Digital Signal Processing

Lecture 12 – digital image processing 2

상명대학교
컴퓨터과학과
강상욱 교수

INTERPOLATION

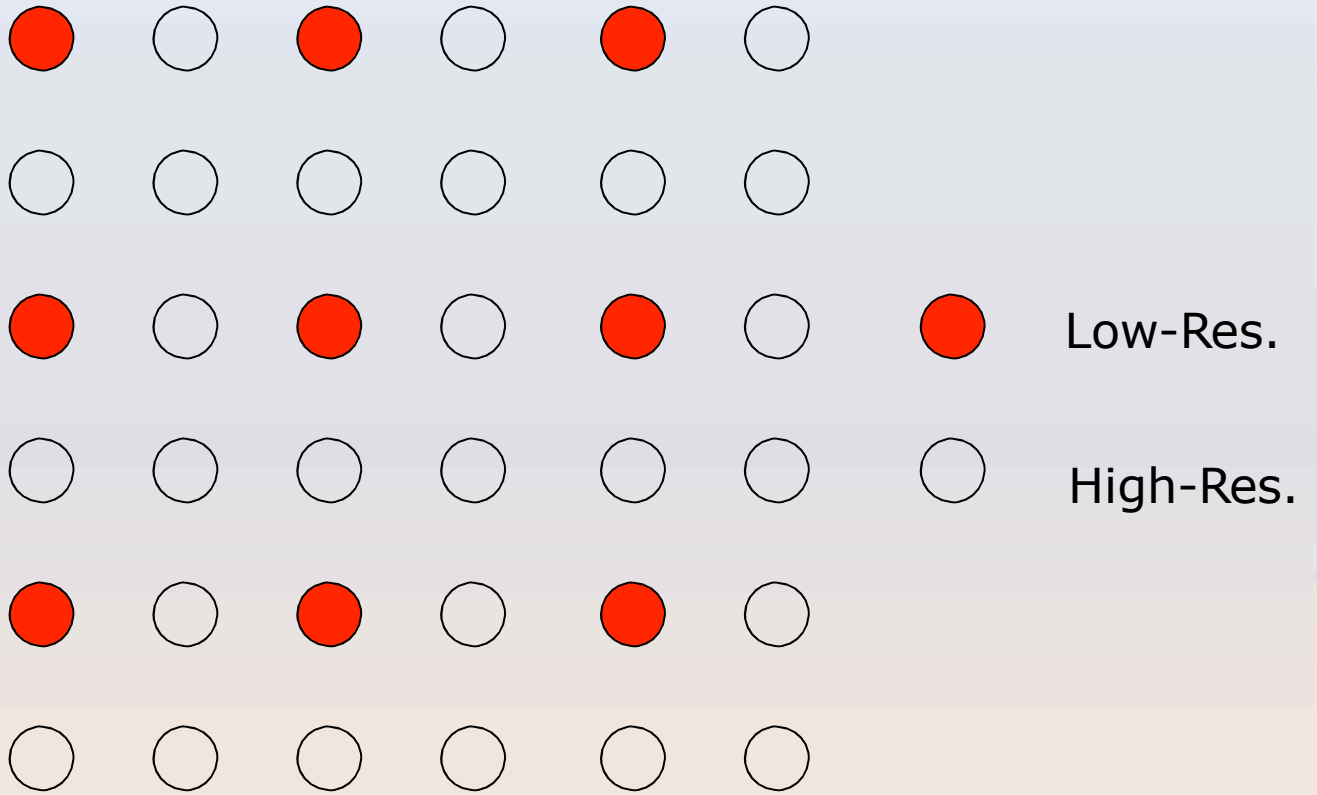
What is image interpolation?

- An image f(x,y) tells us the intensity values when x and y are both integers.
- Image interpolation refers to the "guess" of intensity values at missing locations, i.e., x and y can be arbitrary.
- Note that it is just a guess. (Note that all sensors have finite sampling distance)
- It is about Digital to Analog conversion.

