Digital Image Processing (CSE 478)
Lecture 6: Image resampling

Vineet Gandhi

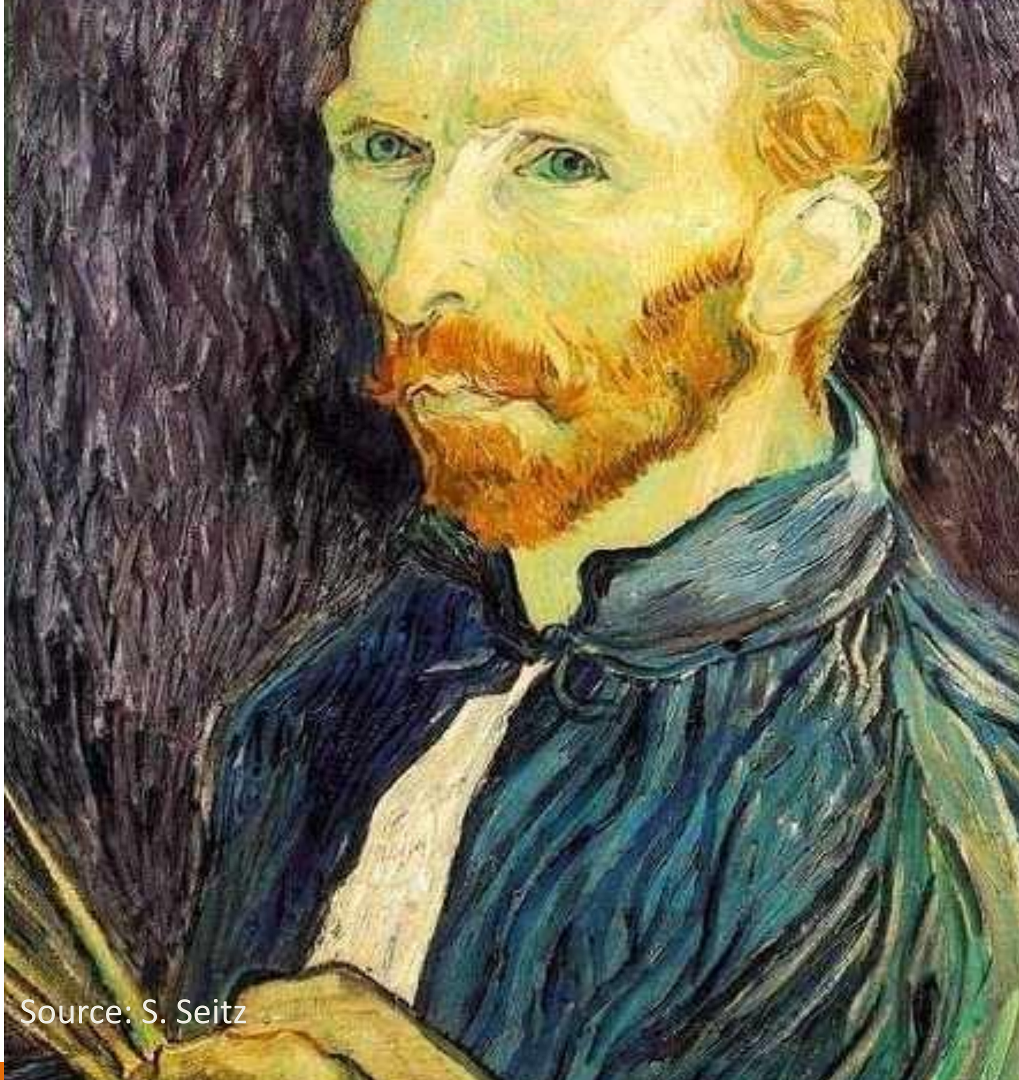
Center for Visual Information Technology (CVIT), IIIT Hyderabad

