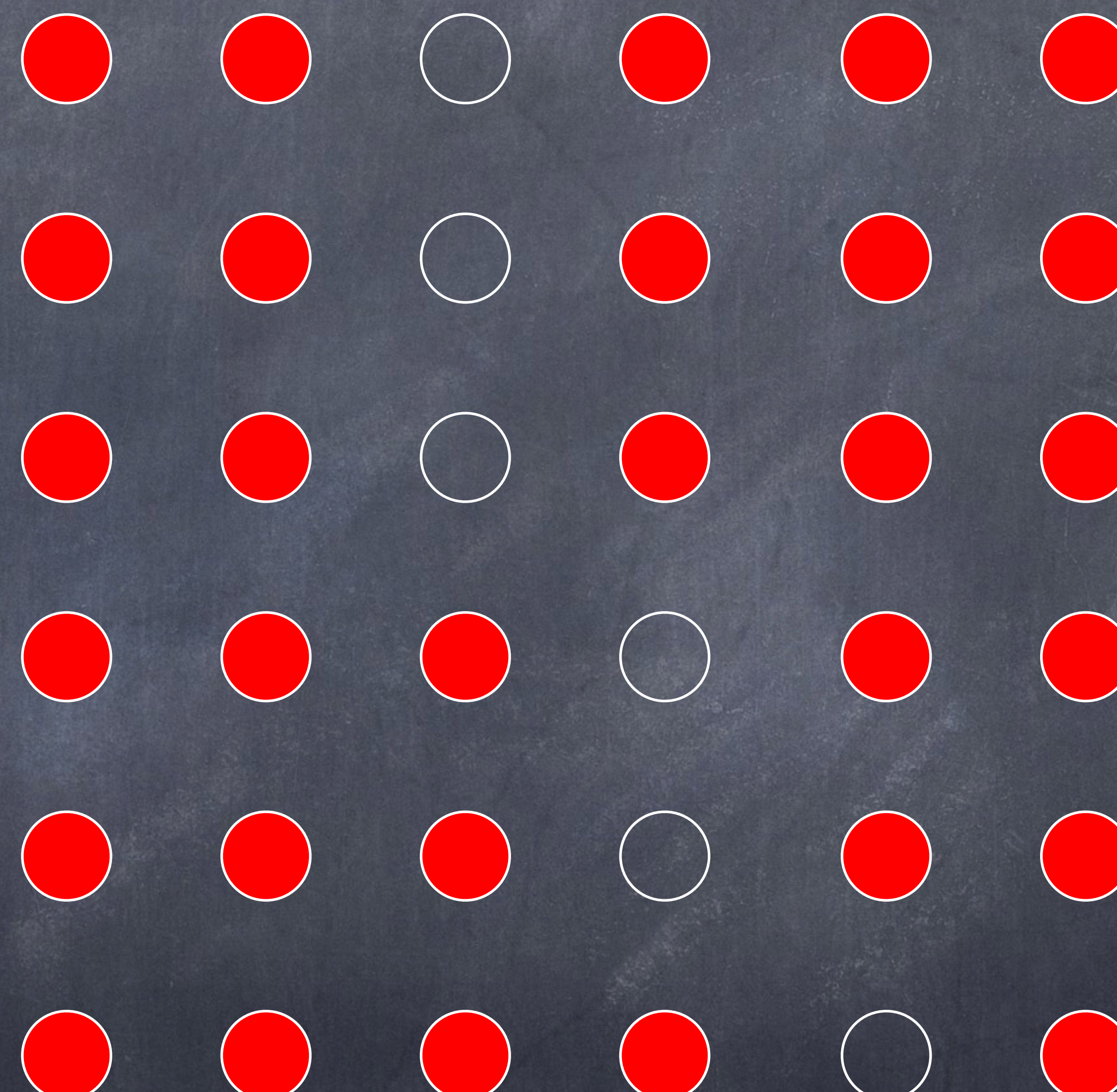


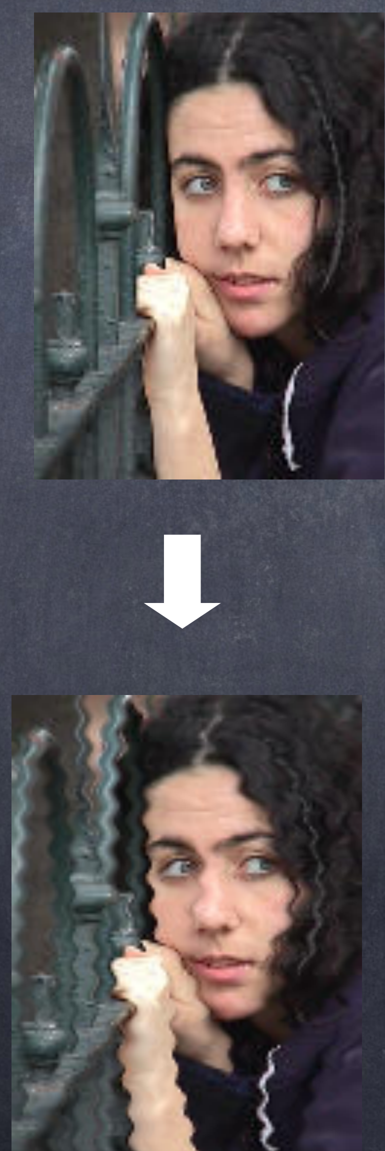
Image Inpainting

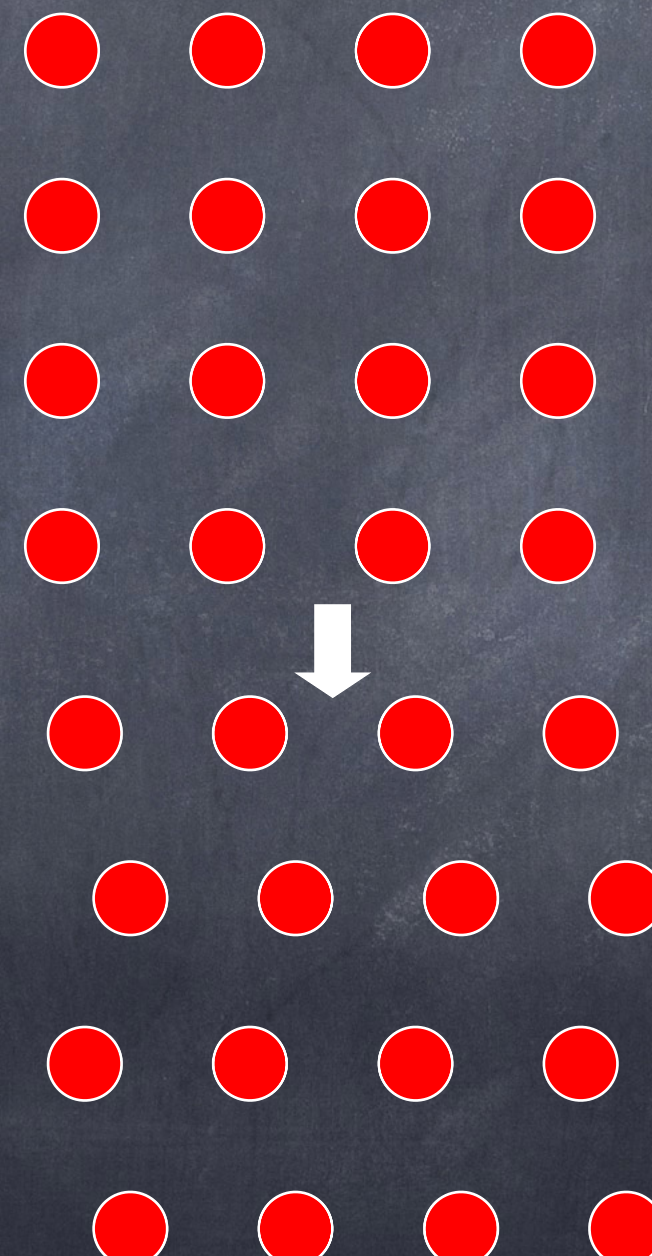




Non-damaged   Damaged

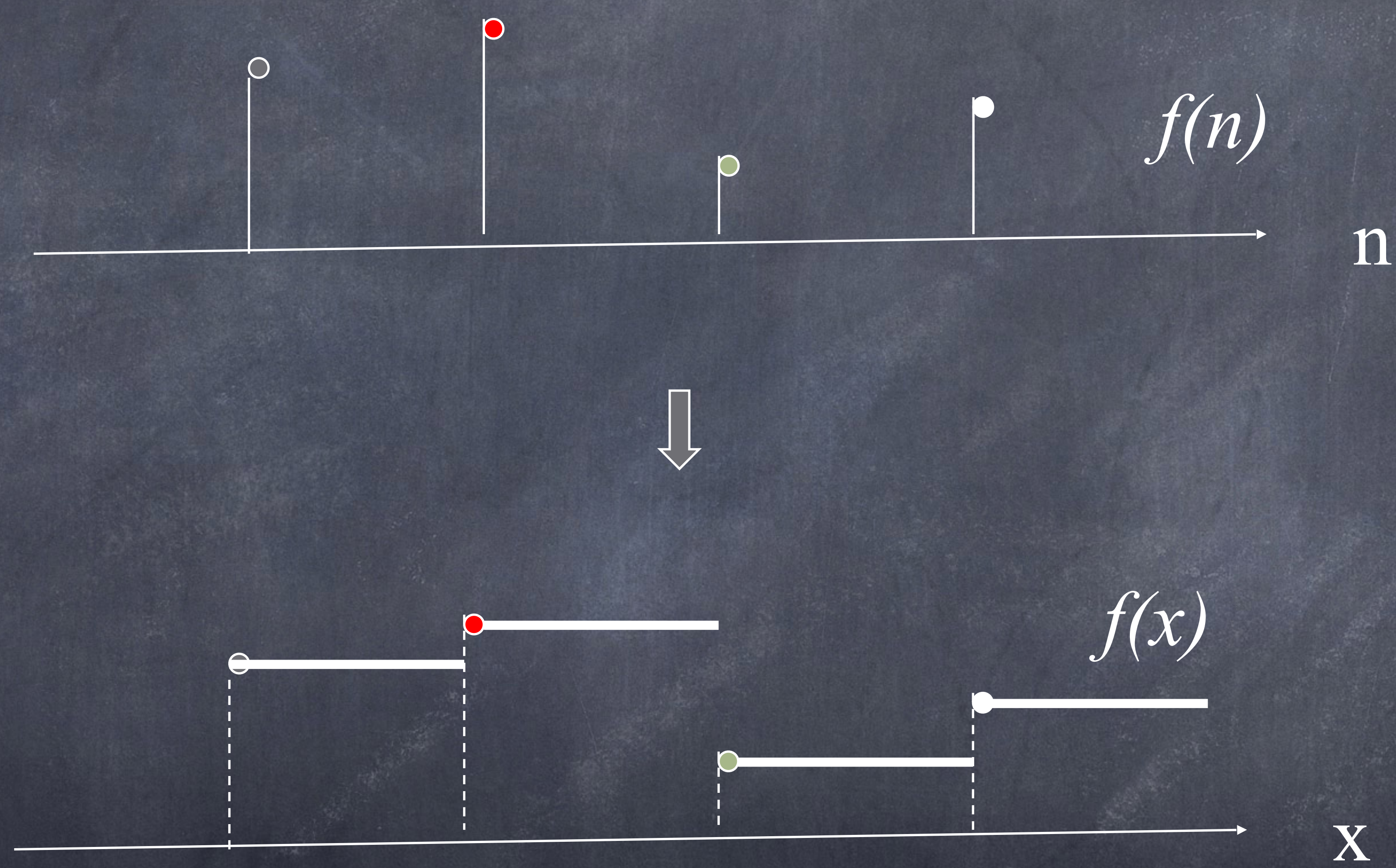
