

# Digital Image Processing (CSE/ECE 478)

## Lecture # 06: Image Resampling

Avinash Sharma

Center for Visual Information Technology (CVIT),  
IIIT Hyderabad



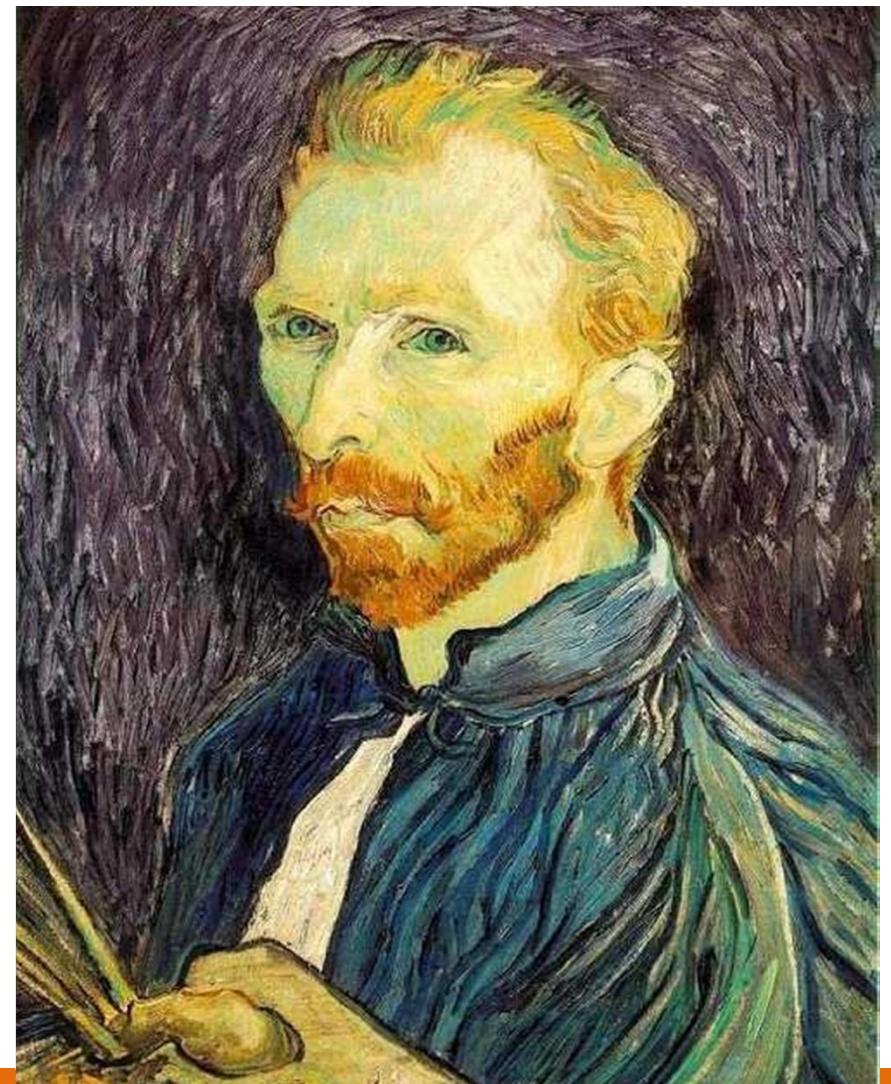
# Today's Class

- Image down sampling
- Gaussian Pyramids
- Image up sampling

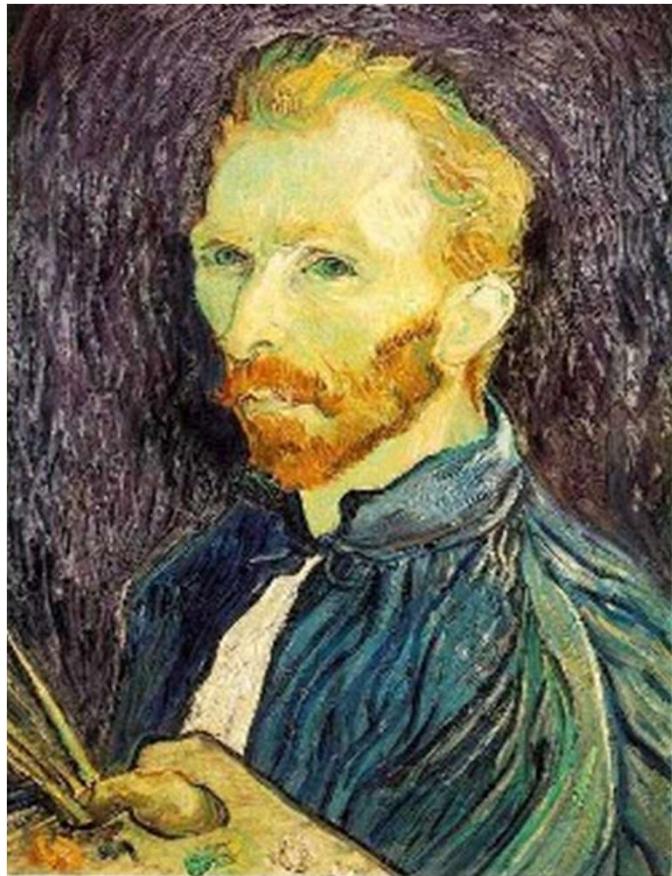


Image too big to fit screen. How can we resize?

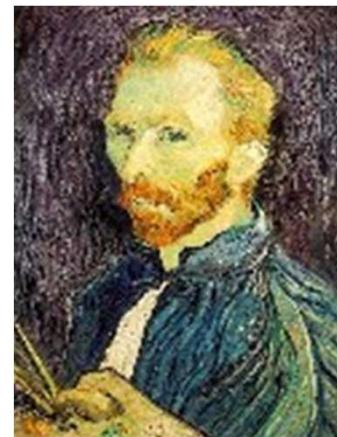
Source: S. Seitz



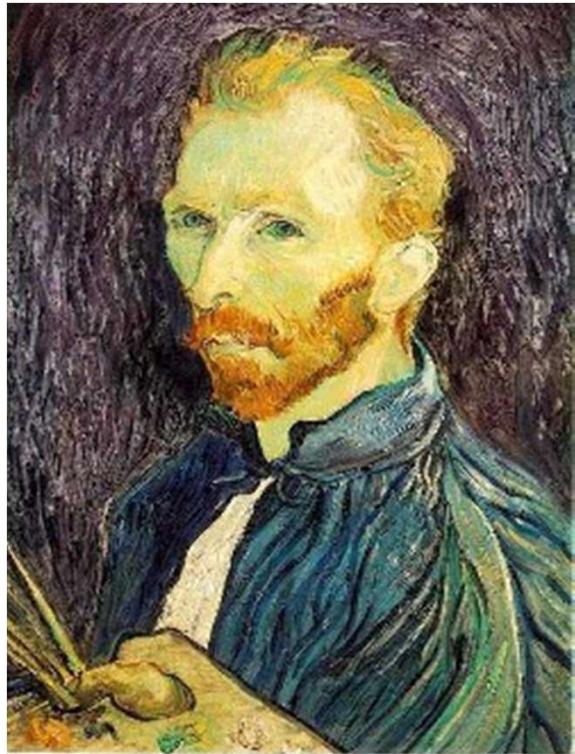
## Image down-sampling



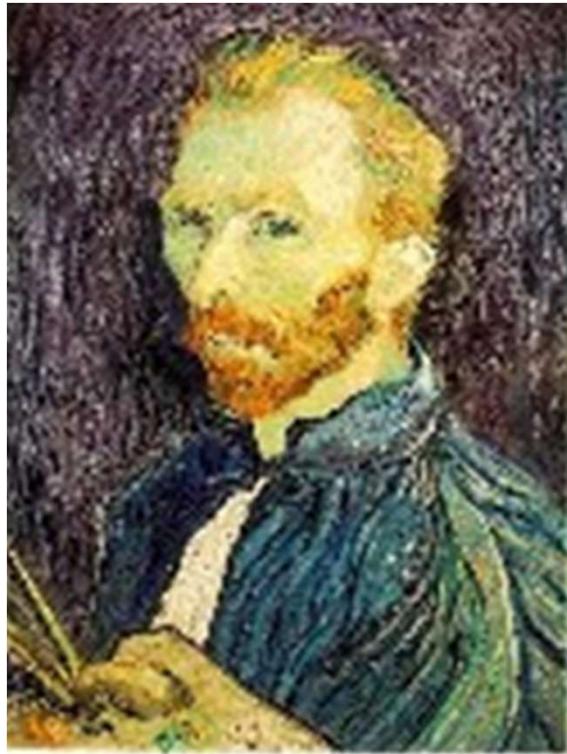
Throw away every other row and column  
to create a 1/2 size image



## Image down-sampling



1/2

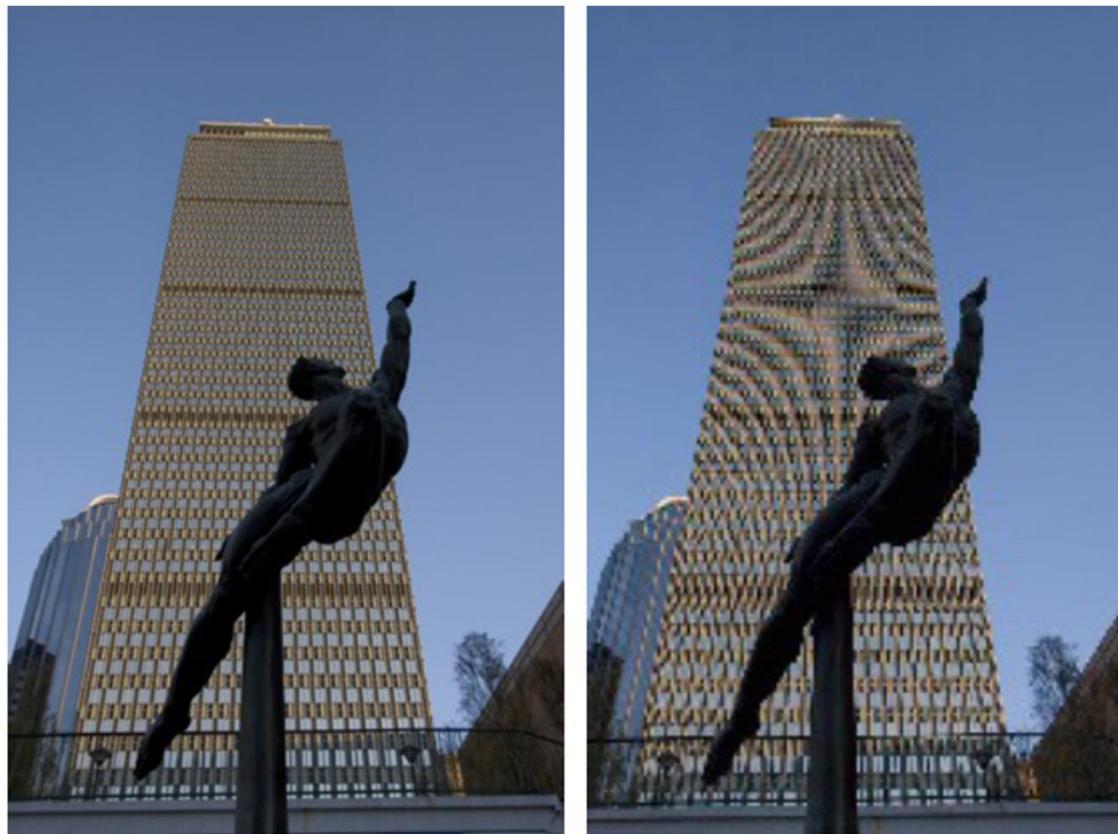


1/4 (2x zoom)



1/8 (4x zoom)