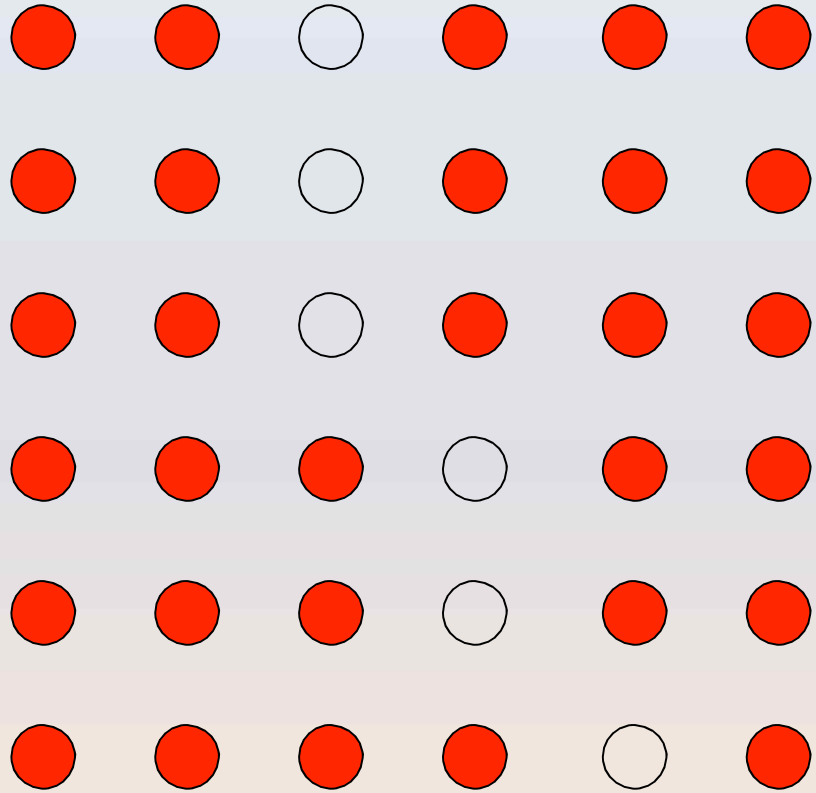
Why need image interpolation?

- We want big images
  - When we see a video clip on a PC, we like to see it in the full screen mode.
  - Sometime, we need a smaller image as well.
- We want good images
  - If some block of an image gets damaged during the transmission, we want to repair it.
- We want cool images
  - Image can be rotated, warped or screwed to make funny and cool images.