Many slides from S. Seitz, Noah Snavely, David Forsyth

Today's Class

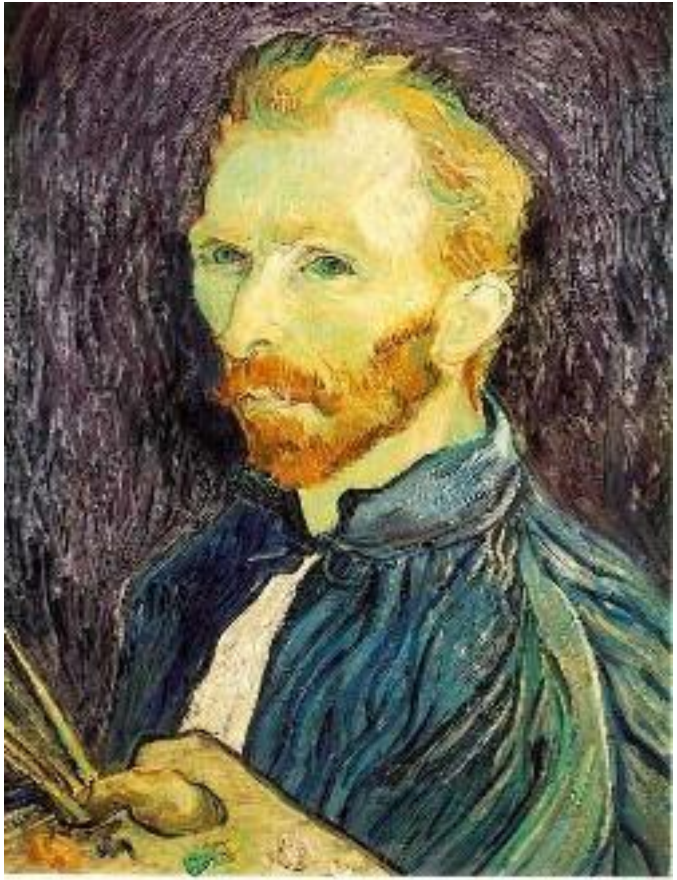
- Image down sampling
- Gaussian Pyramids
- Image up sampling

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

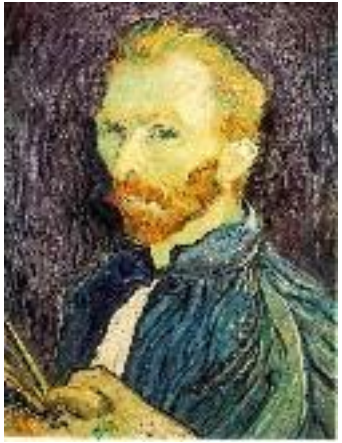


Source: S. Seitz

Image down-sampling



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



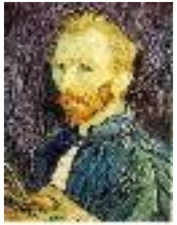
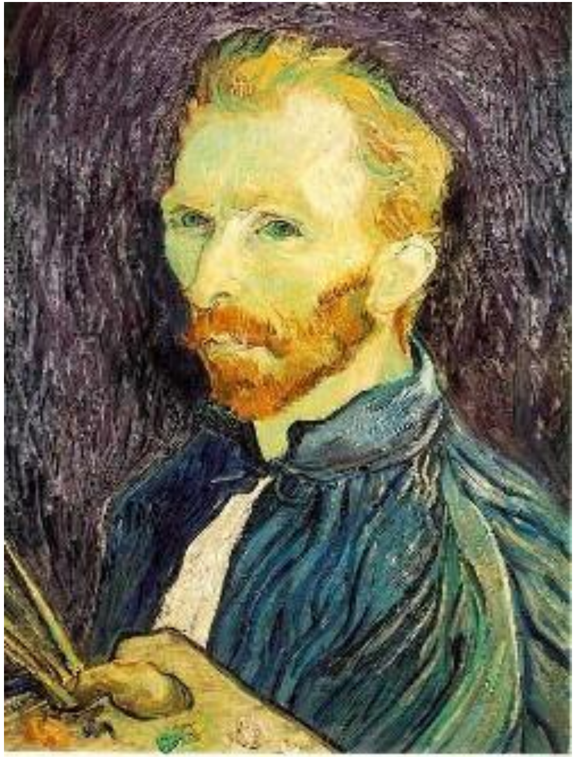
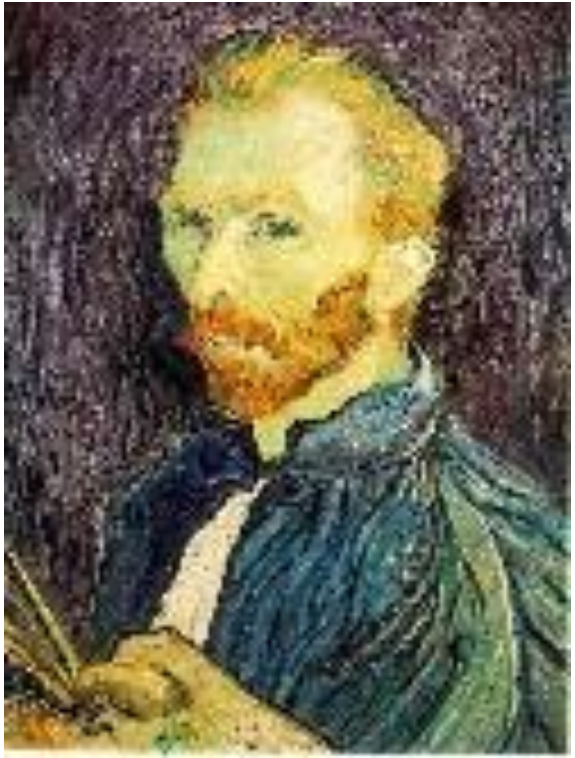