## Image down-sampling



Courtesy: F. Durand

## Image down-sampling



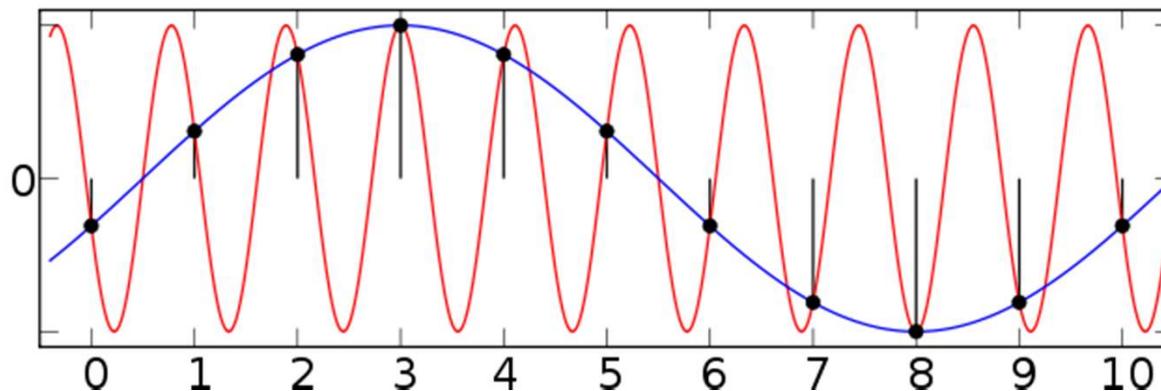
Original



1/2 (2x zoom)

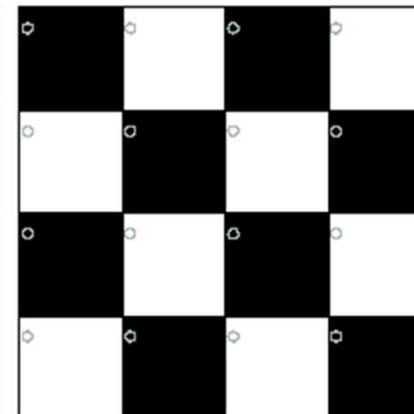
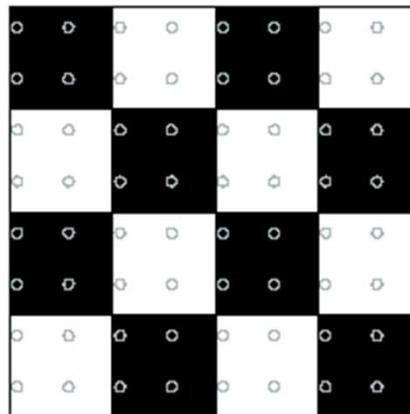
# Aliasing

- Sampling rate is not enough to capture the amount of detail
- To avoid aliasing
  - sampling rate  $\geq 2 * \text{max frequency}$  in the image (two samples per cycle)
  - minimum sampling rate is called Nyquist rate (on the basis of sampling theorem proposed by Harry Nyquist and Claude Shannon )

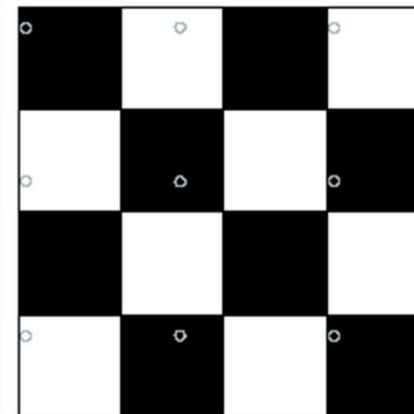
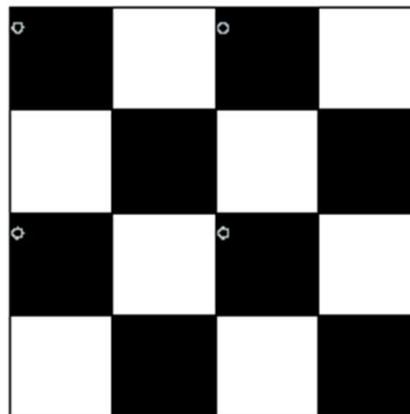


Courtesy: wikipedia

# Nyquist limit



Good sampling

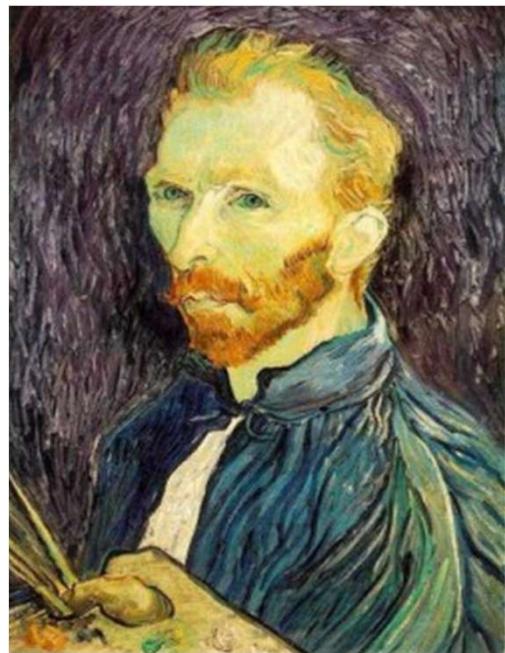


Bad sampling



# Gaussian pre-filtering

- Solution: filter the image, *then* subsample



Gaussian 1/2



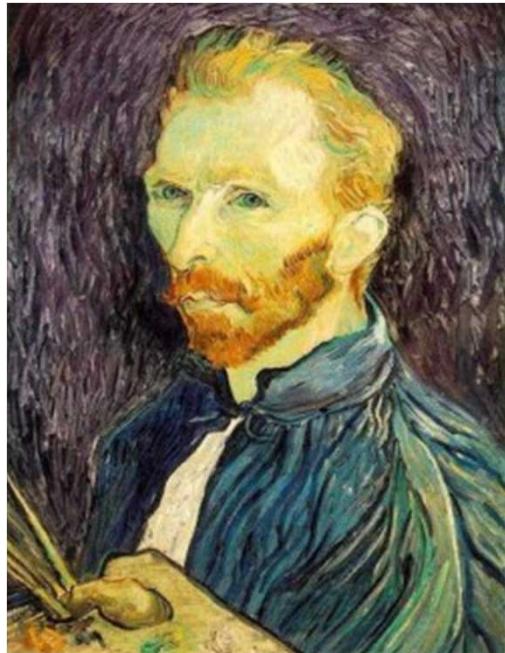
G 1/4



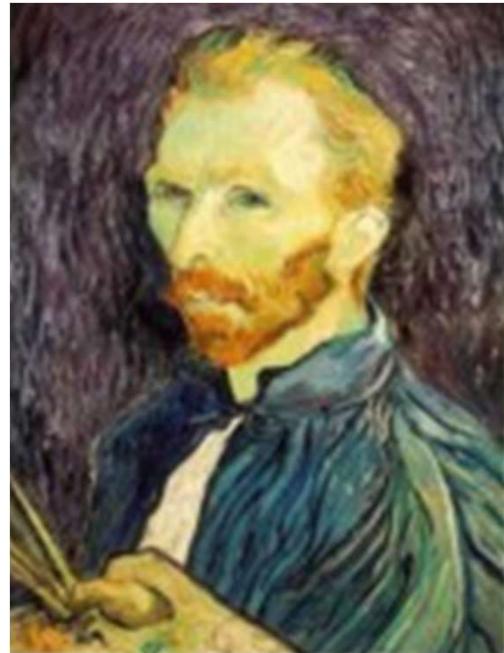
G 1/8

## Down-sampling with Gaussian pre-filtering

- Solution: filter the image, *then* subsample



Gaussian 1/2

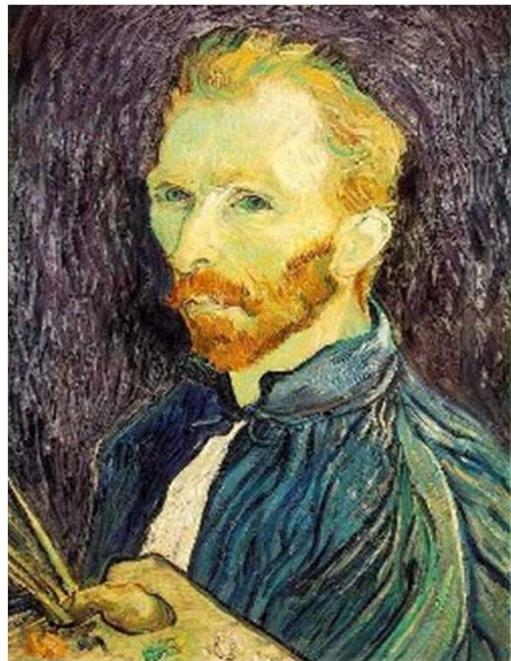


Gaussian 1/4 (2x zoom)

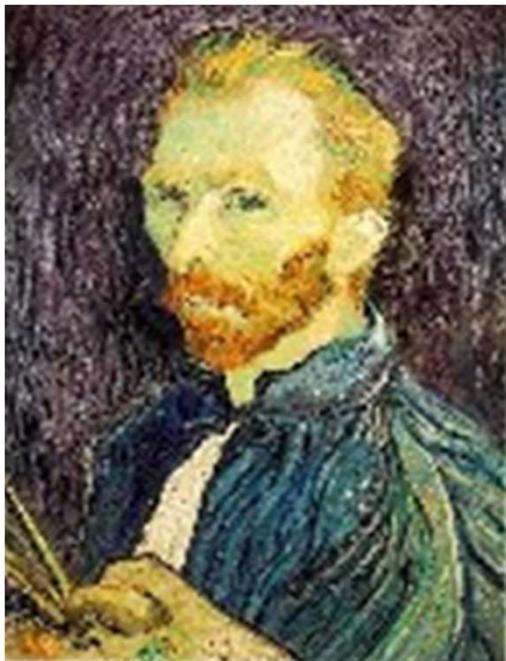


Gaussian 1/8 (4x zoom)

Compare with...



1/2



1/4 (2x zoom)



1/8 (4x zoom)

## Down-sampling with Gaussian pre-filtering

Smoothing removes high frequency components!