Ex 1 – resolution enhancement

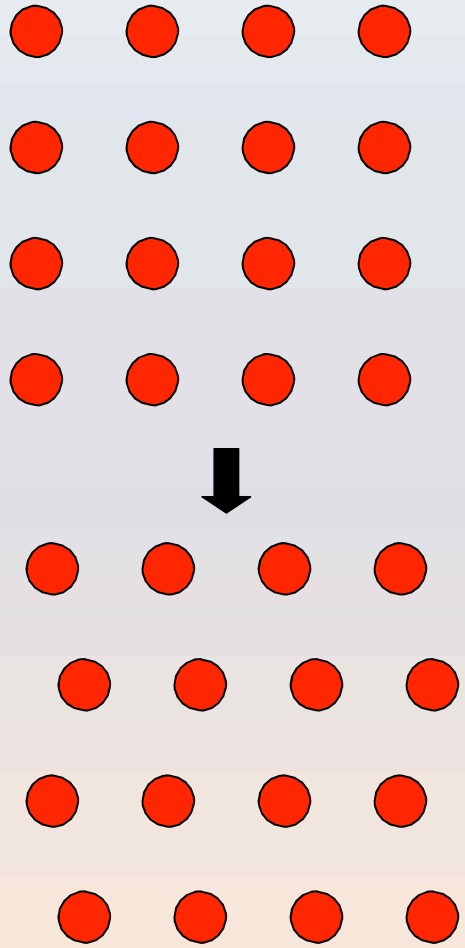


Ex 2 – Image inpainting



Non-damaged Damaged

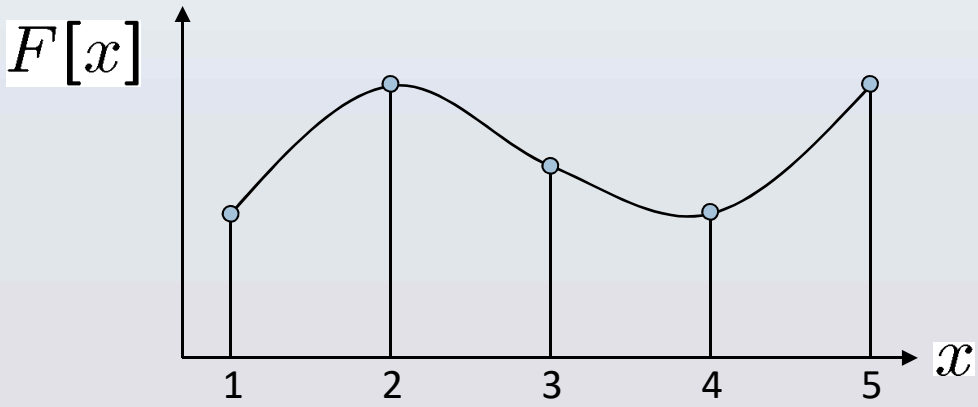
Ex 3 – Image warping



Upsampling

- This image is too small for this screen: 
- How can we make it 10 times as big?
- The simplest approach: repeat each row and column 10 times
- “Nearest neighbor interpolation”

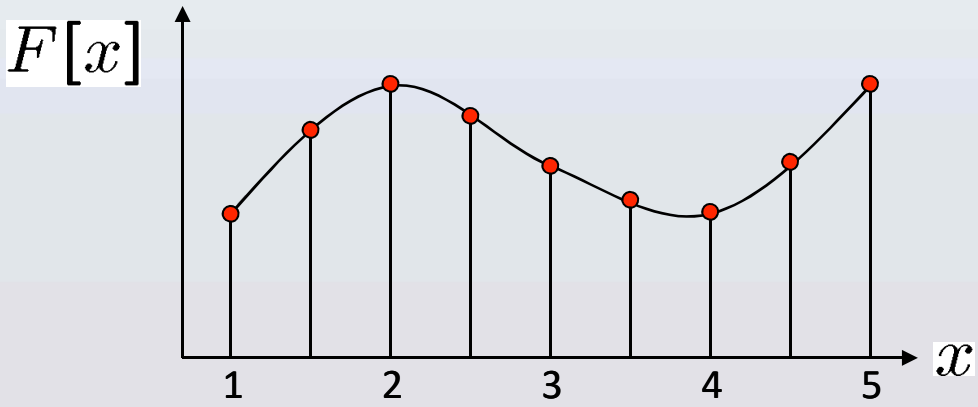
Interpolation in 1-D



d = 1 in this example

- It is a discrete point--sampling of a continuous function
- If we could somehow reconstruct the original function, any new image could be generated, at any resolution and scale