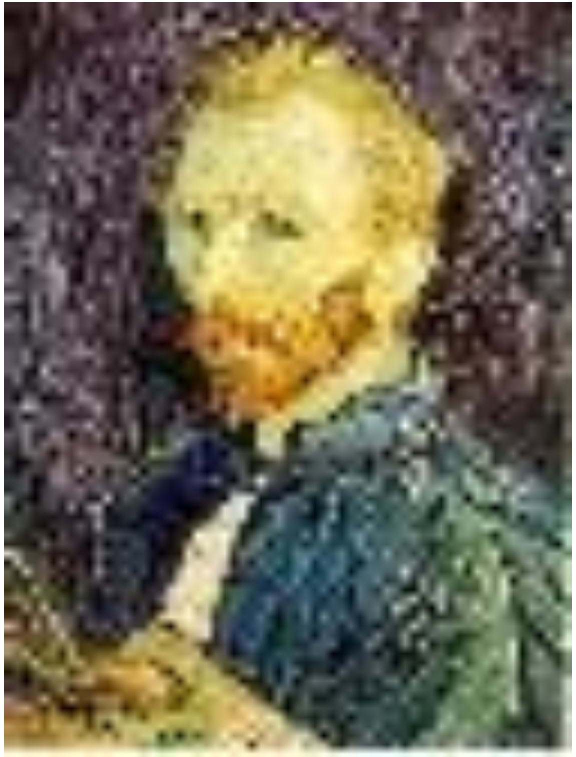
Image down-sampling



1/2

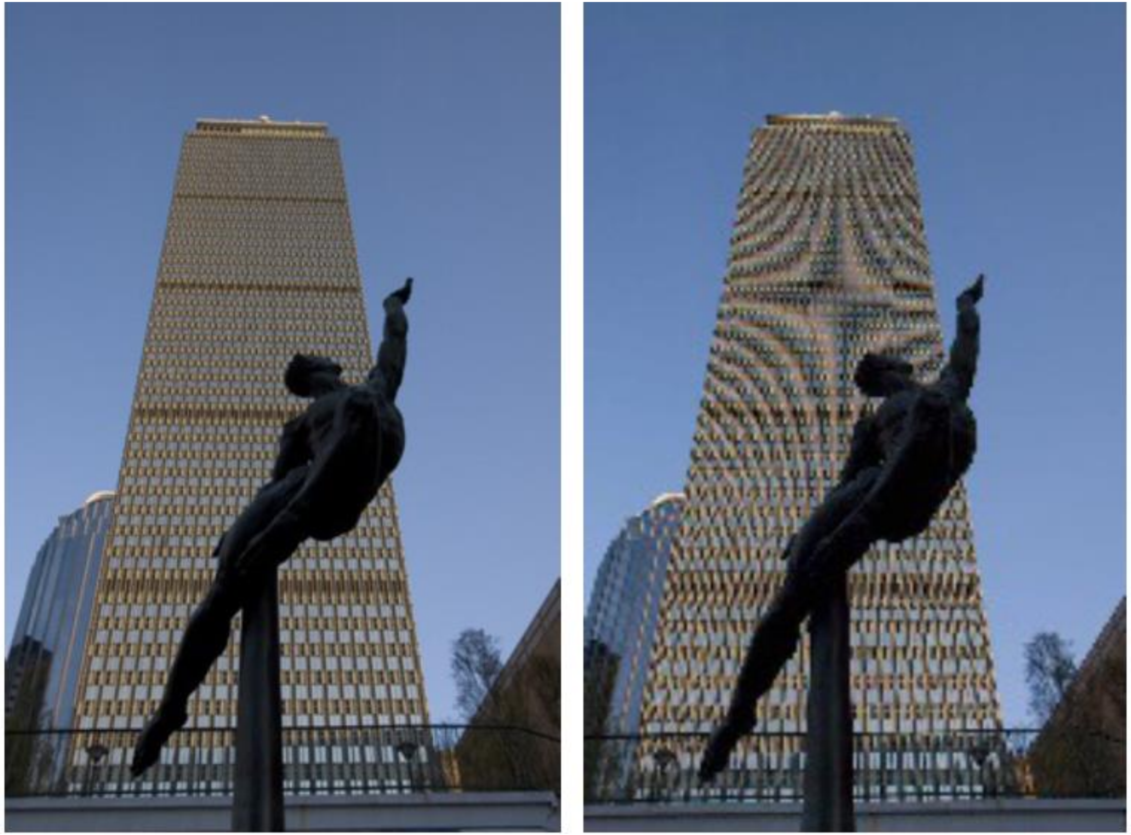