Gaussian  
Pyramid



## Gaussian pyramid



512

256

128

64

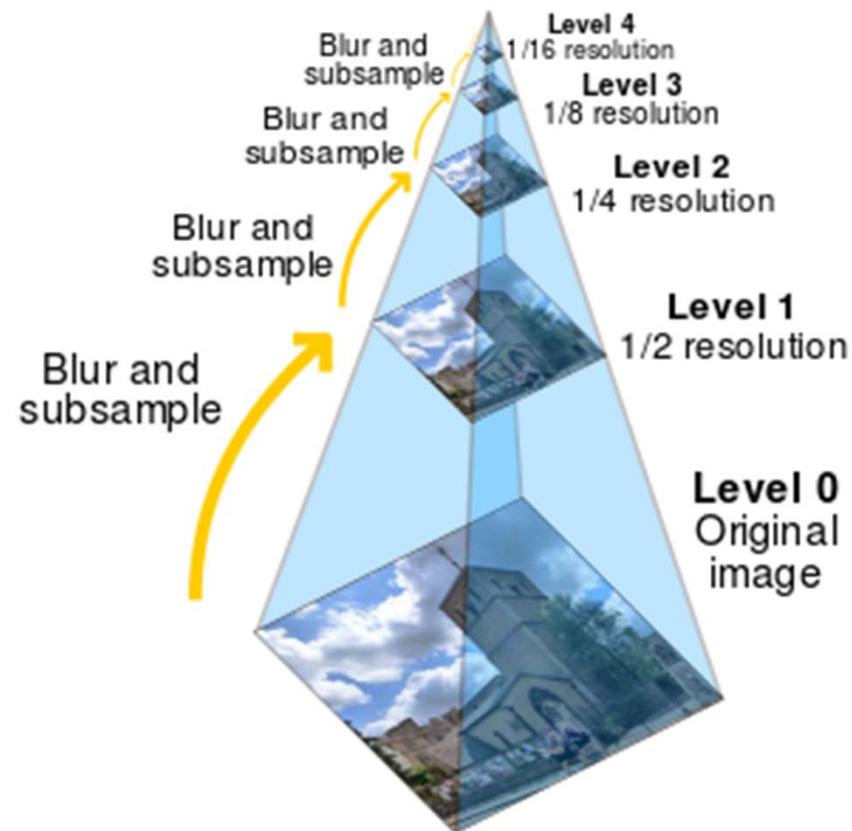
32

16

8



# Gaussian pyramid

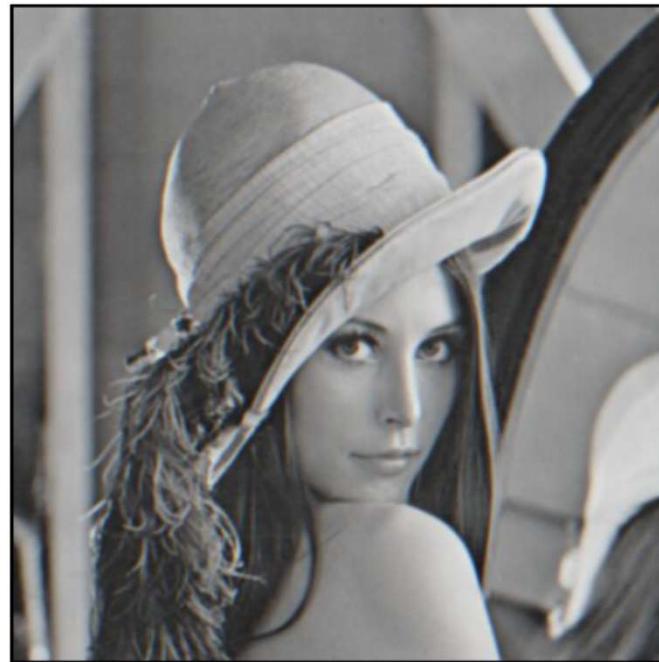


Courtesy: wikipedia

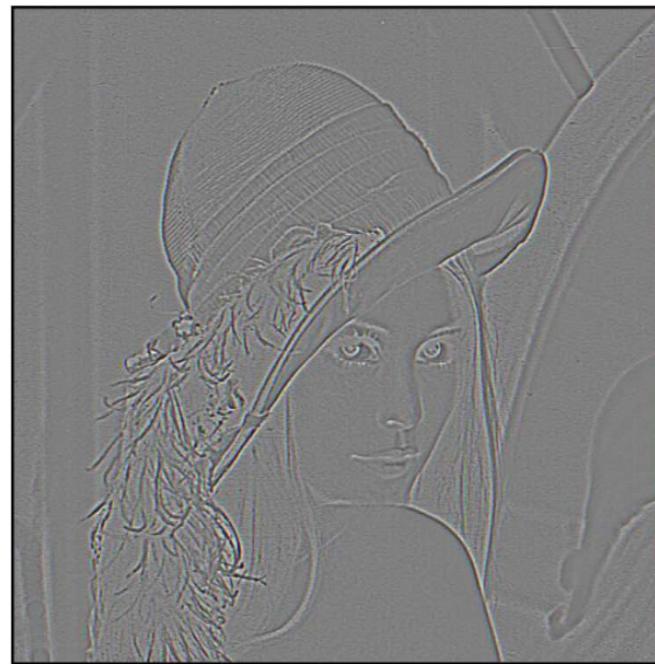
# What does smoothing takes away?



# What does smoothing takes away?

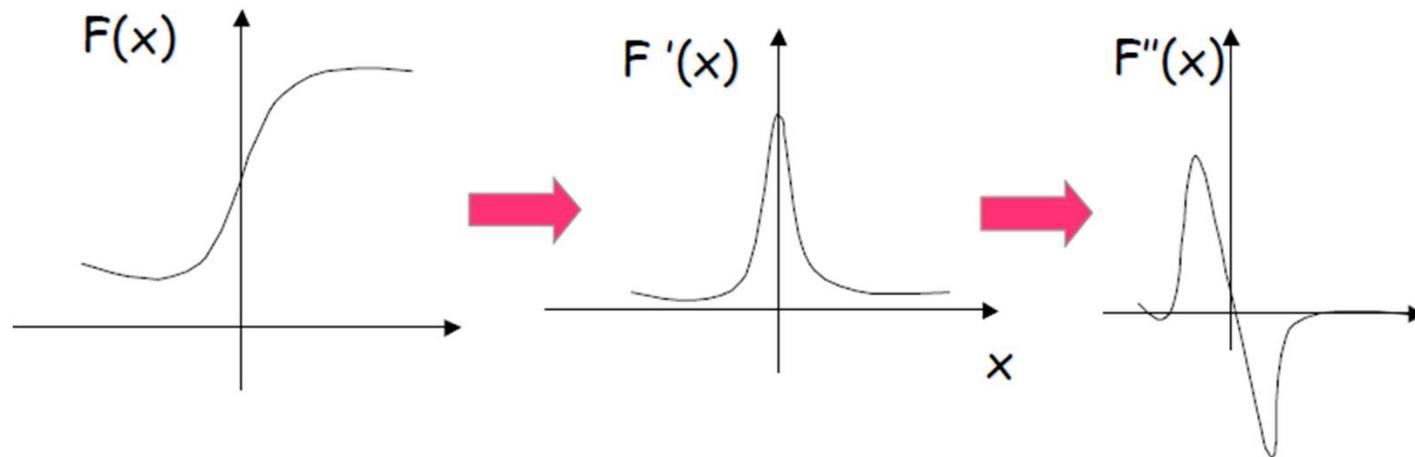


# What does smoothing takes away?



# Laplacian from Difference of Gaussian (DoG)

- 1D Signal Edge Characterization

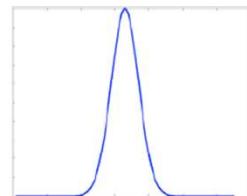


Source R. Collins

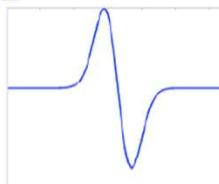
# Laplacian from Difference of Gaussian (DoG)

- 1D Gaussian and it's derivatives

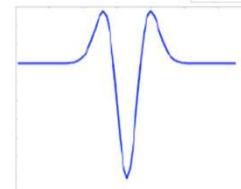
$$g(x) = e^{-\frac{x^2}{2\sigma^2}}$$



$$g'(x) = -\frac{1}{2\sigma^2} 2xe^{-\frac{x^2}{2\sigma^2}} = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$$



$$g''(x) = \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right) e^{-\frac{x^2}{2\sigma^2}}$$

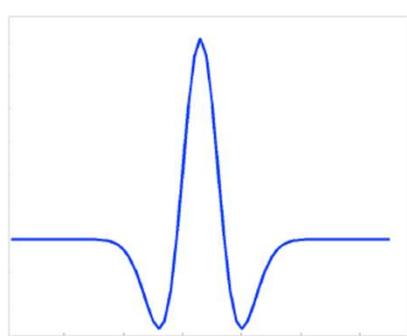


Source R. Collins

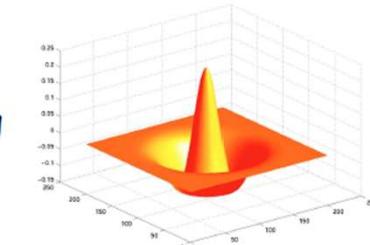
# Laplacian from Difference of Gaussian (DoG)

- Extension to 2D Laplacian of Gaussian (LoG) Operator

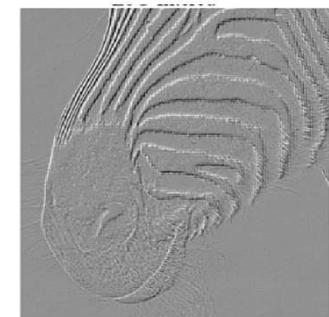
$$g''(x) = \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right)e^{-\frac{x^2}{2\sigma^2}}$$



2D  
analog



LoG "Mexican Hat"

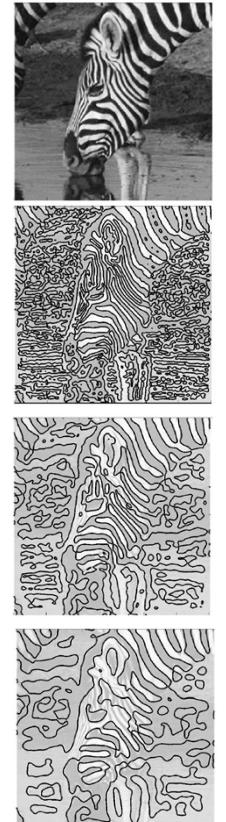
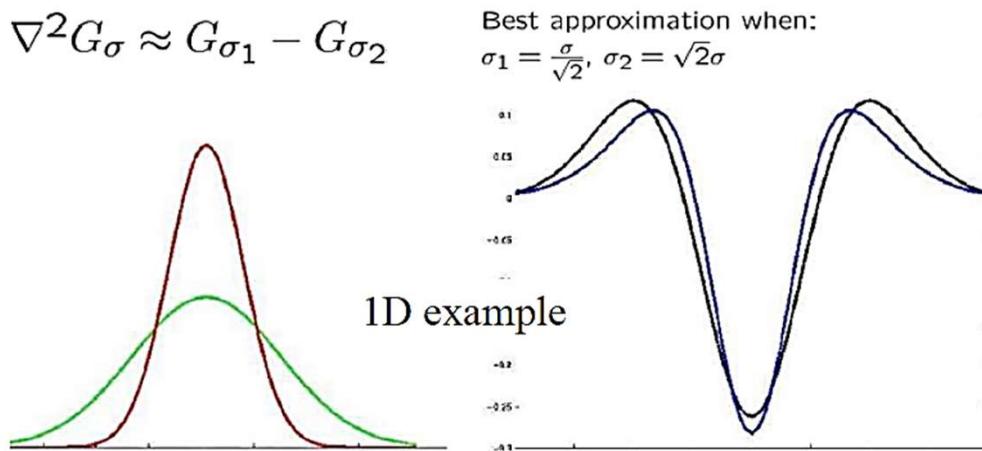


Band-Pass Filter (suppresses both high and low frequencies)

Source R. Collins

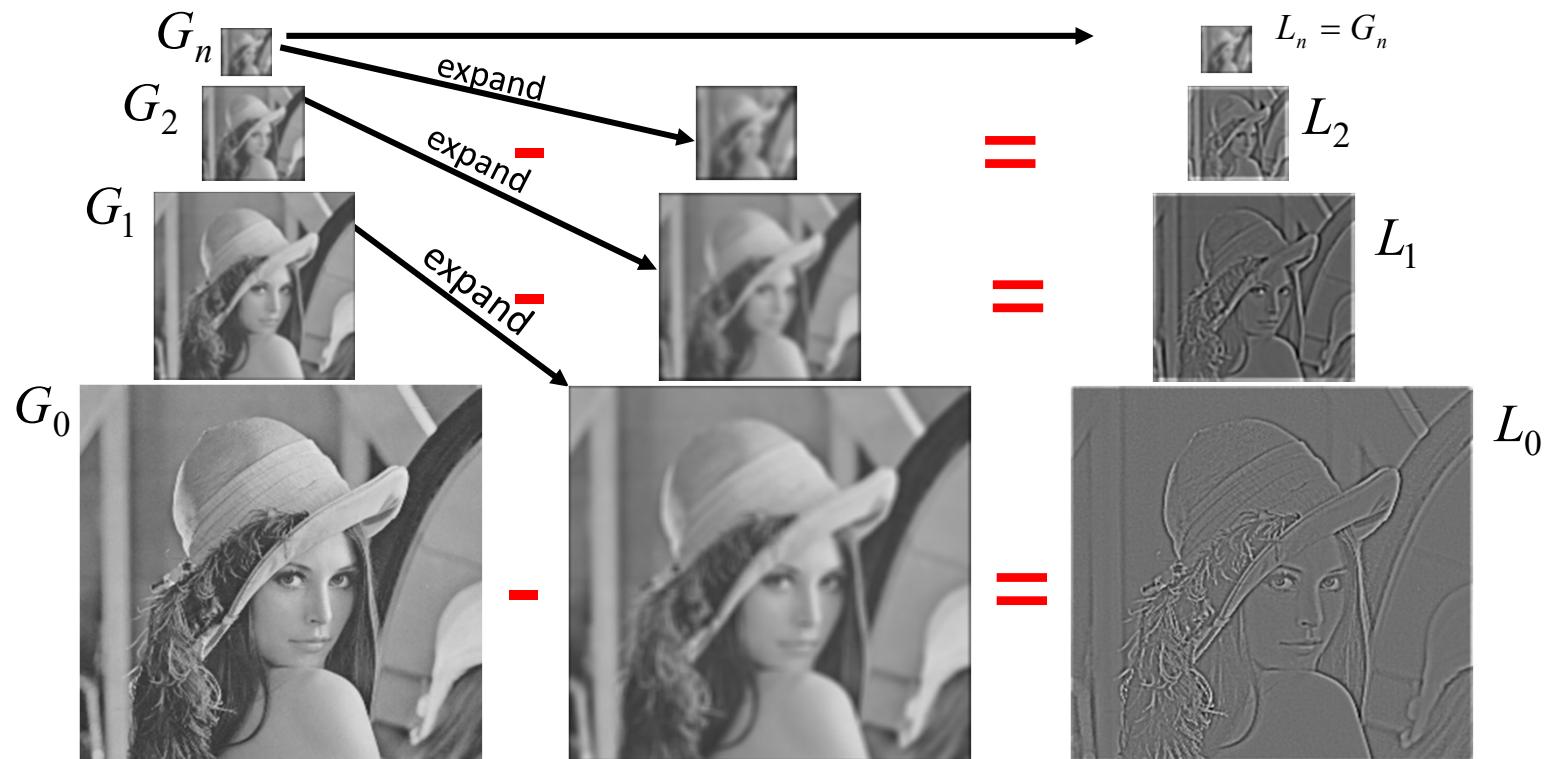
# Laplacian from Difference of Gaussian (DoG)

- LoG can be approximate by a Difference of two Gaussians (DoG) at different scales. It enables to capture **edges at different scales**.