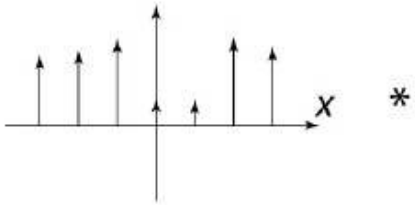
Interpolation in 1-D



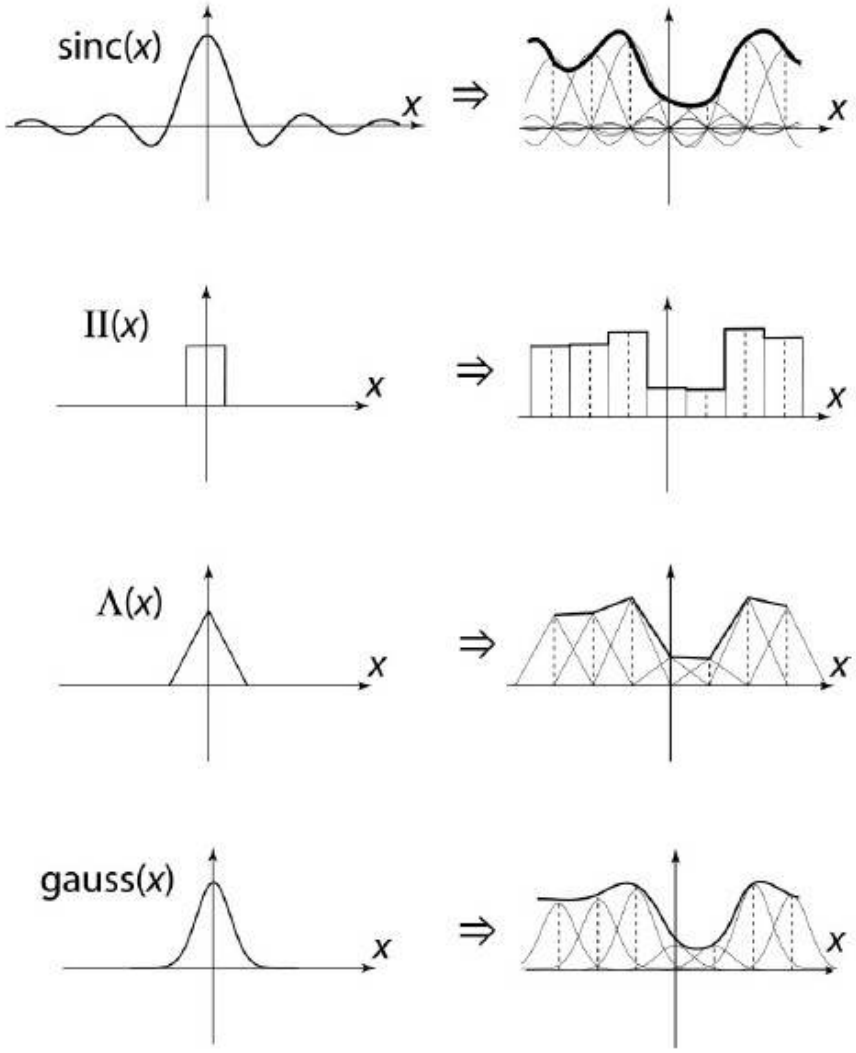
d = 1 in this example

- It is a discrete point--sampling of a continuous function
- If we could somehow reconstruct the original function, any new image could be generated, at any resolution and scale