Image Warping



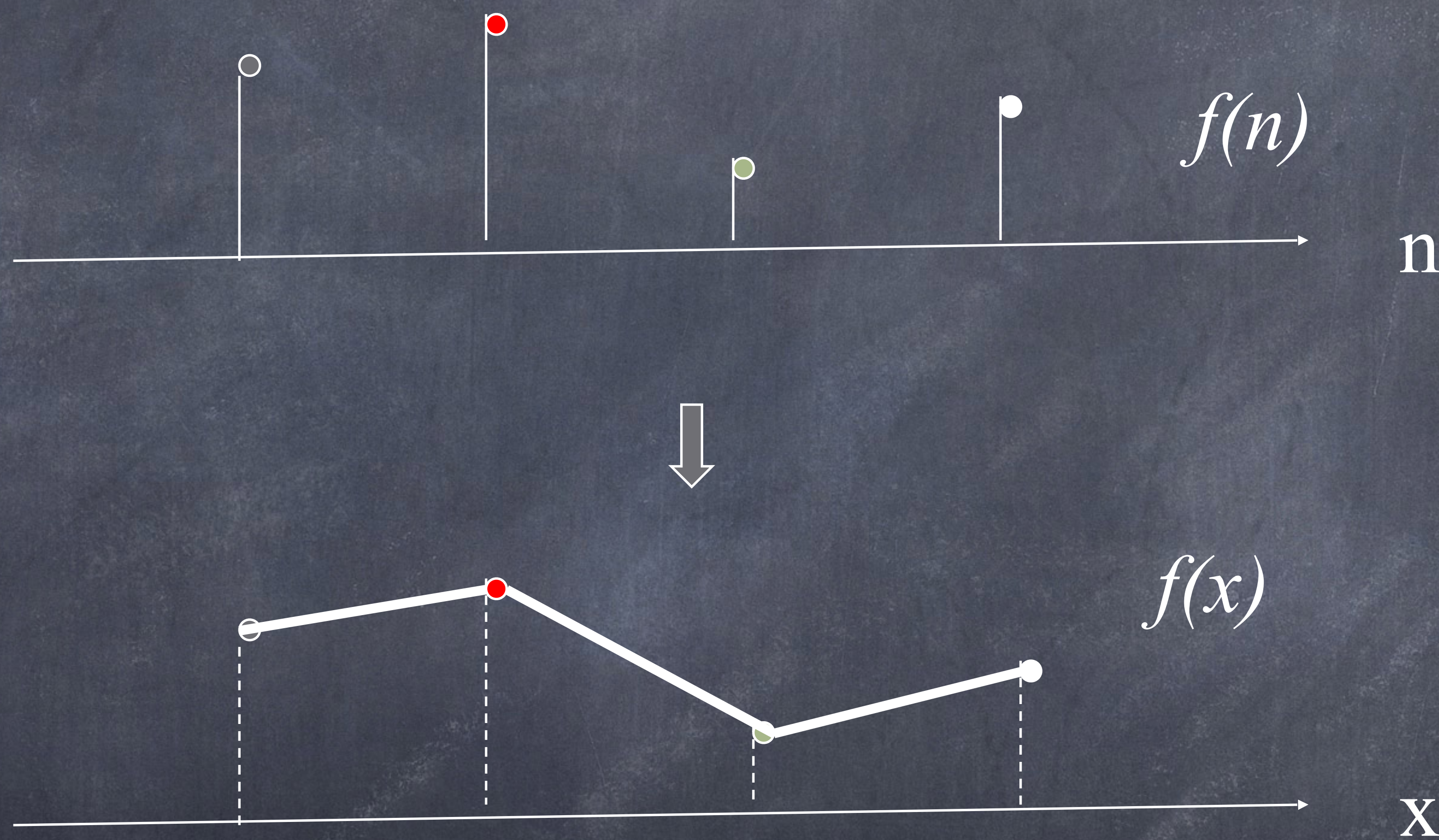


resampling

1D Zero-order (Replication)

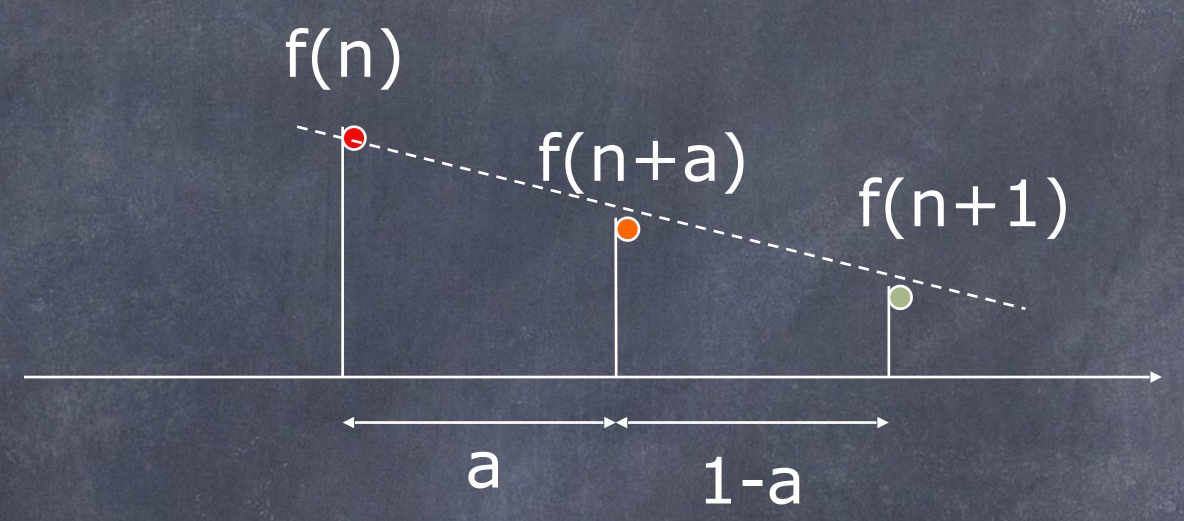


1D First-order Interpolation (Linear)