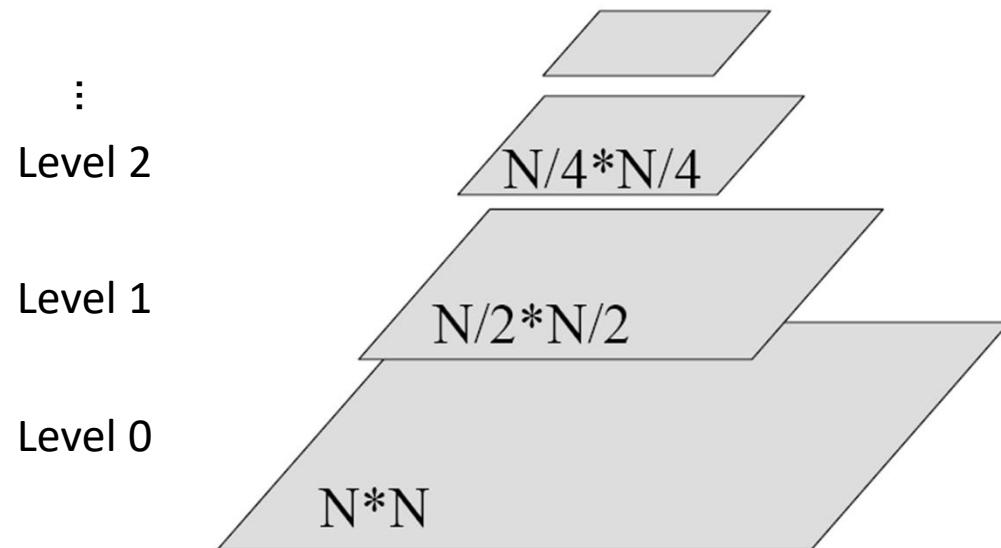


Source R. Collins

## Laplacian pyramid



## Space required for image pyramid



$$N^2 + \frac{1}{4}N^2 + \frac{1}{16}N^2 + \dots = 1\frac{1}{3}N^2$$



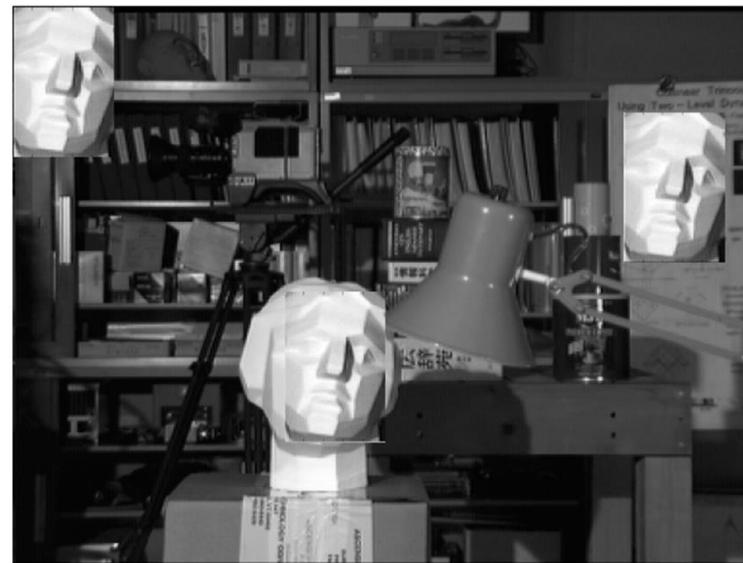
# Application Gaussian pyramid

- Efficient multi-scale detection

*Template*



*Search Region*



# Efficient multi-scale detection

*Template*



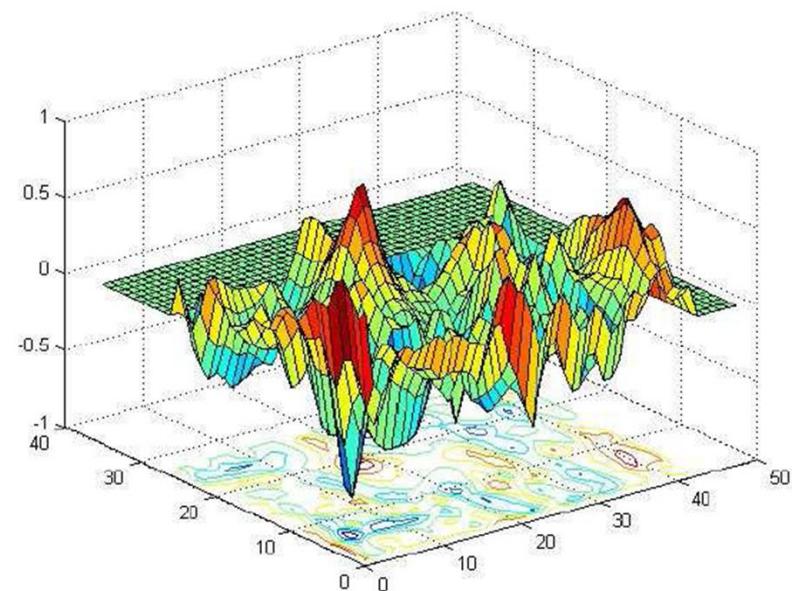
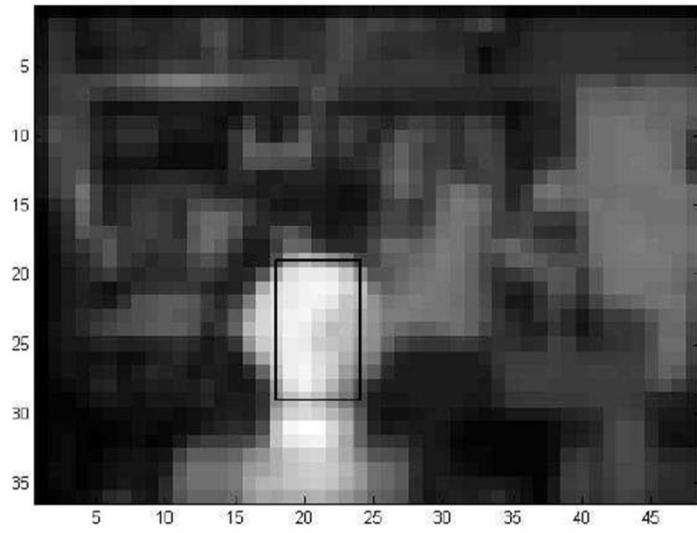
*Search Region*

Original Image



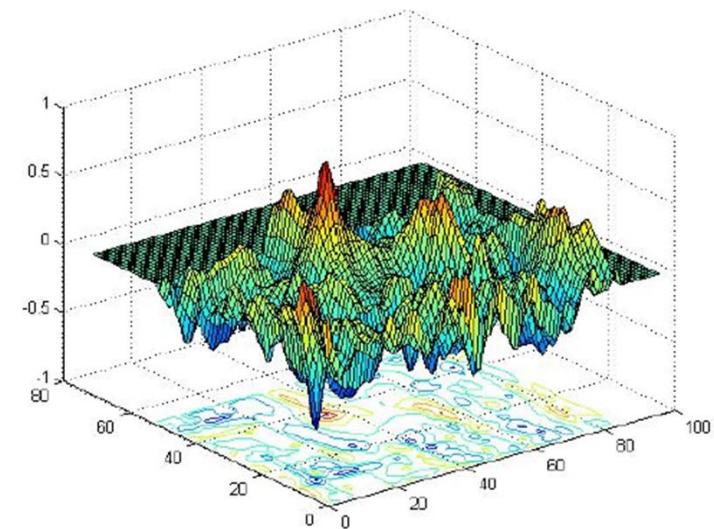
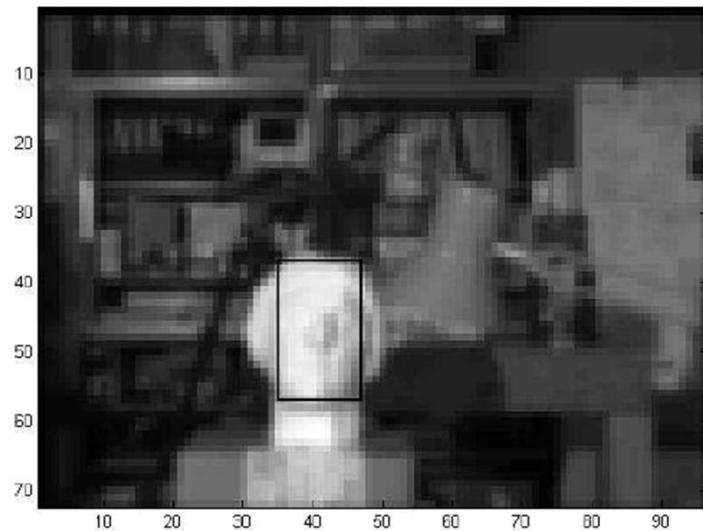
## Efficient multi-scale detection

- Level 3 search: at the lowest level we search the entire image with correlation template



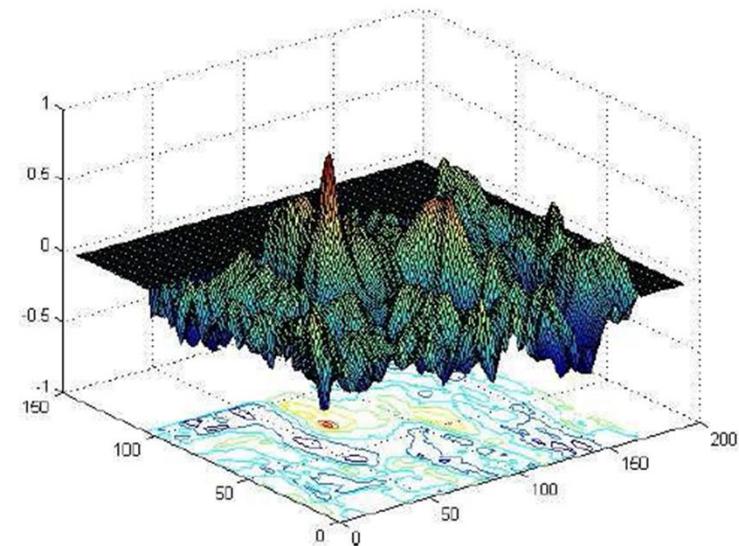
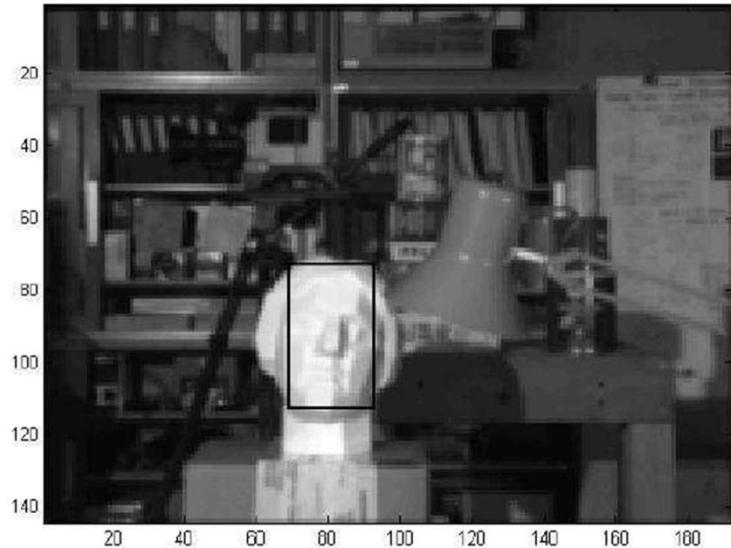
# Efficient multi-scale detection