1/4 (2x zoom)



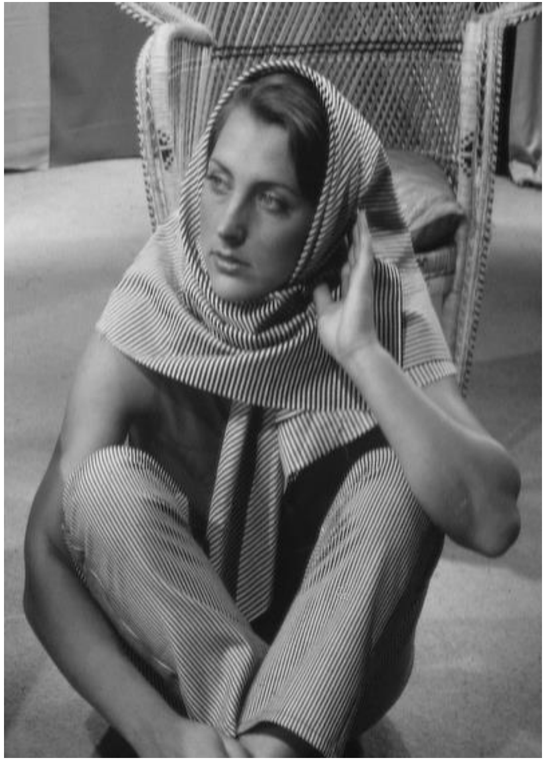
1/8 (4x zoom)