Linear Interpolation Formula

Heuristic: the closer to a pixel, the higher weight is assigned
Principle: line fitting to polynomial fitting (analytical formula)



$$\boxed{f(n+a)=(1-a)\times f(n)+a\times f(n+1),\ 0<a<1}$$

Note: when a=0.5, we simply have the average of two

Numerical Examples

f(n)=[0,120,180,120,0]

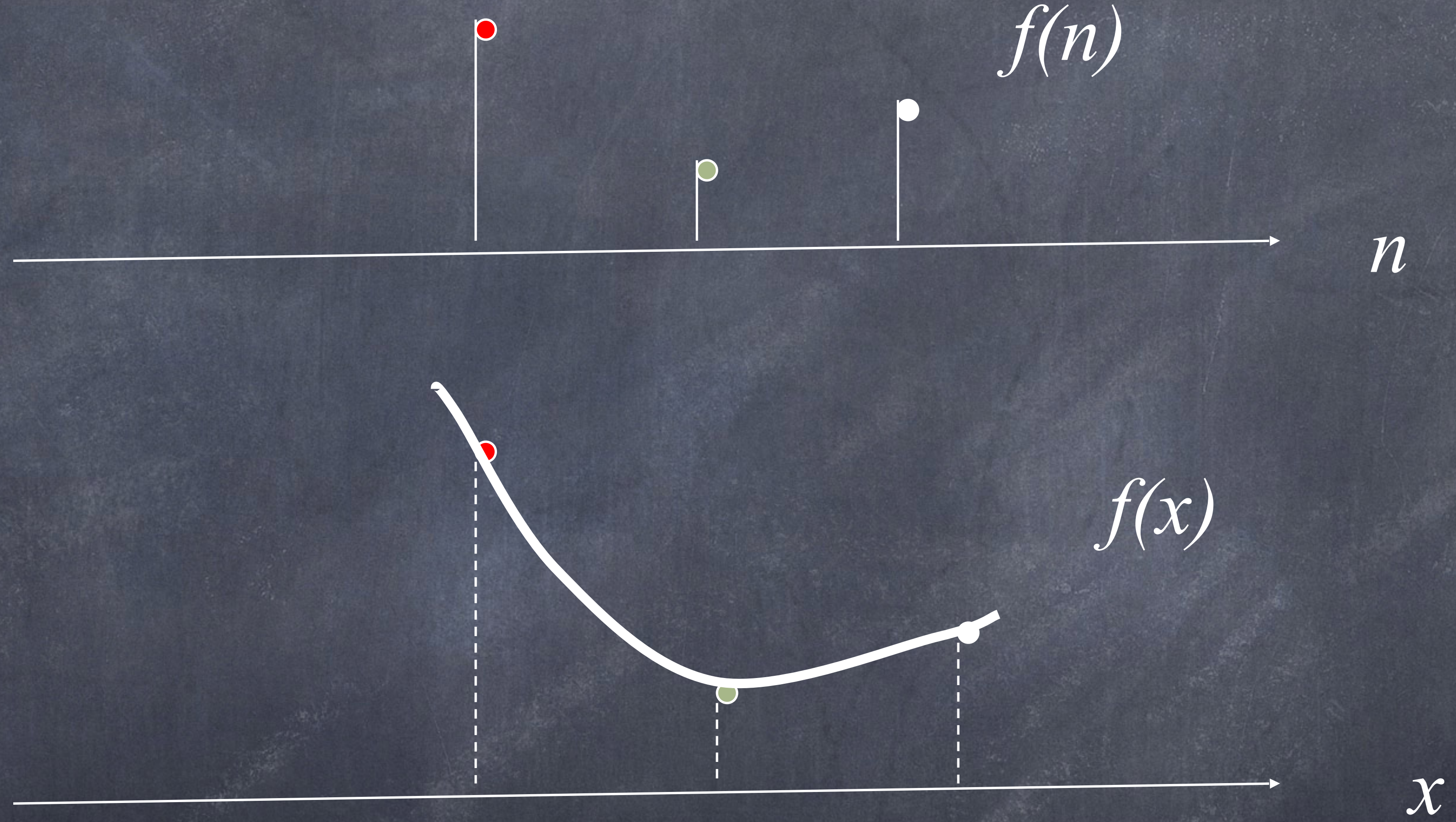
↓ Interpolate at 1/2-pixel

f(x)=[0,60,120,150,180,150,120,60,0], x=n/2

↓ Interpolate at 1/3-pixel

f(x)=[0,20,40,60,80,100,120,130,140,150,160,170,180,…], x=n/6

1D Third-order Interpolation (Cubic)*



Cubic spline fitting

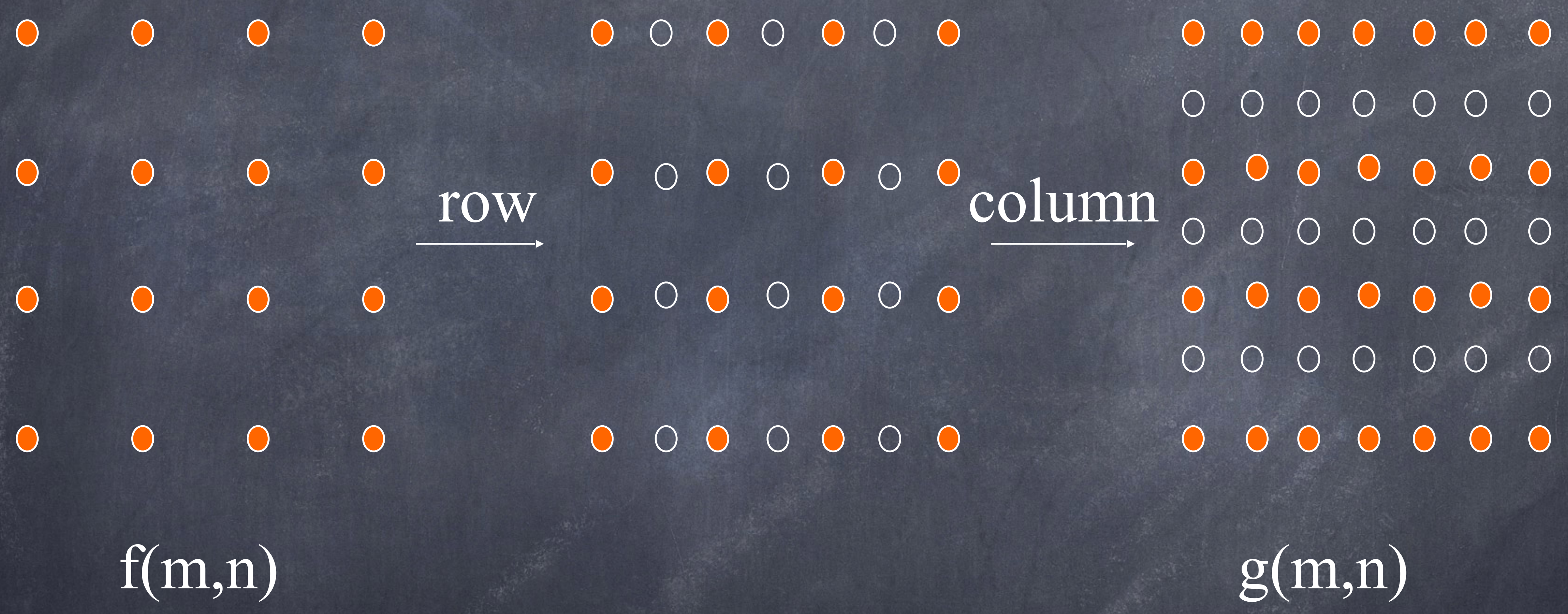
http://en.wikipedia.org/wiki/Spline_interpolation