- Level 2 search: constrained to a neighbourhood of high response centers in the previous level



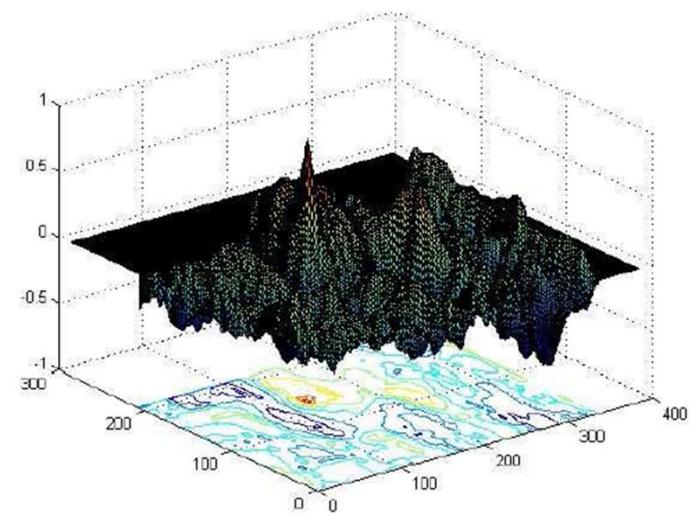
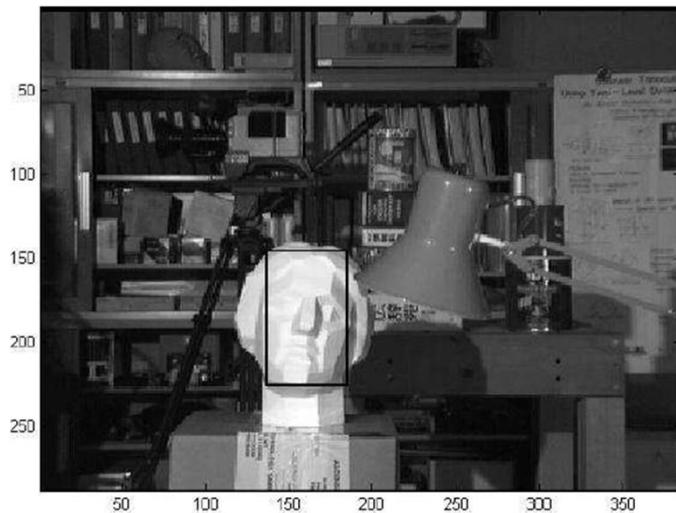
# Efficient multi-scale detection

- Level 1 search: again constrained based on results of level 1



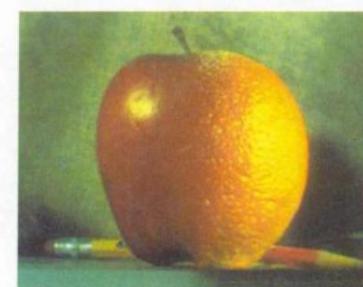
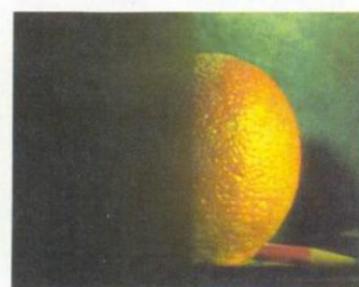
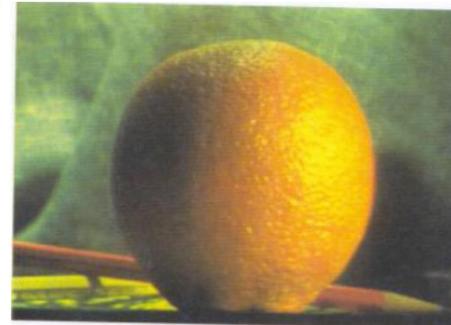
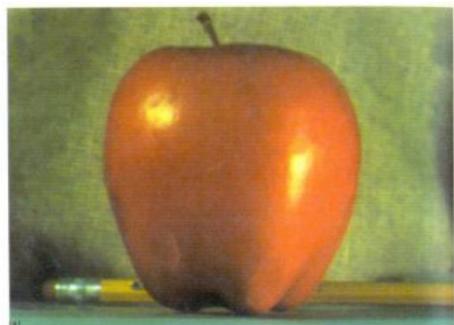
# Efficient multi-scale detection

- Level 0 search: total time reduced to 0.5 second from 31 seconds



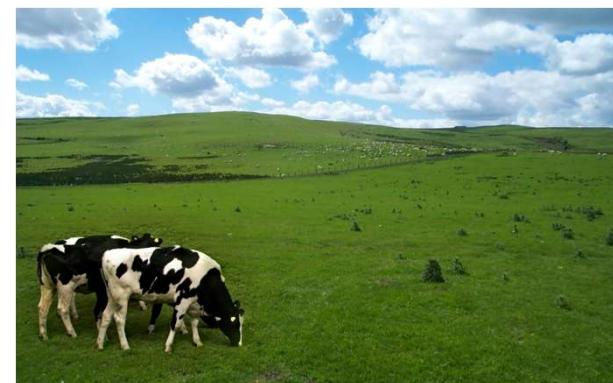
## Application Gaussian pyramid

- Blending Apples and Oranges



Burt and Adelson 1983

## Application Gaussian pyramid

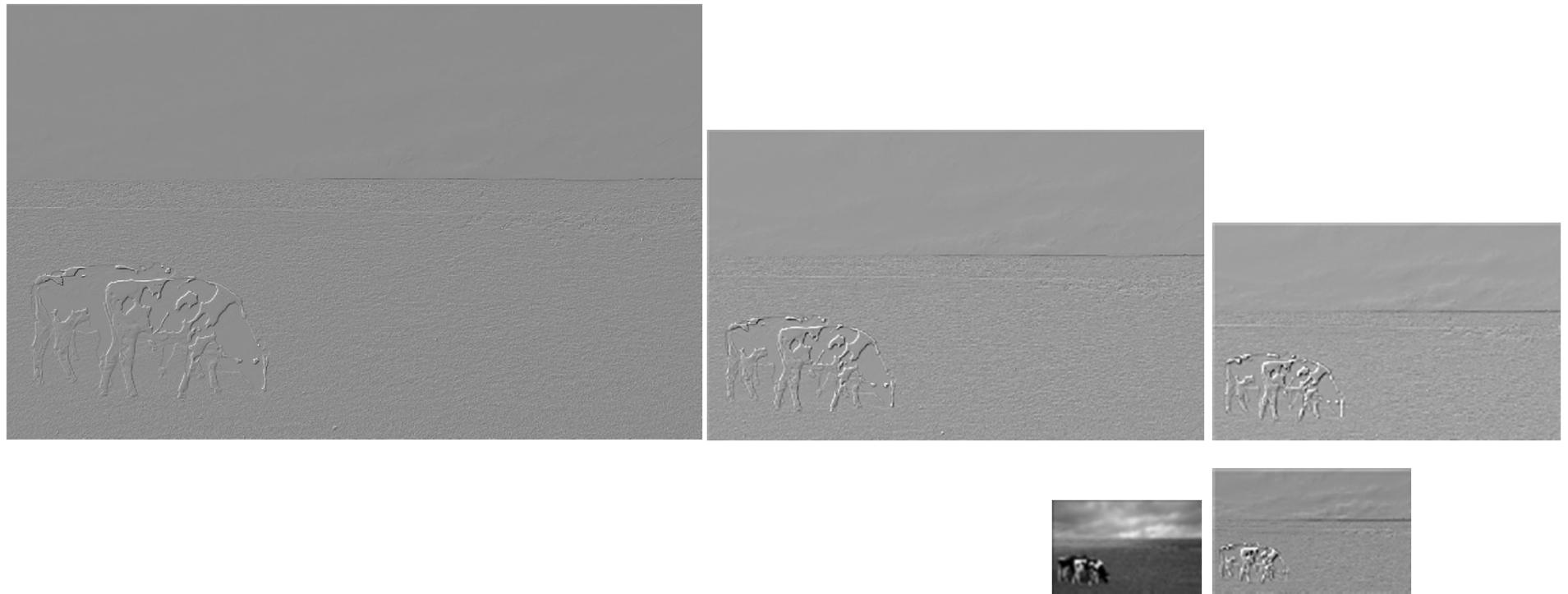


---

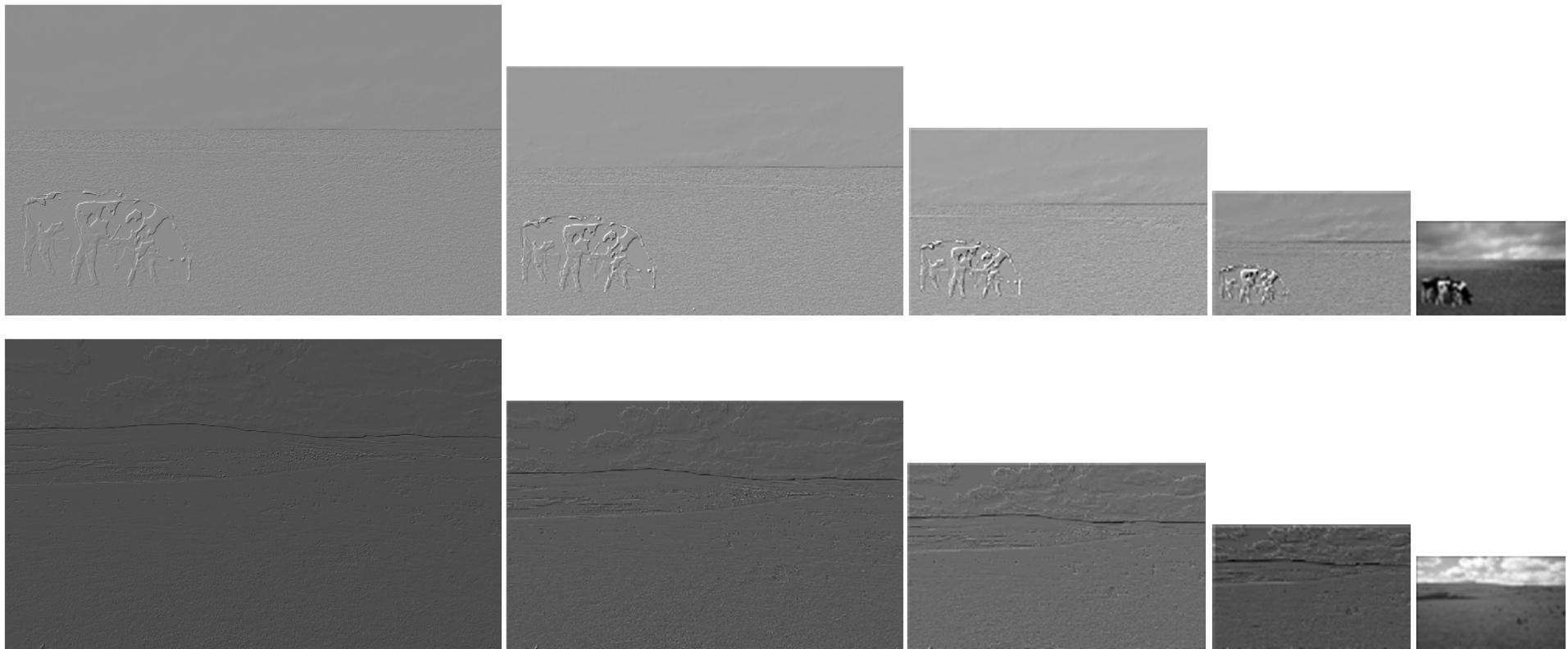




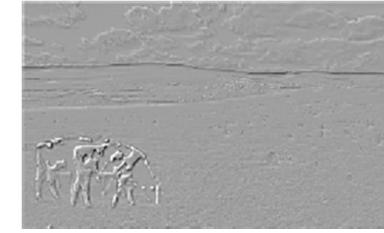
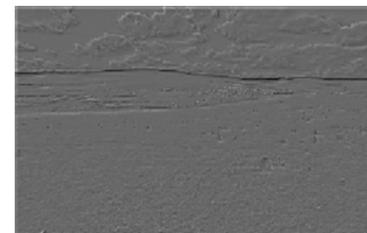
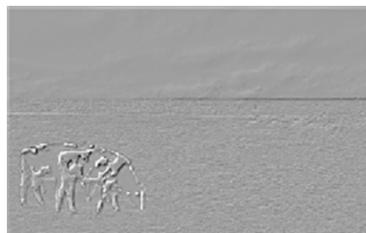
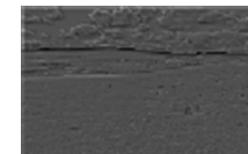
## Application Gaussian pyramid