Image down-sampling

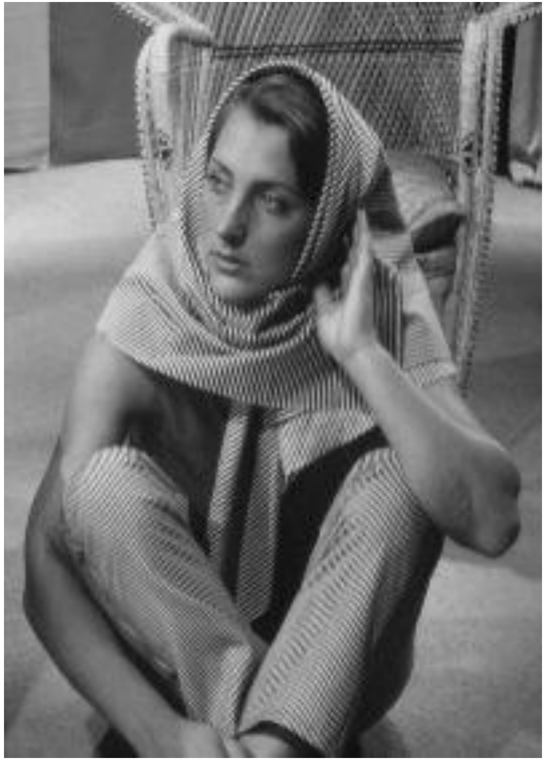


Courtesy: F. Durand

Image down-sampling



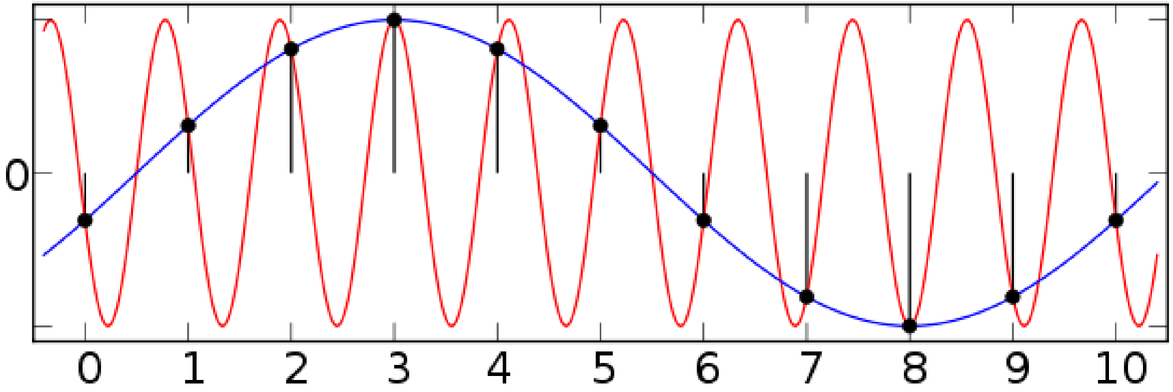
Original



1/2 (2x zoom)

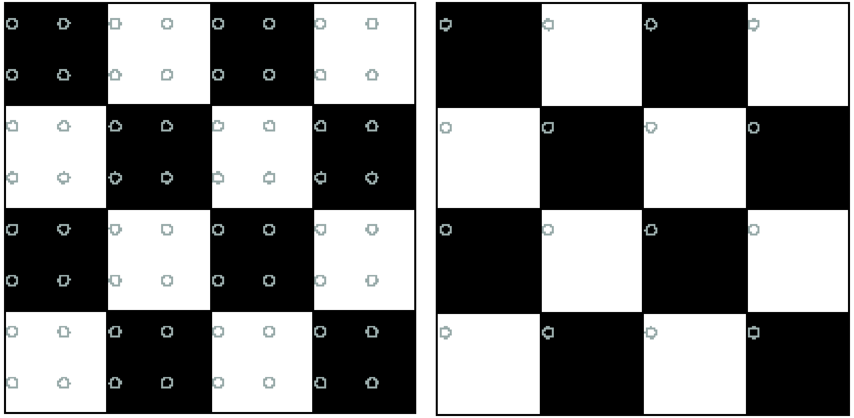
Aliasing

- Sampling rate is not enough to capture the amount of detail
- To avoid aliasing
  - sampling rate ≥ 2 * 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)