From 1D to 2D

- Engineers' wisdom: divide and conquer
    - 2D interpolation can be decomposed into two sequential 1D interpolations.
    - The ordering does not matter (row-column = column-row)
    - Such separable implementation is not optimal but enjoys low computational complexity

"If you don't know how to solve a problem, there must be a related but easier problem you know how to solve. See if you can reduce the problem to the easier one." - rephrased from G. Polya's "How to Solve It"

Graphical Interpretation of Interpolation at Half-pel



Numerical Examples

a	b
c	d

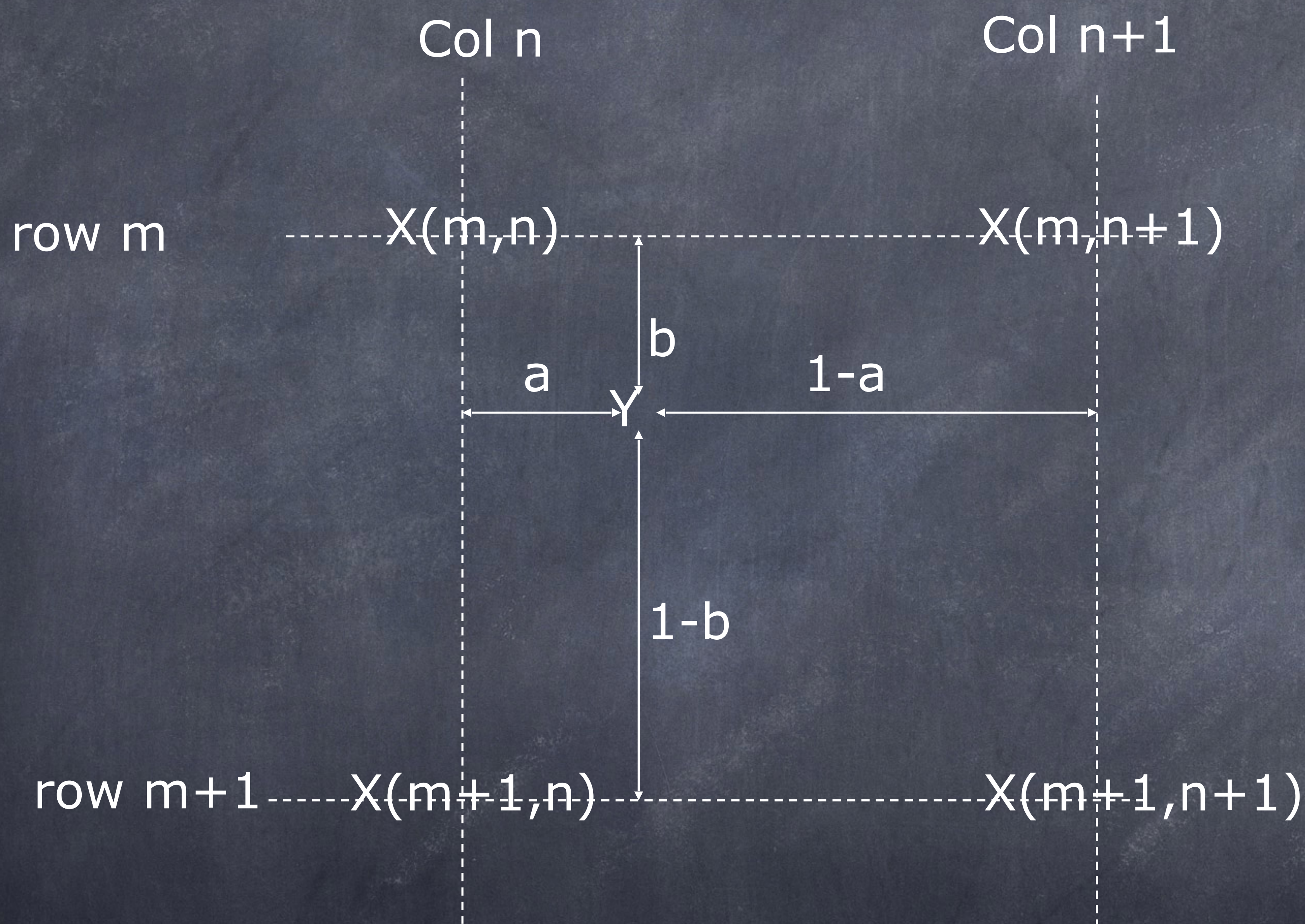
zero-order

a	a	b	b
a	a	b	b
c	c	d	d
c	c	d	d

first-order

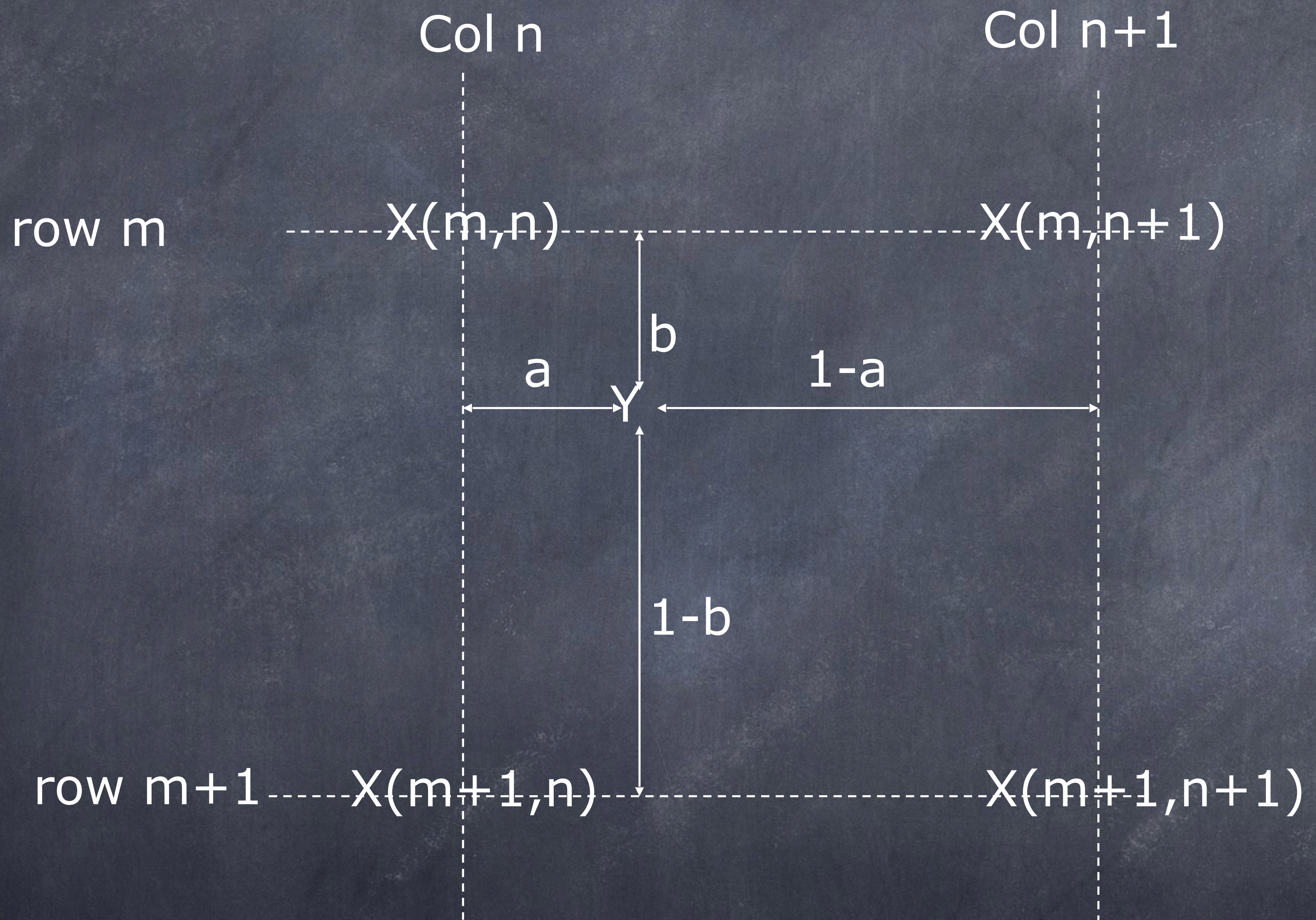
a	(a+b)/2	b
(a+c)/2	(a+b+c+d)/4	(b+d)/2
c	(c+d)/2	d

Numerical Examples



Q: what is the interpolated value at Y?

Numerical Examples



Q: what is the interpolated value at Y?