Interpolation in 1-D



*



"Ideal" reconstruction

Nearest-neighbor interpolation

Linear interpolation

Gaussian reconstruction

Interpolation in 1-D

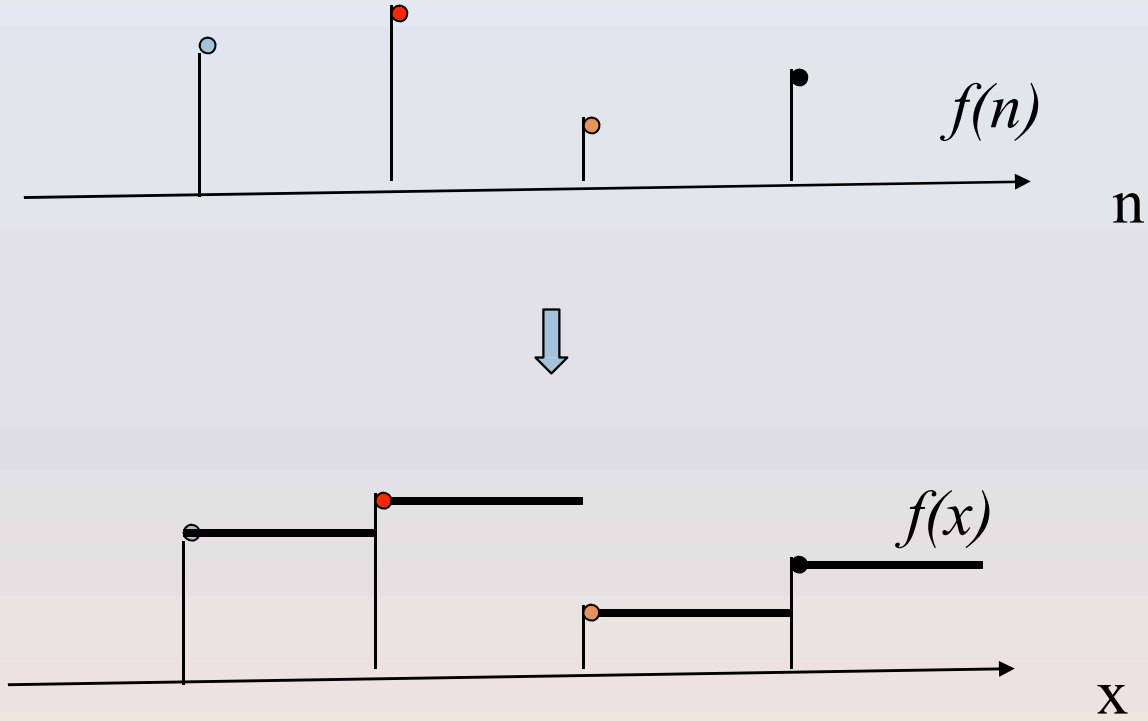
Original image:  x 10

Nearest-neighbor interpolation

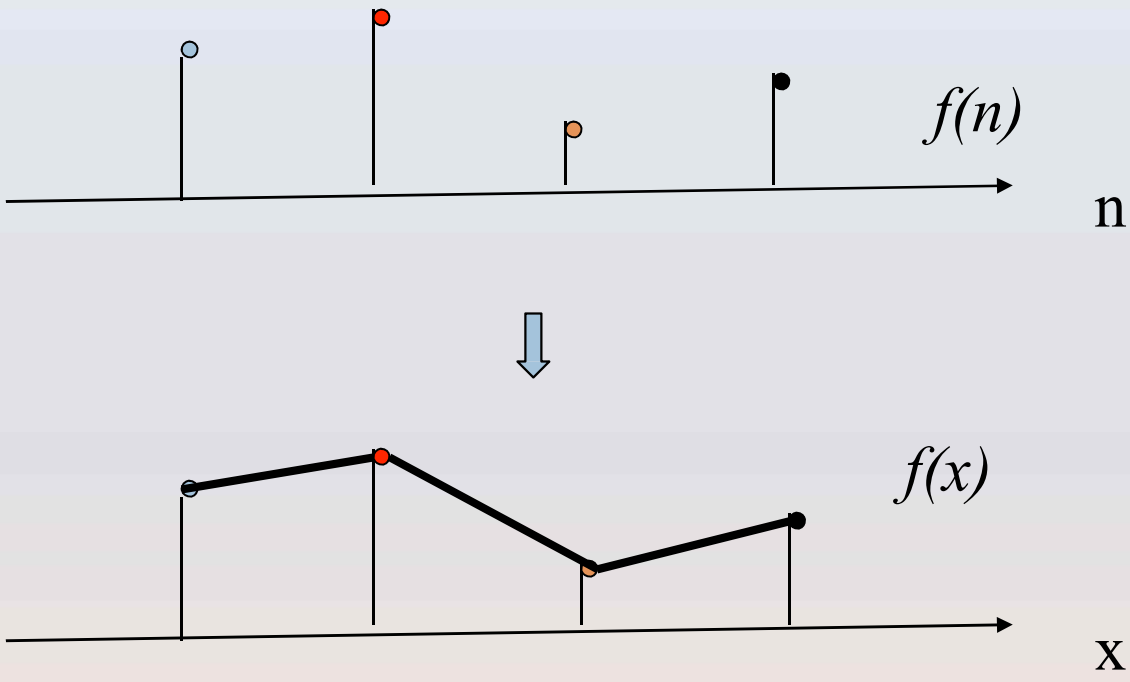
Bilinear interpolation

Bicubic interpolation

1-D zero-order (replication)