# Application Gaussian pyramid



## Application Gaussian pyramid

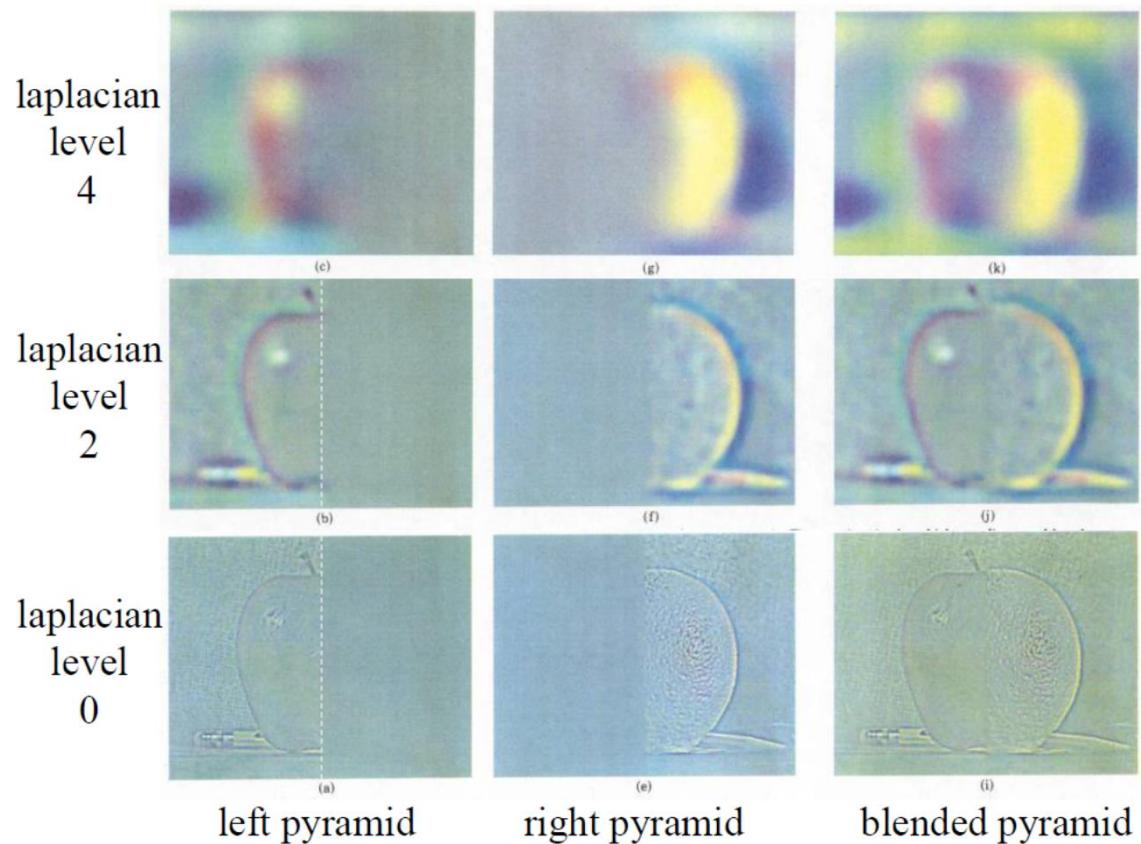


•  
•  
•  
•  
•

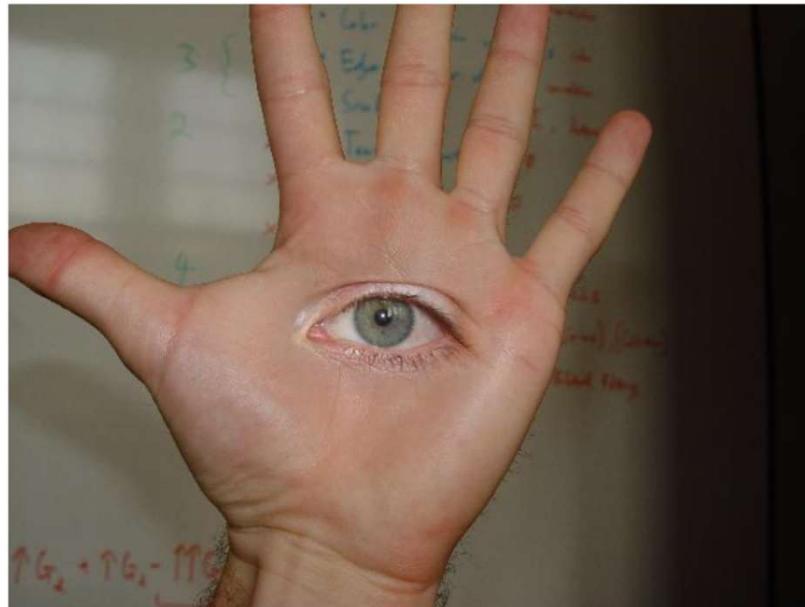




# Image Blending



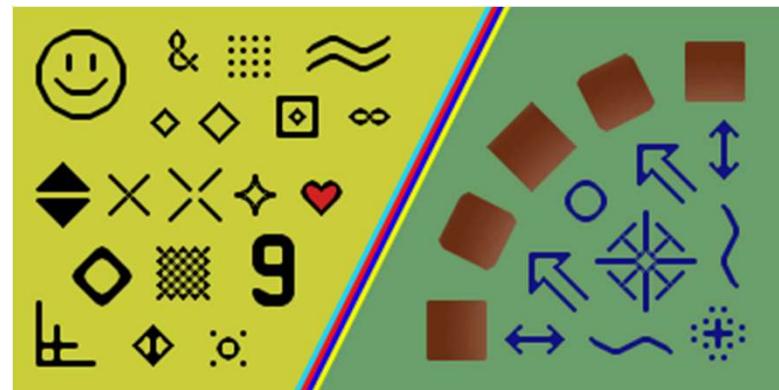
# Image Blending



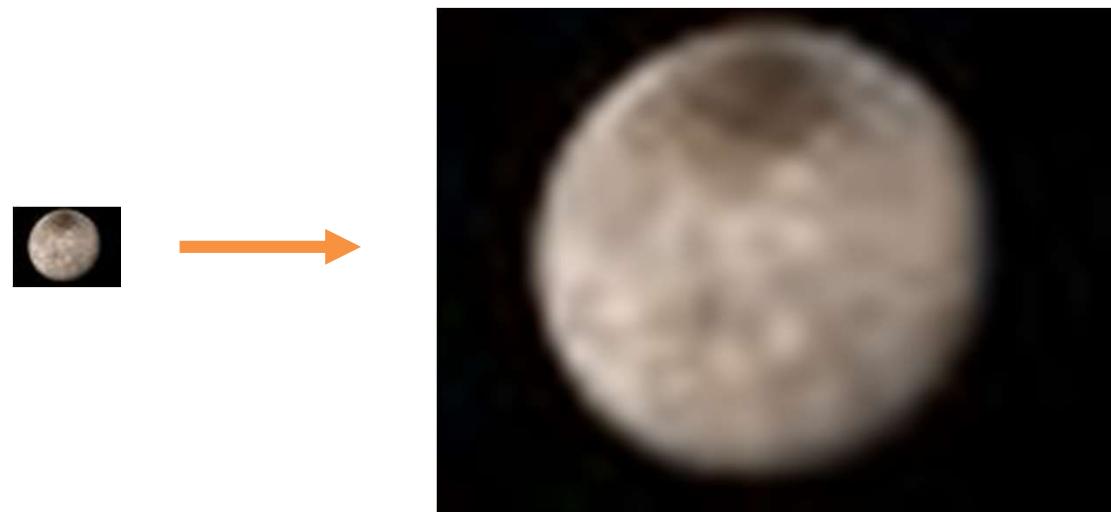
© prof. dmartin

---

# Questions?

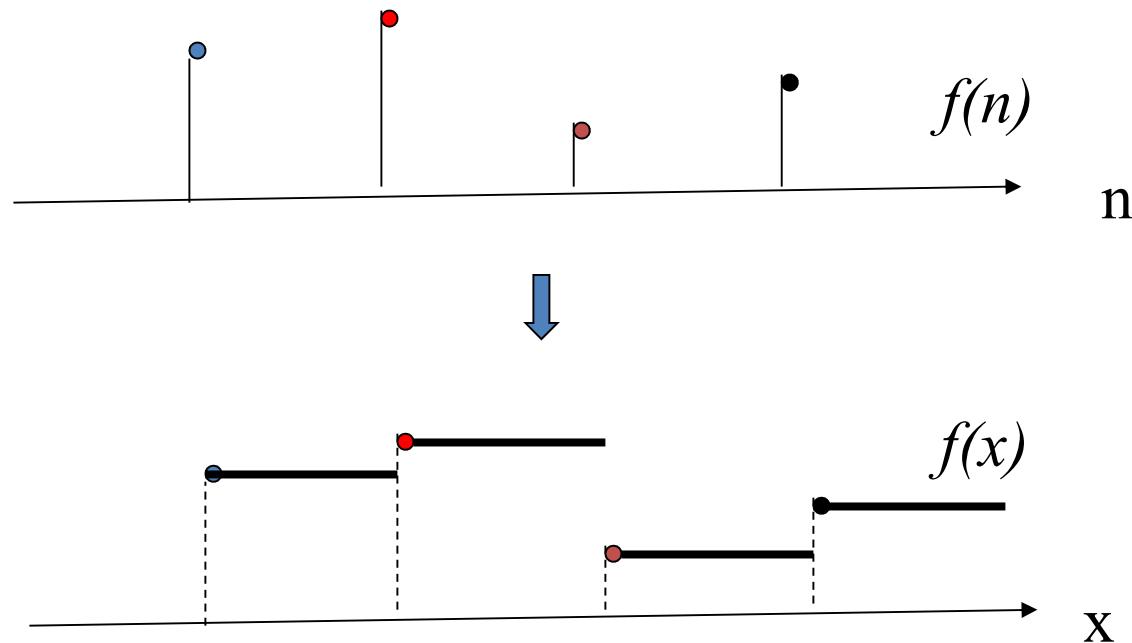


## Up-sampling



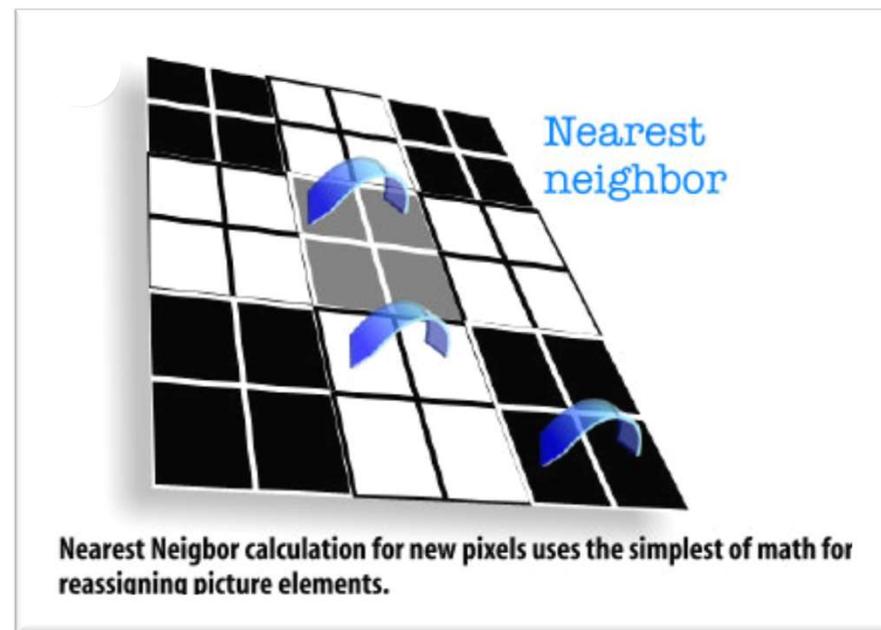
# Nearest neighbor interpolation

- Just repeat elements



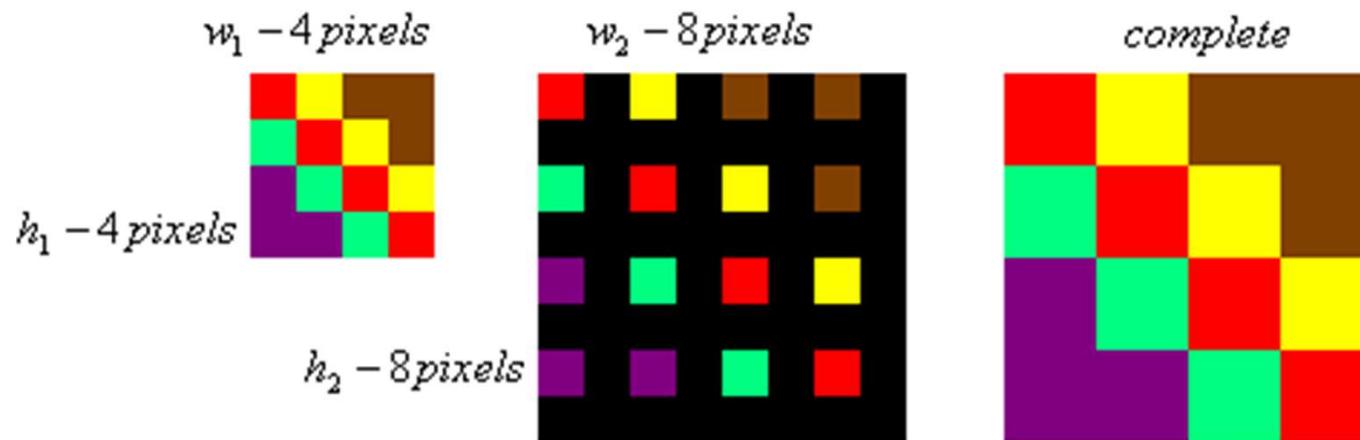
# Nearest Neighbour Interpolation

- Just repeat elements