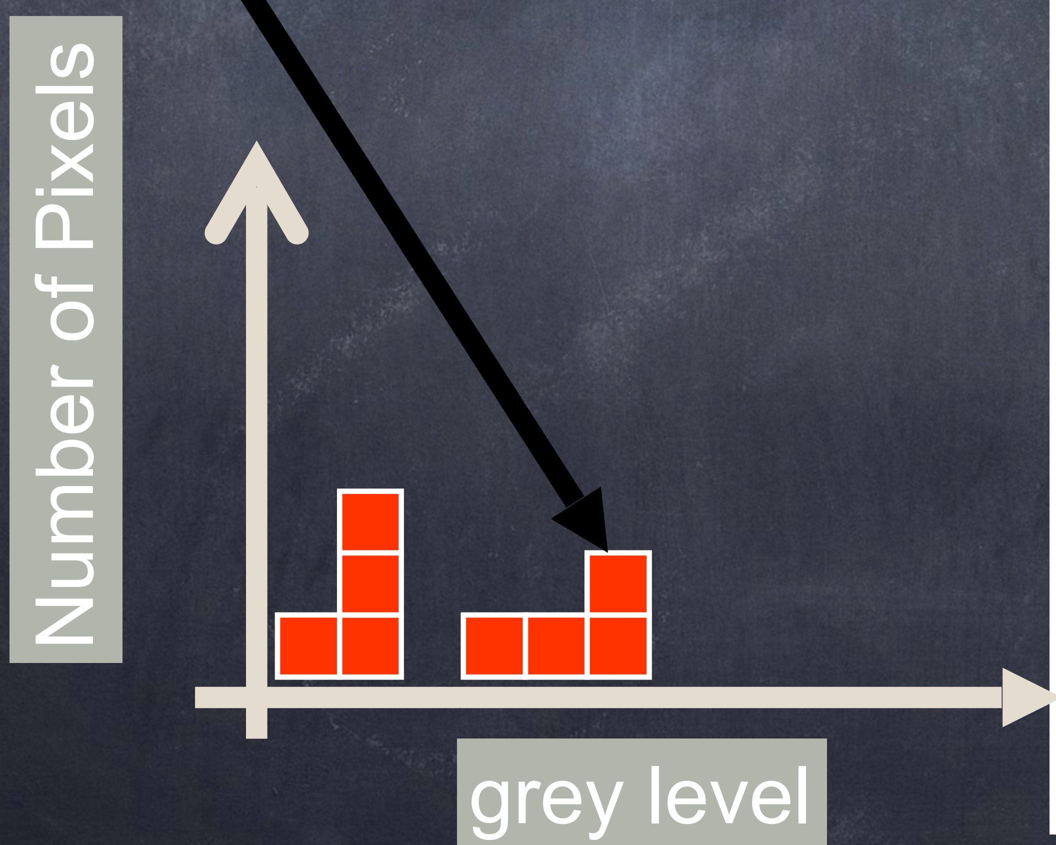
Ans.: $(1-a)(1-b)X(m,n)+(1-a)bX(m+1,n)$
$+a(1-b)X(m,n+1)+abX(m+1,n+1)$

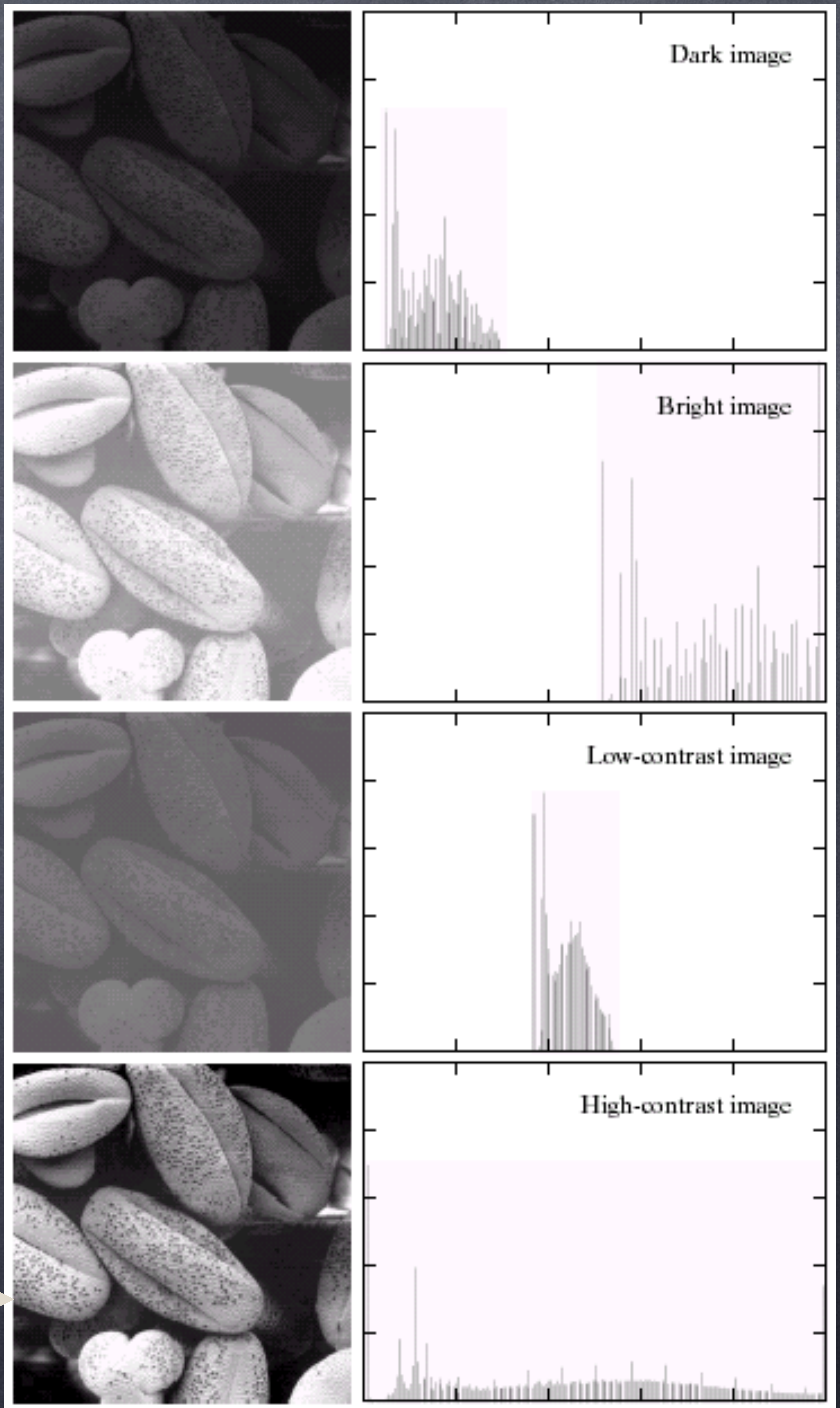
Image Histogram

- What is histogram?
- What is image histogram?