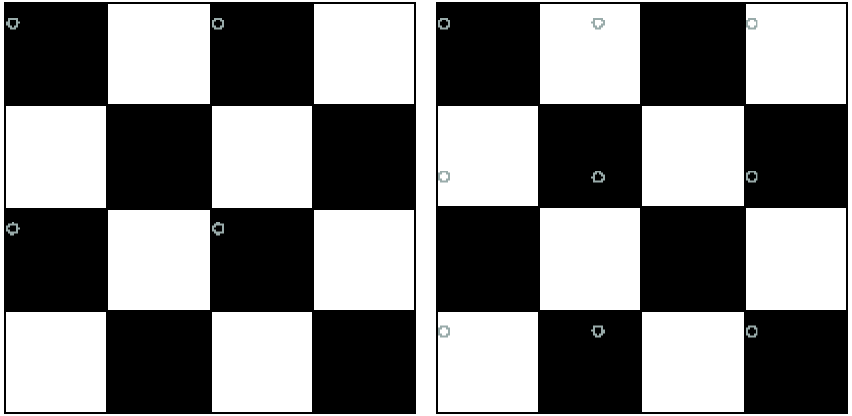


Courtesy: wikipedia

Nyquist limit



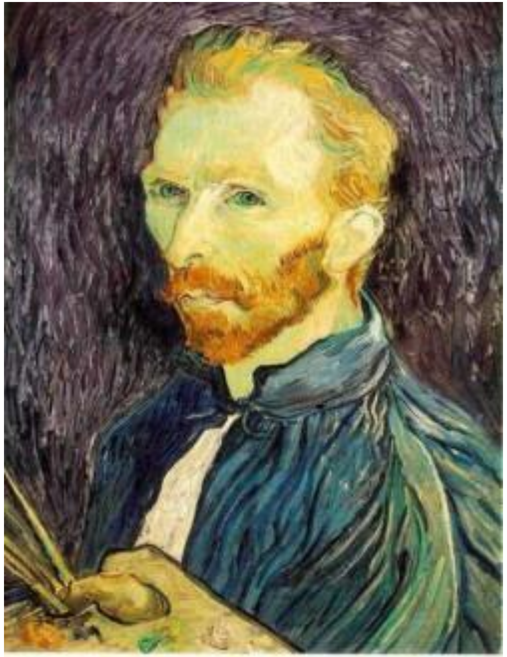
Good sampling



Bad sampling

Gaussian pre-filtering

- Solution: filter the image, *then* subsample



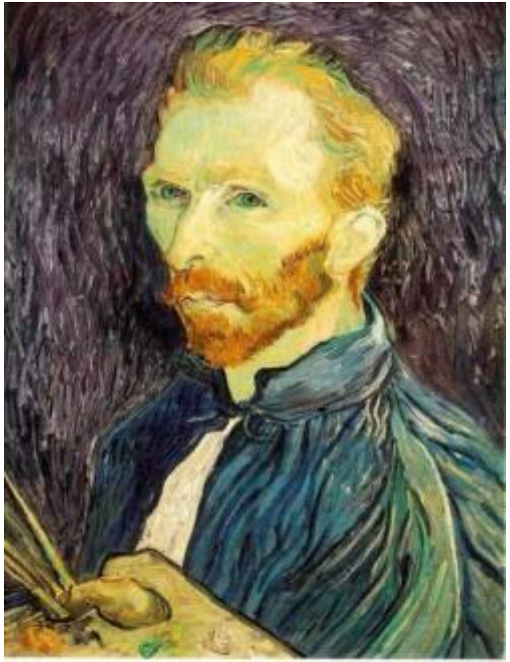
Gaussian 1/2

G 1/4