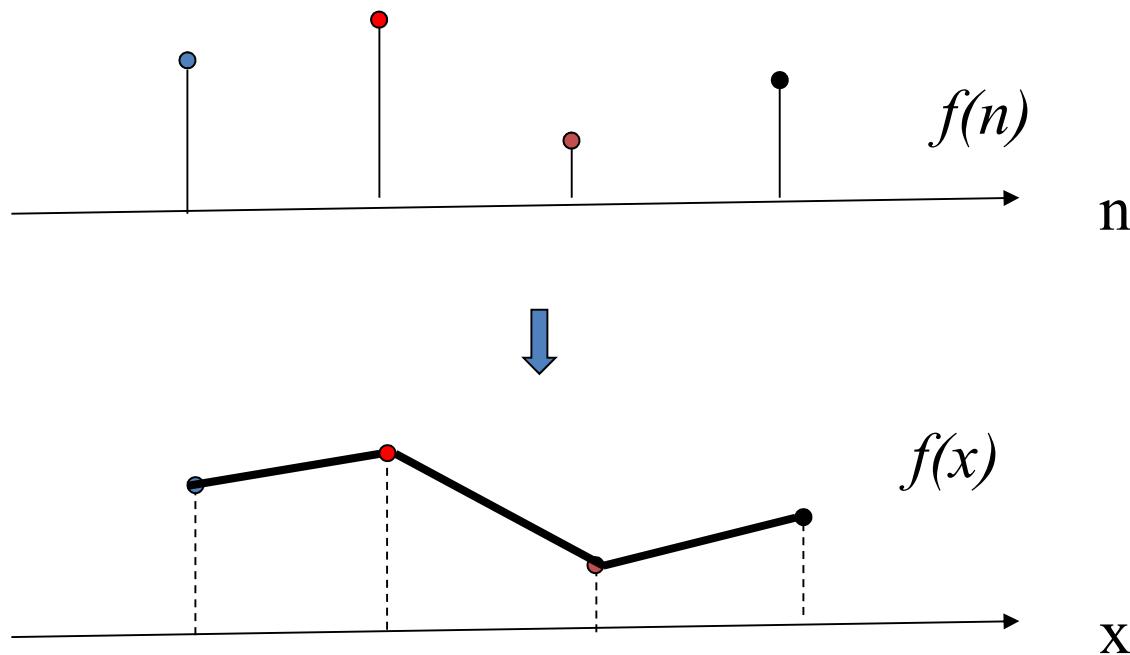
# Nearest Neighbour Interpolation

- Just repeat elements



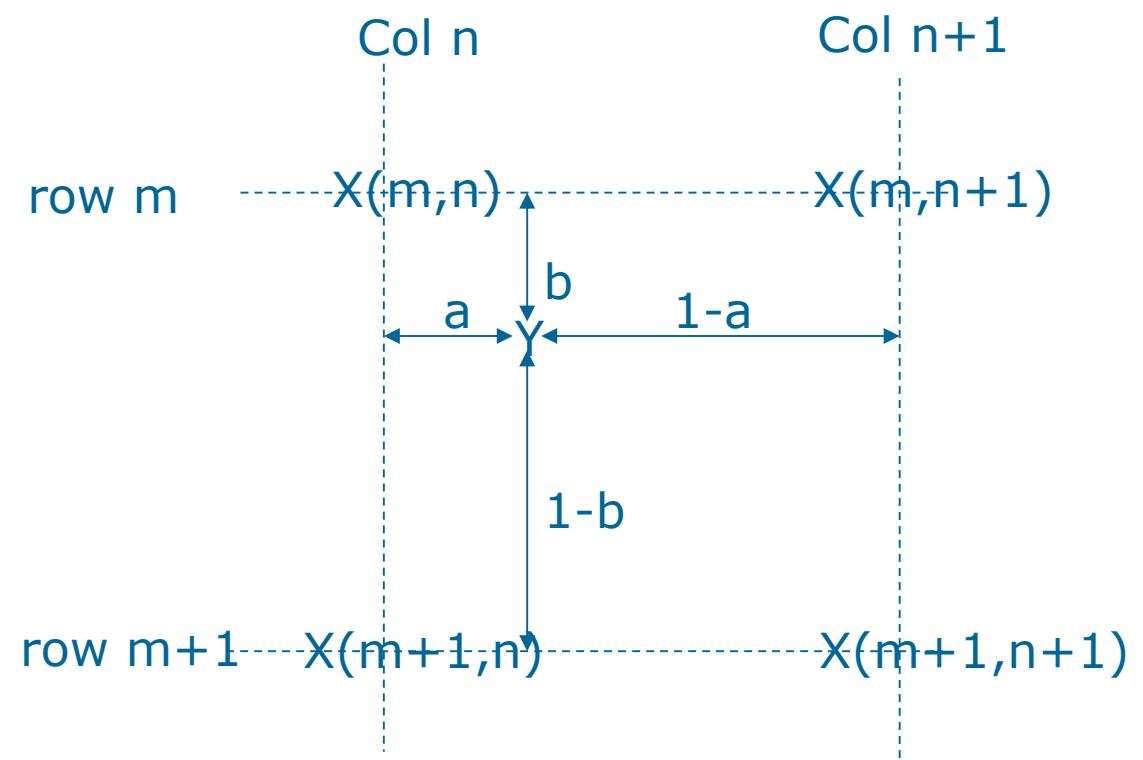
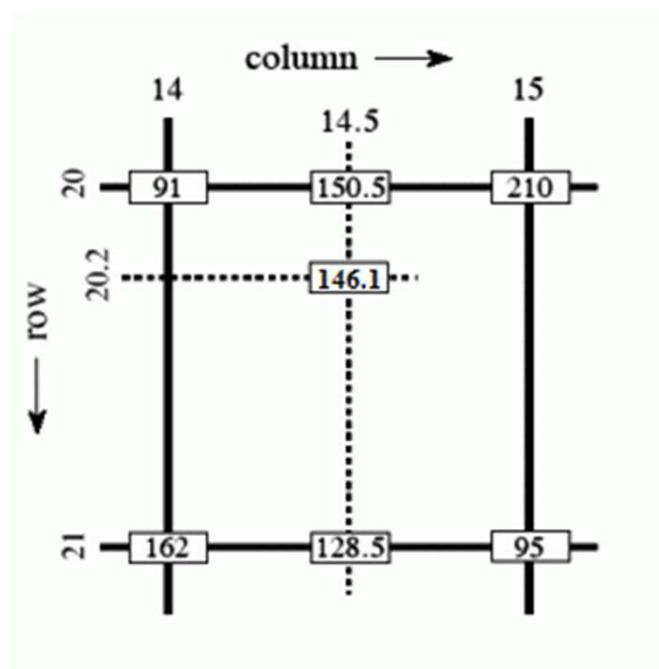
# Linear interpolation

- Linear combination



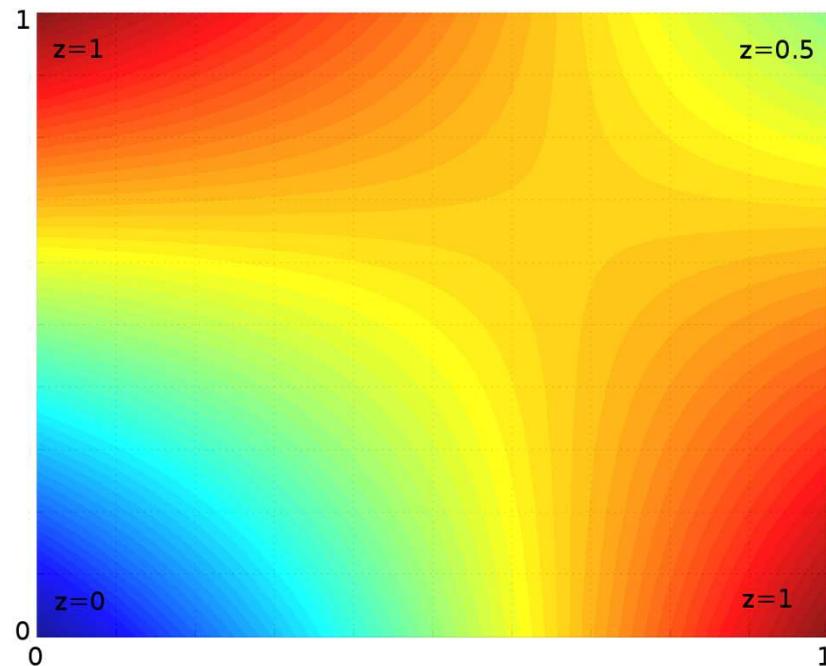
## Bilinear interpolation (2D)

- Divide and conquer



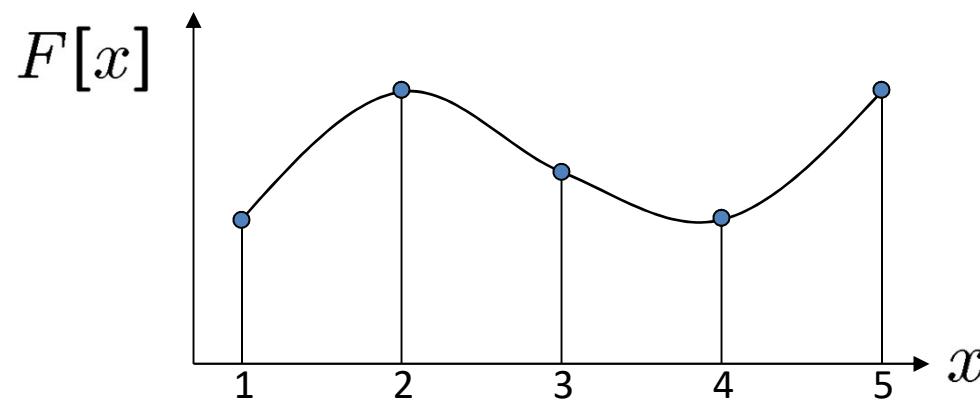
## Bilinear interpolation (2D)

- Although each step is linear in the sampled values and in the position, the interpolation as a whole is not linear but rather quadratic in the sample location.