Histogram Processing

1	4	5	0
3	1	5	1





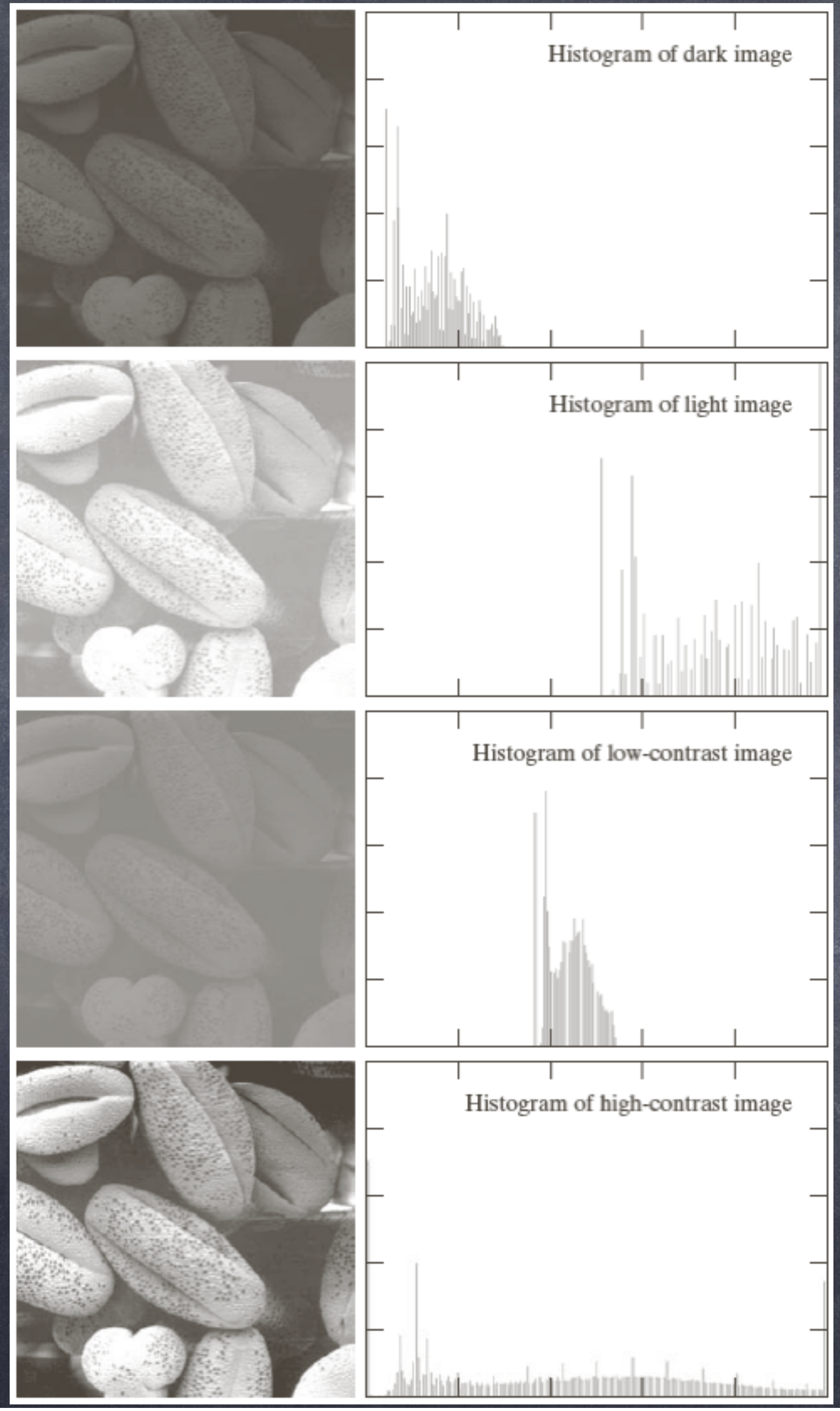


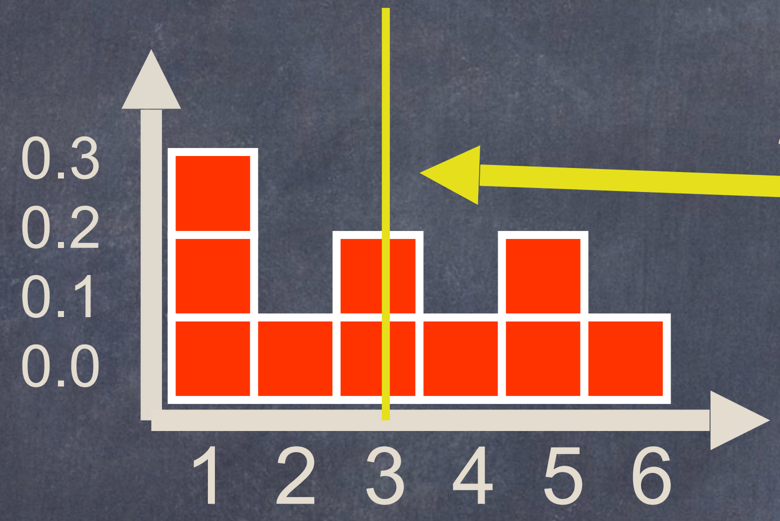
FIGURE 3.16 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms.

The histogram shows the number of pixels having a certain grey-value

The value given by the histogram for a certain grey value can be read as the probability of randomly picking a pixel having that grey value

What can the histogram tell about the image ?

The MEAN VALUE (or average grey level)



1*0.3+2*0.1+3*0.2+4*0.1+5*0.2+6*0.1=
2.6

The MEAN value is the average gray value of the image, the ‘overall brightness appearance’.

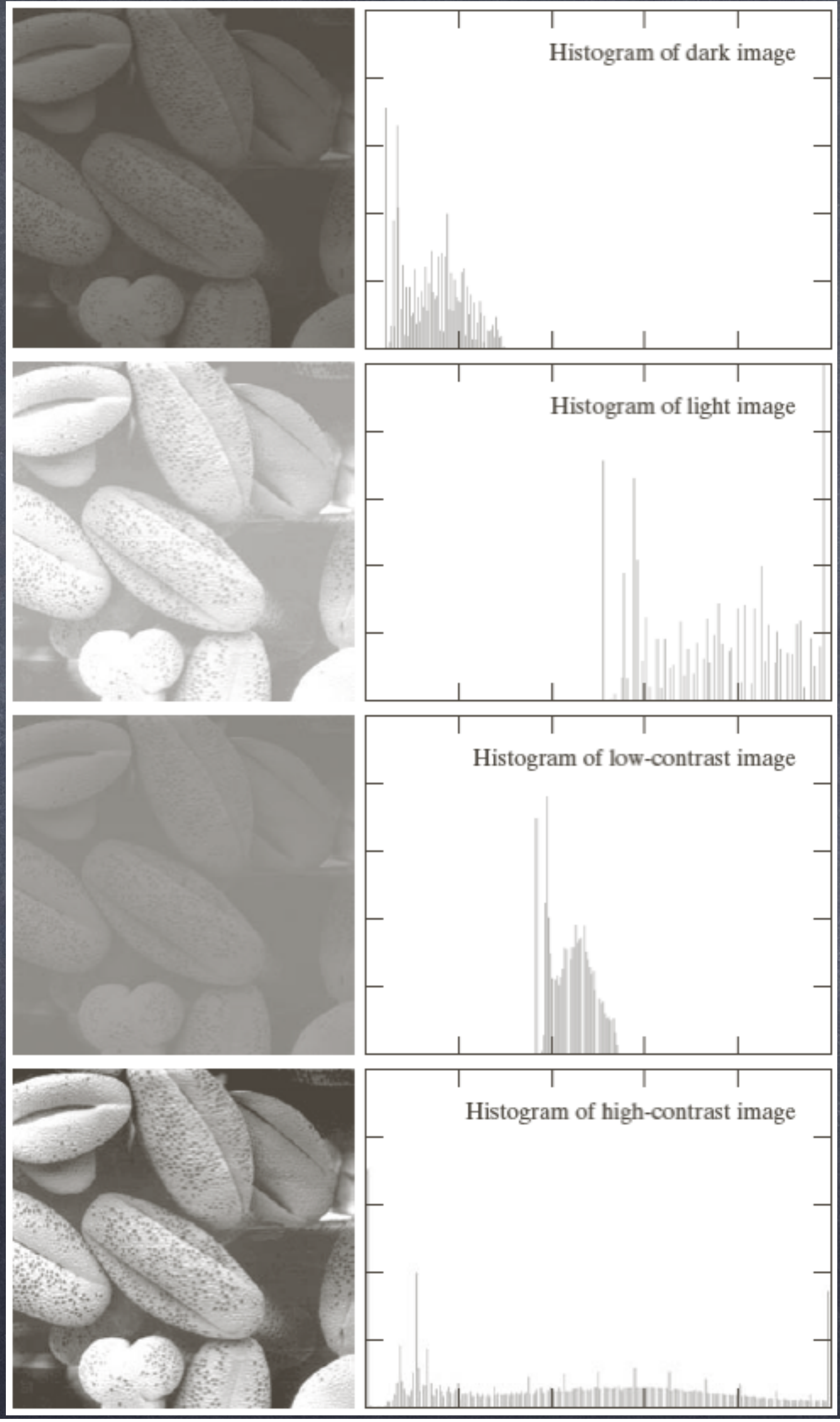
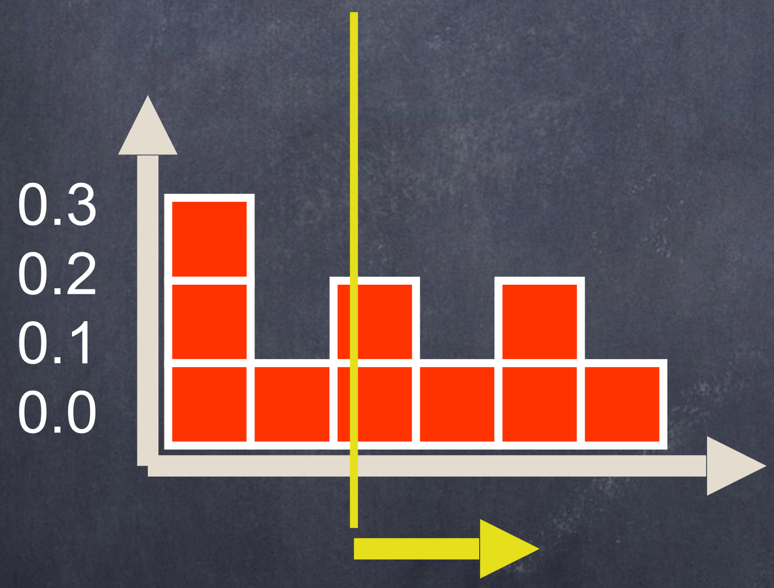
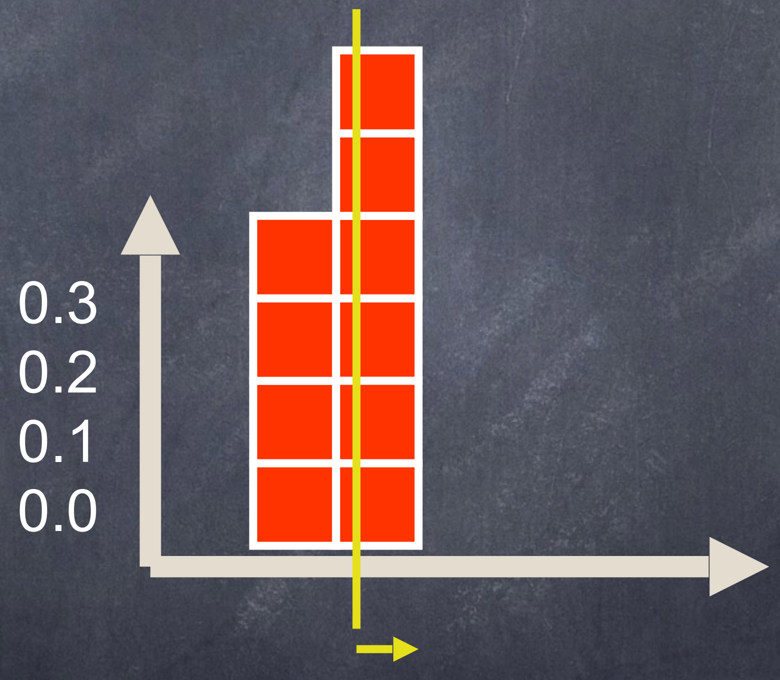