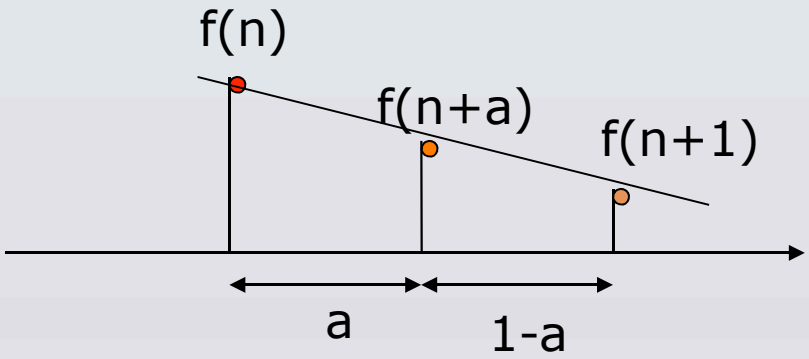


1-D first-order (Linear)



Linear interpolation example

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



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

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

Linear interpolation formula

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

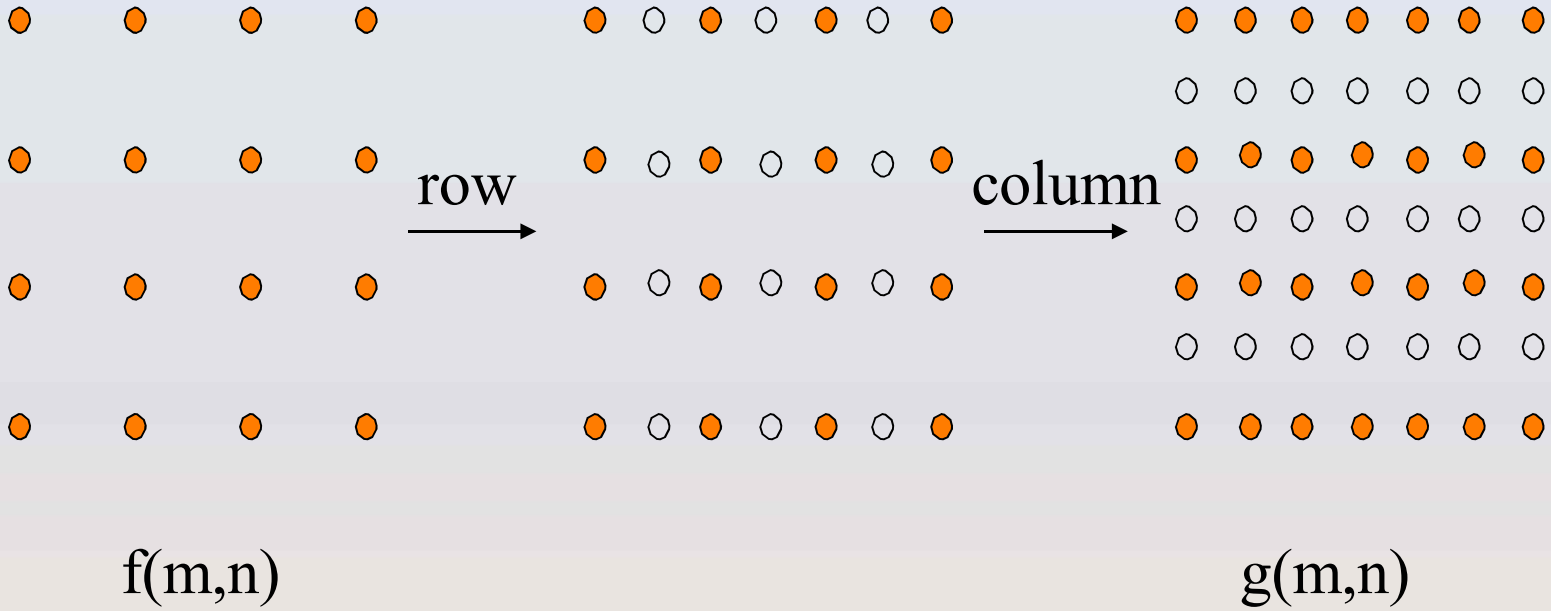
↓ Interpolate at 1/2-pixel

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

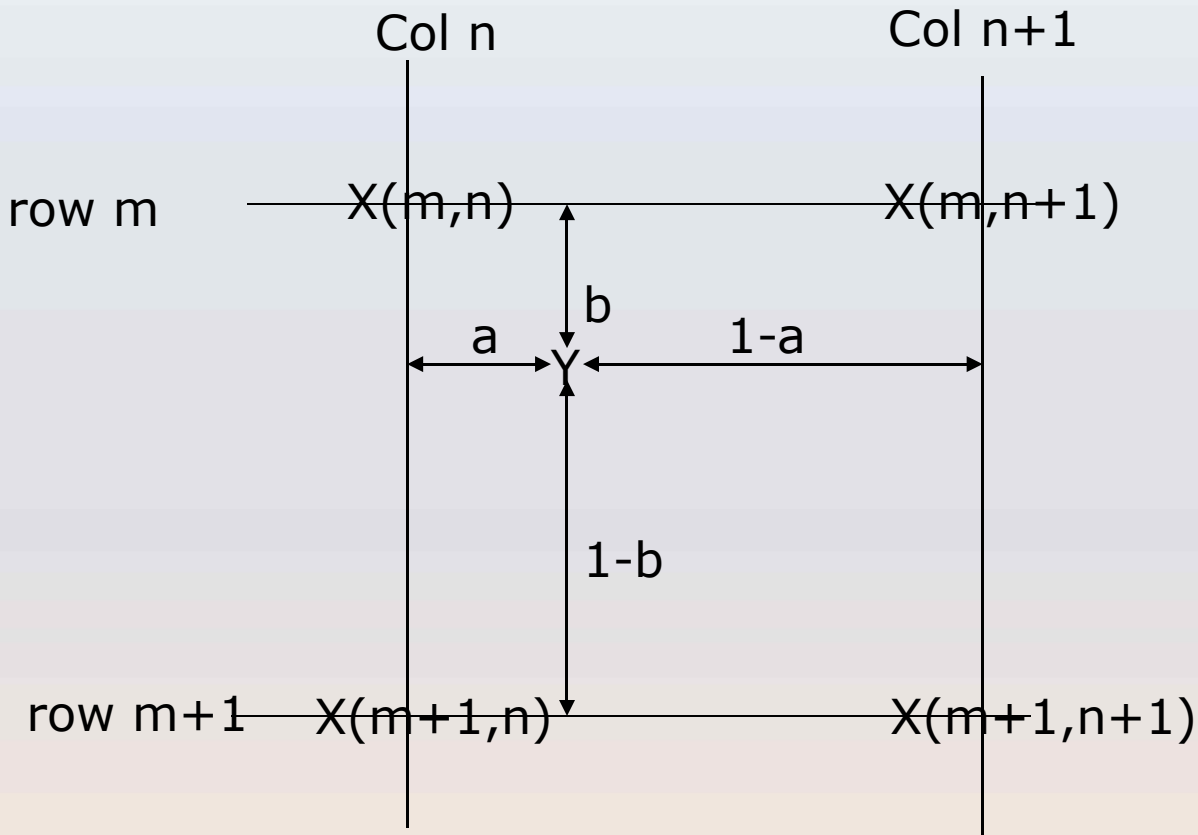
↓ Interpolate at 1/3-pixel