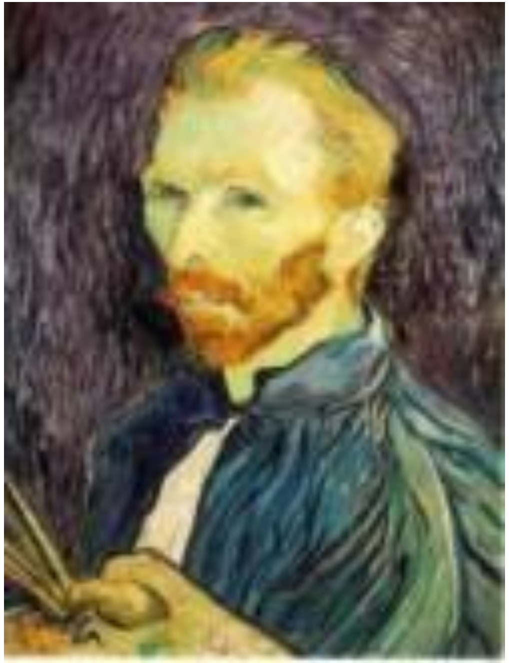
G 1/8

Down-sampling with Gaussian pre-filtering

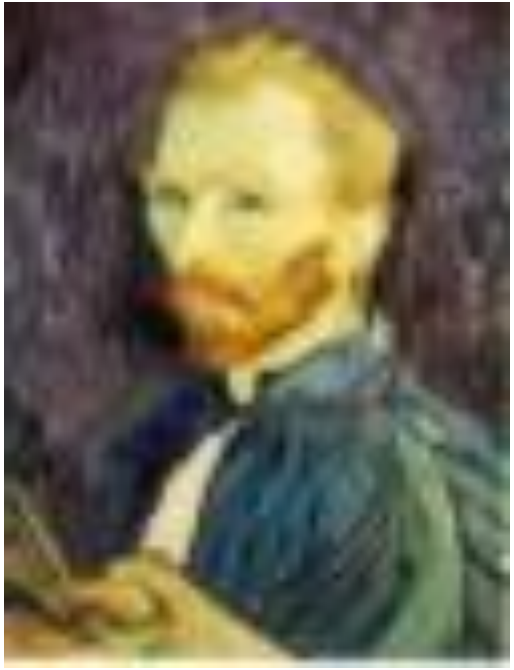
- Solution: filter the image, *then* subsample



Gaussian 1/2

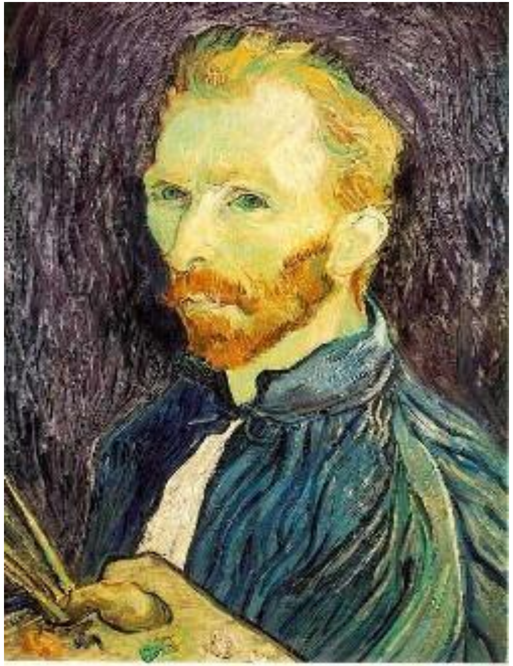


Gaussian 1/4 (2x zoom)

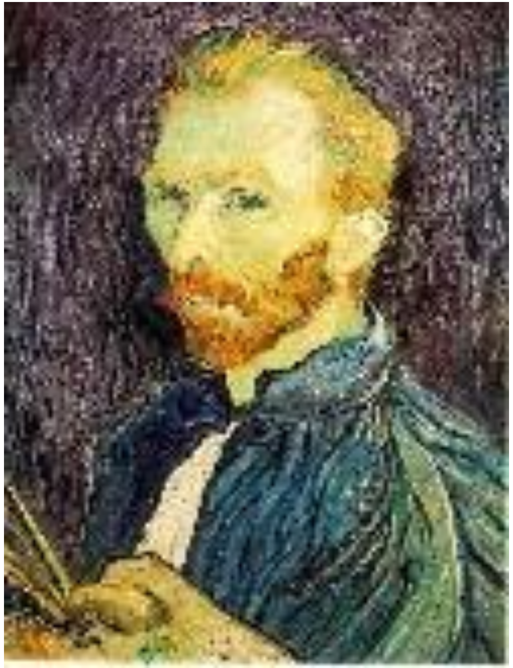


Gaussian 1/8 (4x zoom)