FIGURE 3.16 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms.

VARIANCE gives a measure about the distribution of the histogram values around the mean.

The STANDARD DEVIATION is a value on the gray level axis, showing the average distance of all pixels to the mean





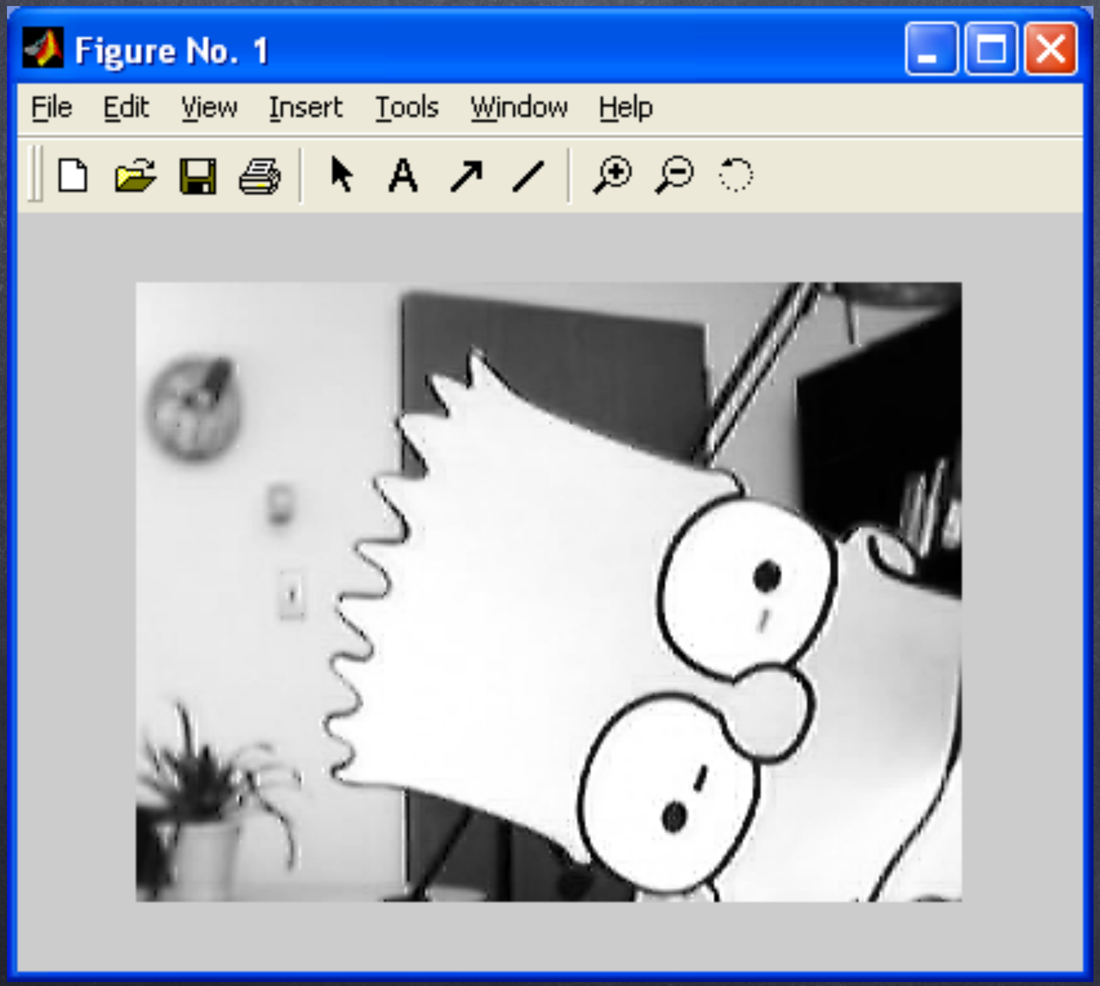
D1 > D2

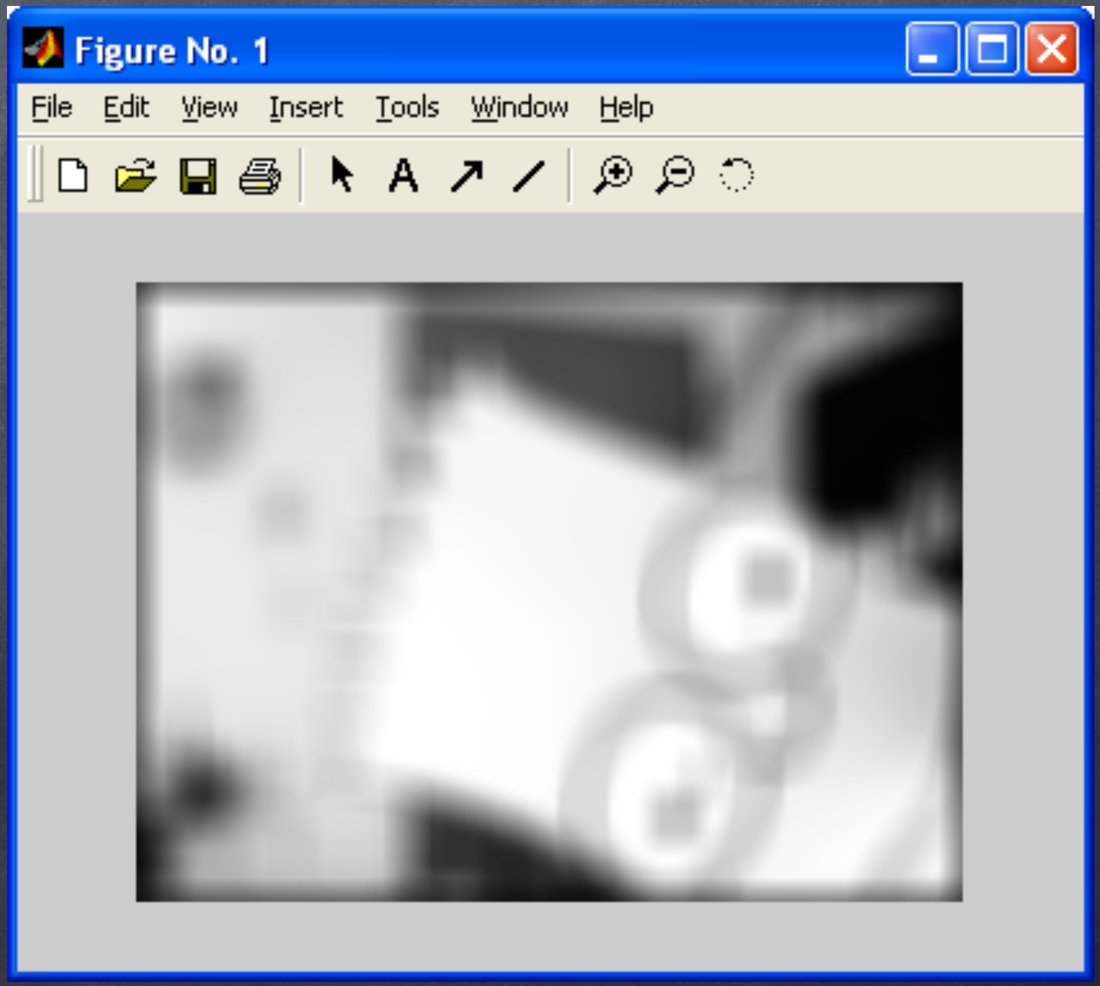
VARIANCE and **STANDARD DEVIATION** of the histogram tell us about the average contrast of the image !