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

From 1-D to 2-D

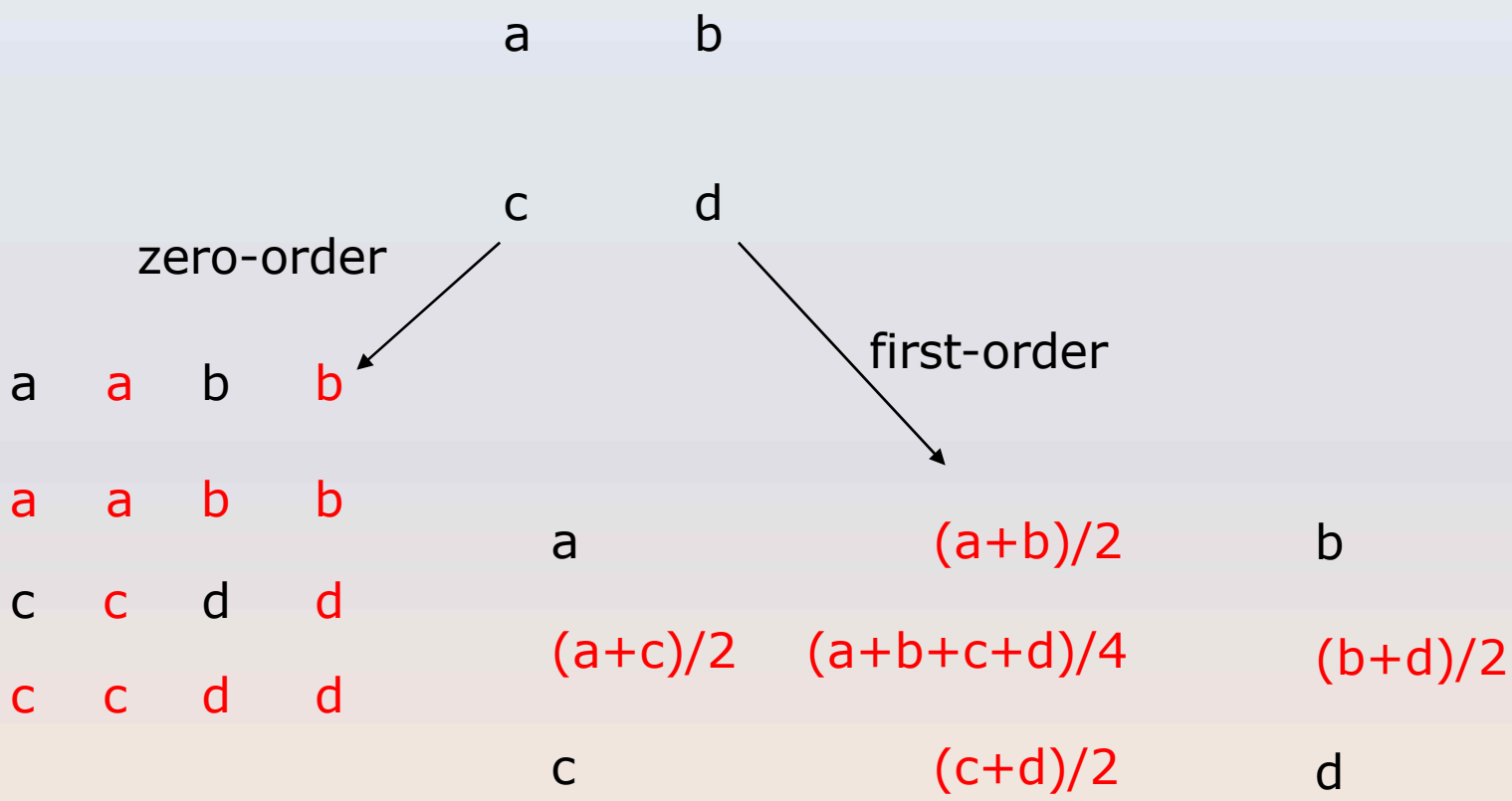


From 1-D to 2-D example



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)

From 1-D to 2-D example



LAB

- Scaling
  - magnification
  - reduction
- Rotation