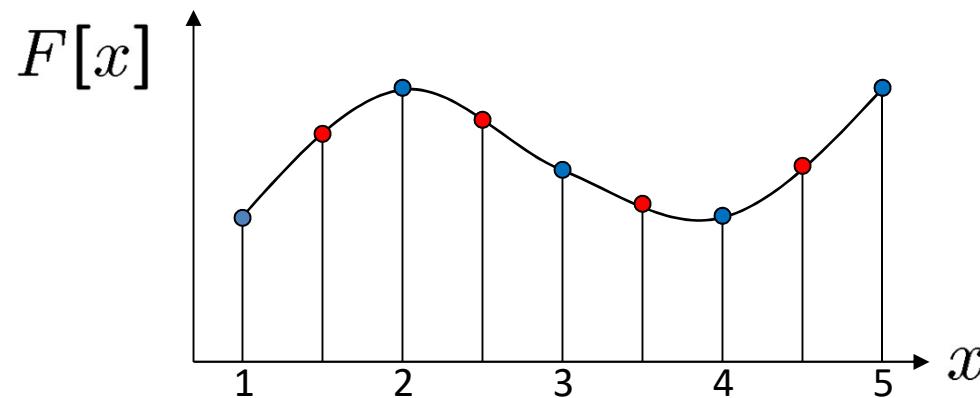
# Principled approach to interpolation

- Estimate the function from quantized values



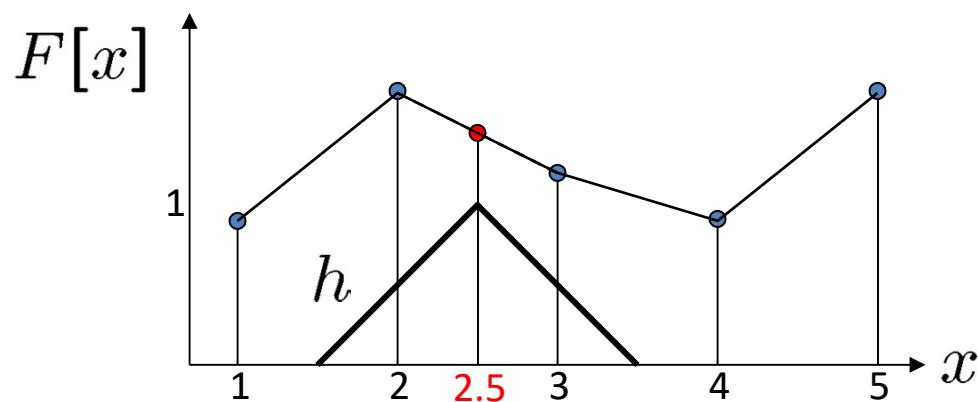
# Principled approach to interpolation

- Estimate the function from quantized values

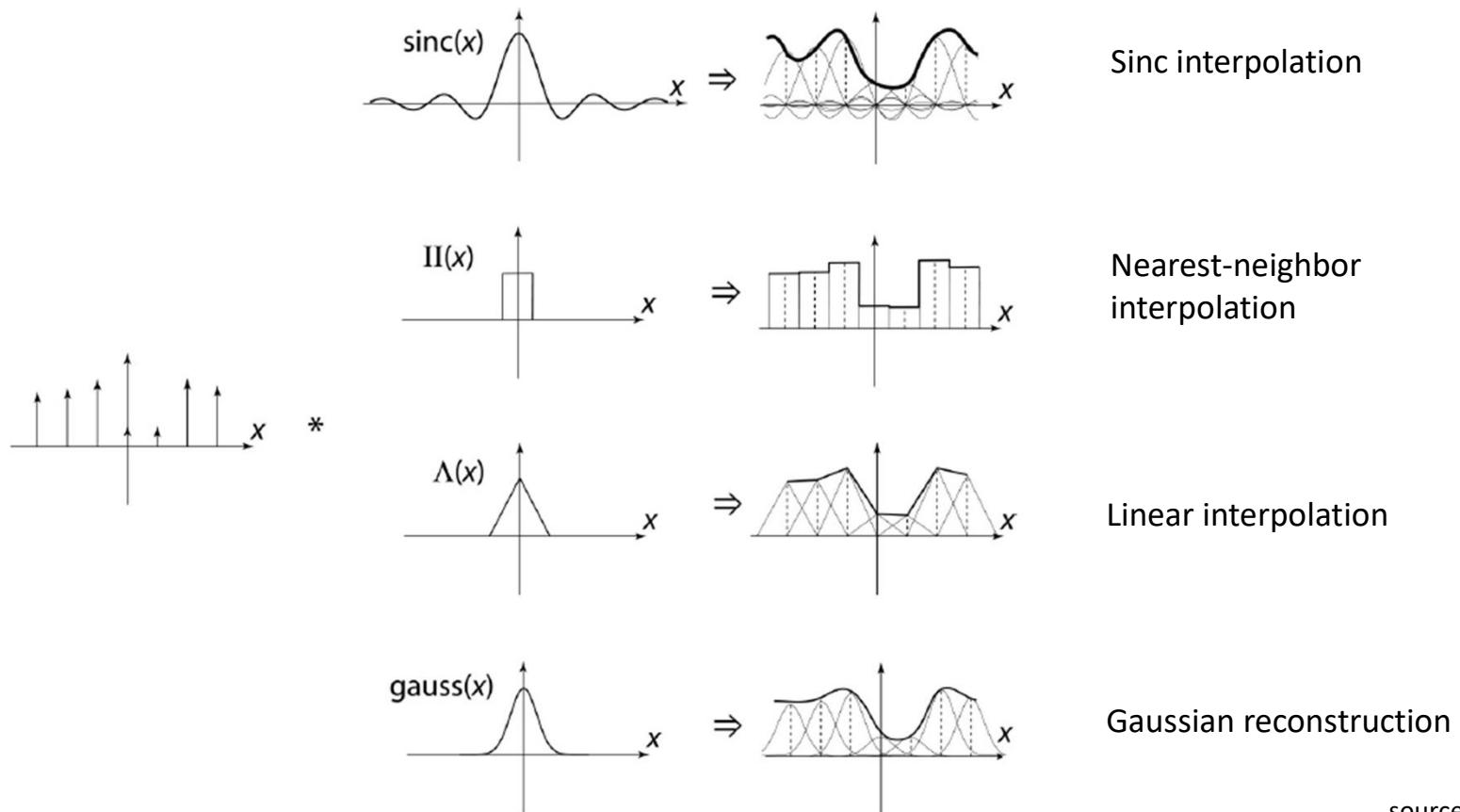


## Principled approach to interpolation

- Not always possible to estimate the function, what should we do?
- Approximation: Up-sampling as filtering

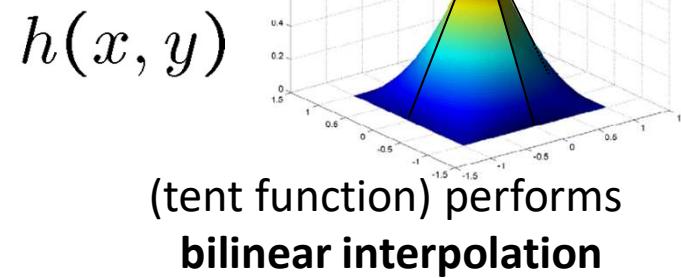
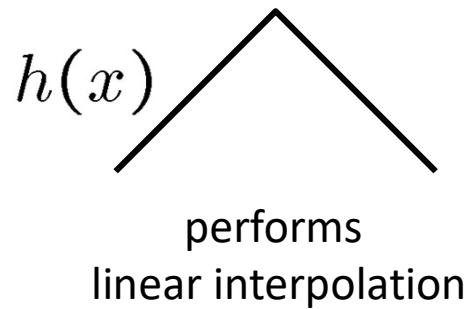


# Interpolation as filtering



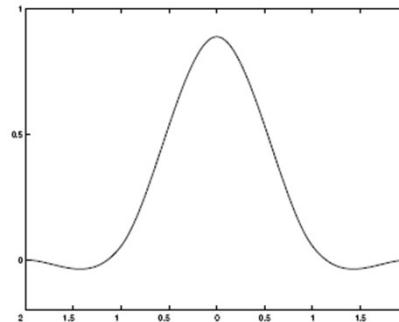
source: B. Curless

## From 1D to 2D

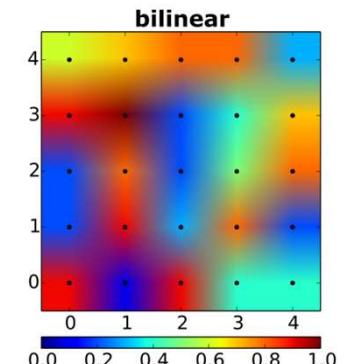
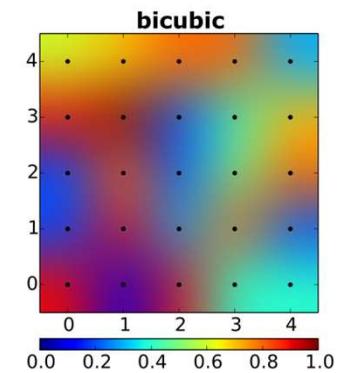


# Bicubic filter

- Commonly used



Cubic reconstruction filter



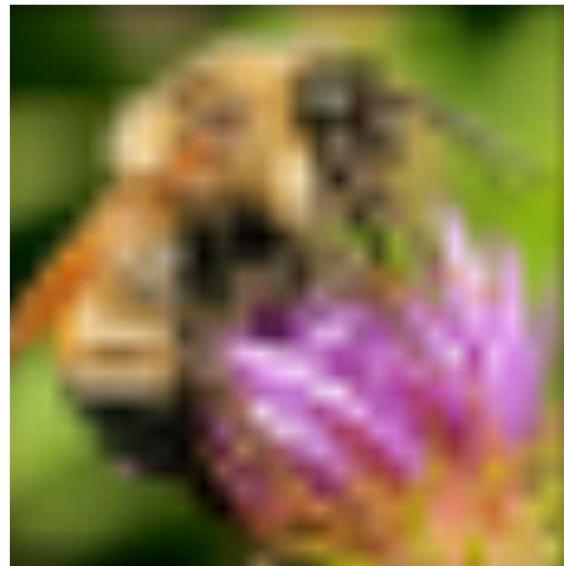
More advanced interpolation are adaptive, for example edge sensitive interpolation!

# Image interpolation

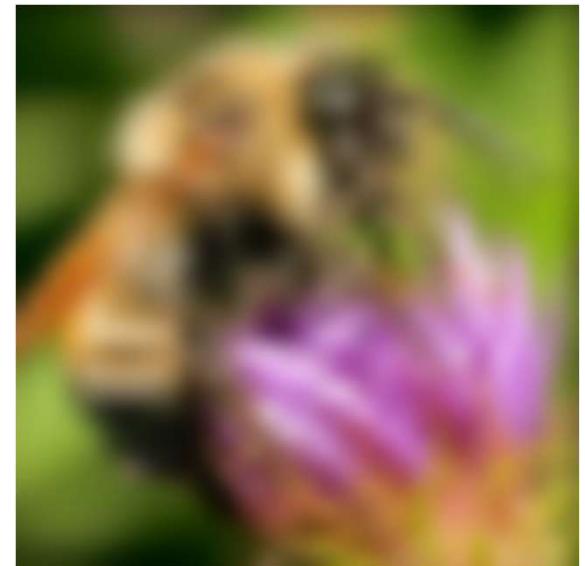
Original image:  x 10



Nearest-neighbor interpolation



Bilinear interpolation



Bicubic interpolation



# Image interpolation

Also used for *resampling*