The higher the VARIANCE (=the higher the STANDARD DEVIATION), the higher the image's contrast !

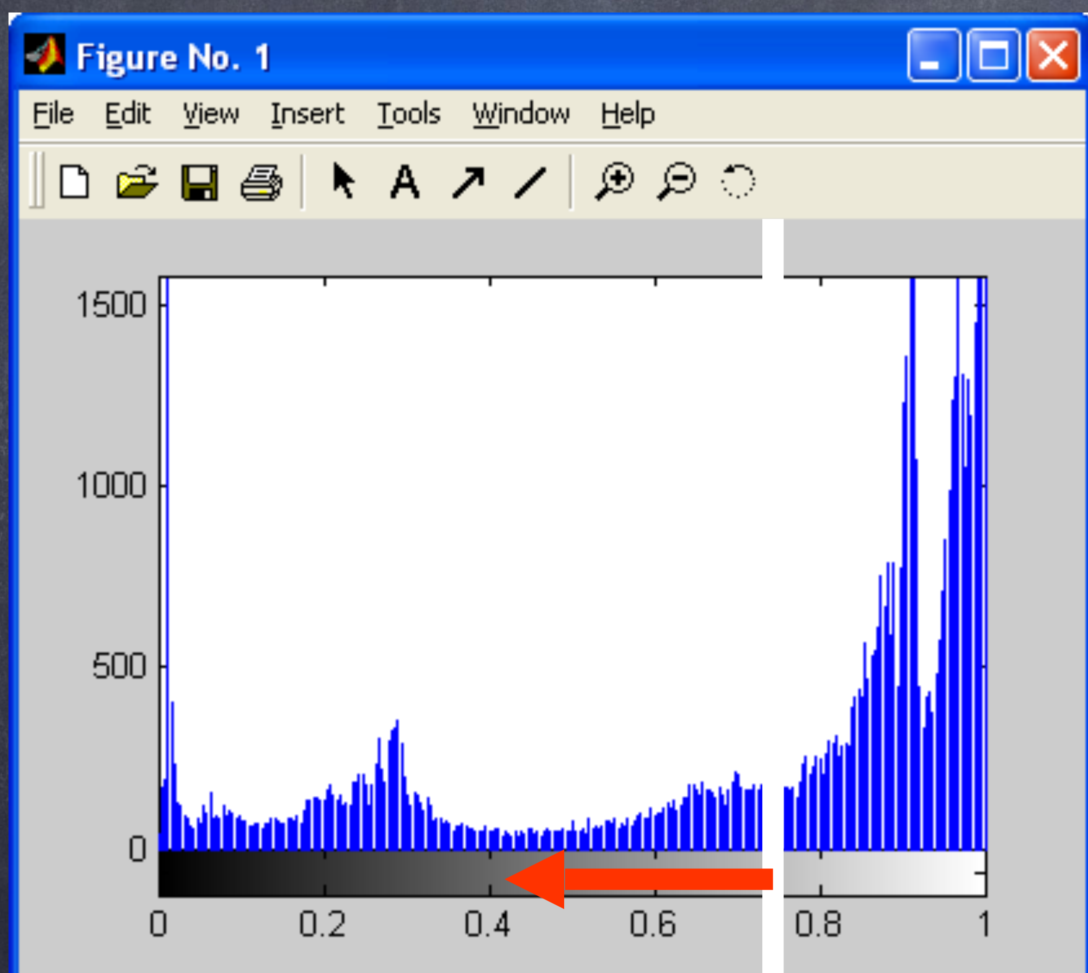
Example:

Image and blurred version

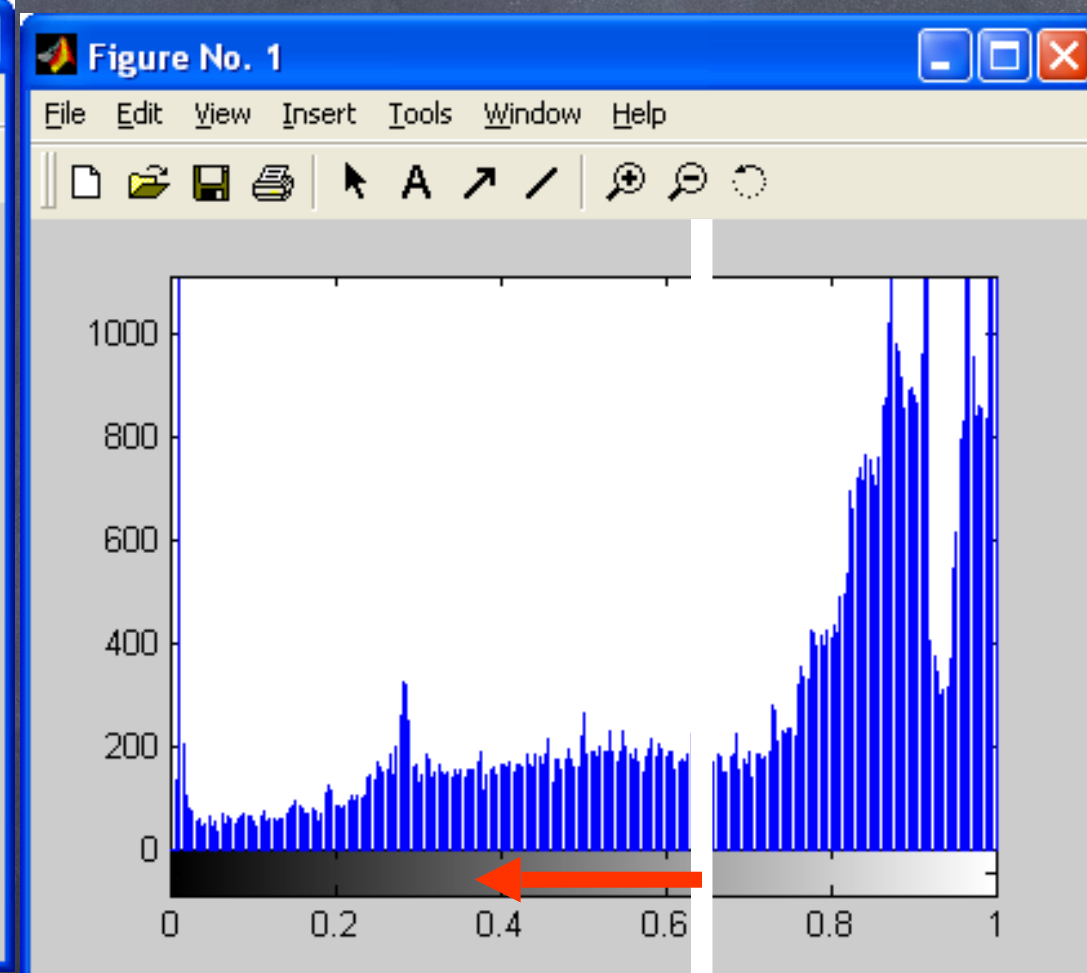




Histograms with MEAN and STANDARD DEVIATION



M=0.73 D=0.32



M=0.71 D=0.27

Design an autofocus system for a digital camera !

Next Week

Convolution/filtering, Pyramids and Wavelet Transforms