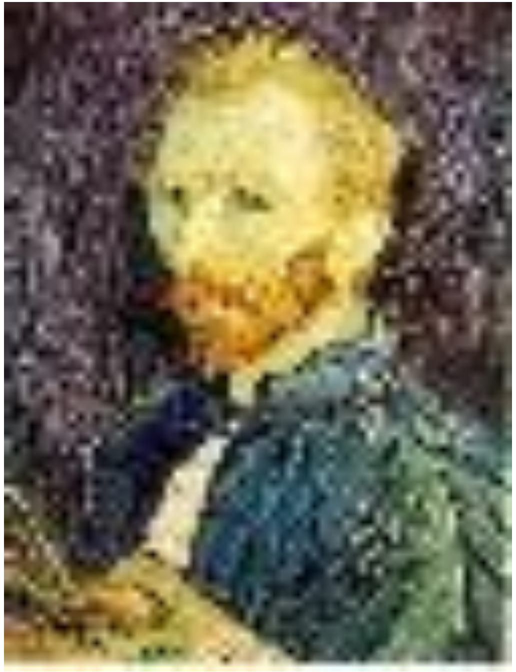
Compare with…



1/2



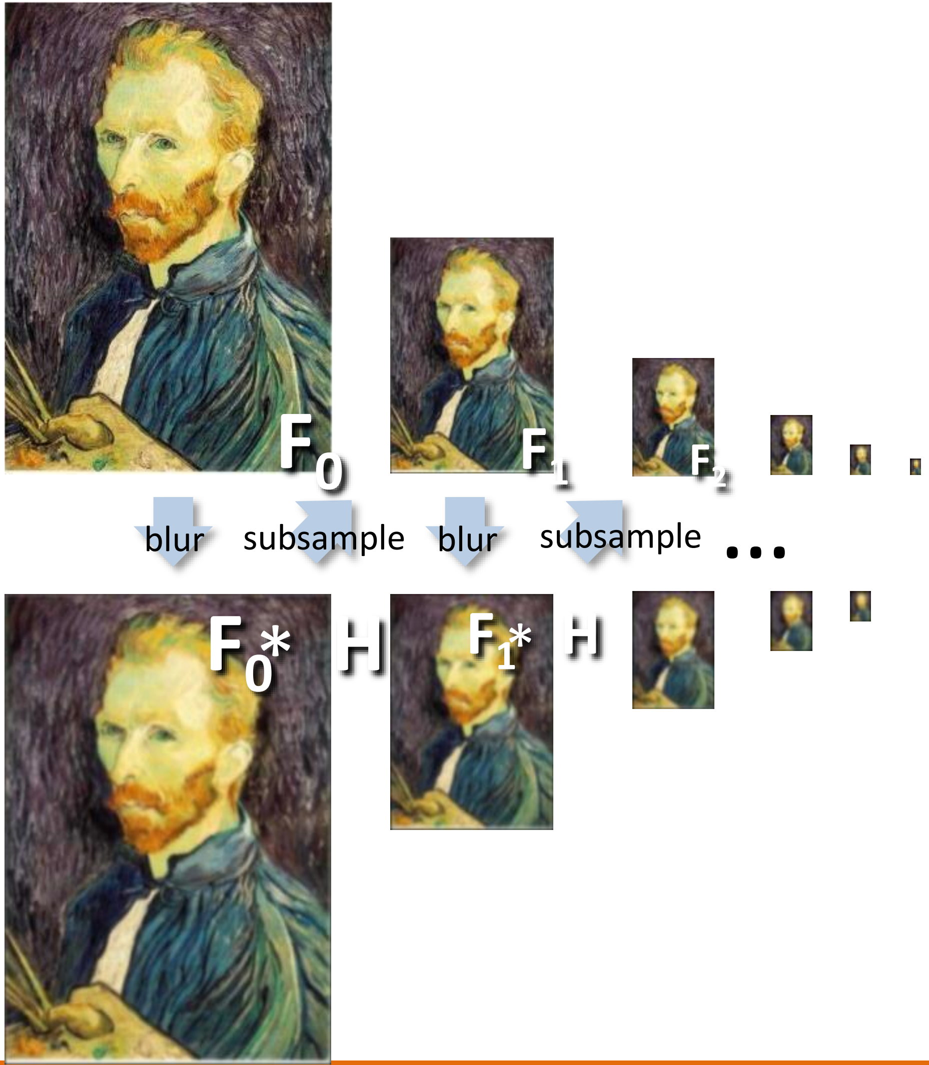
1/4 (2x zoom)



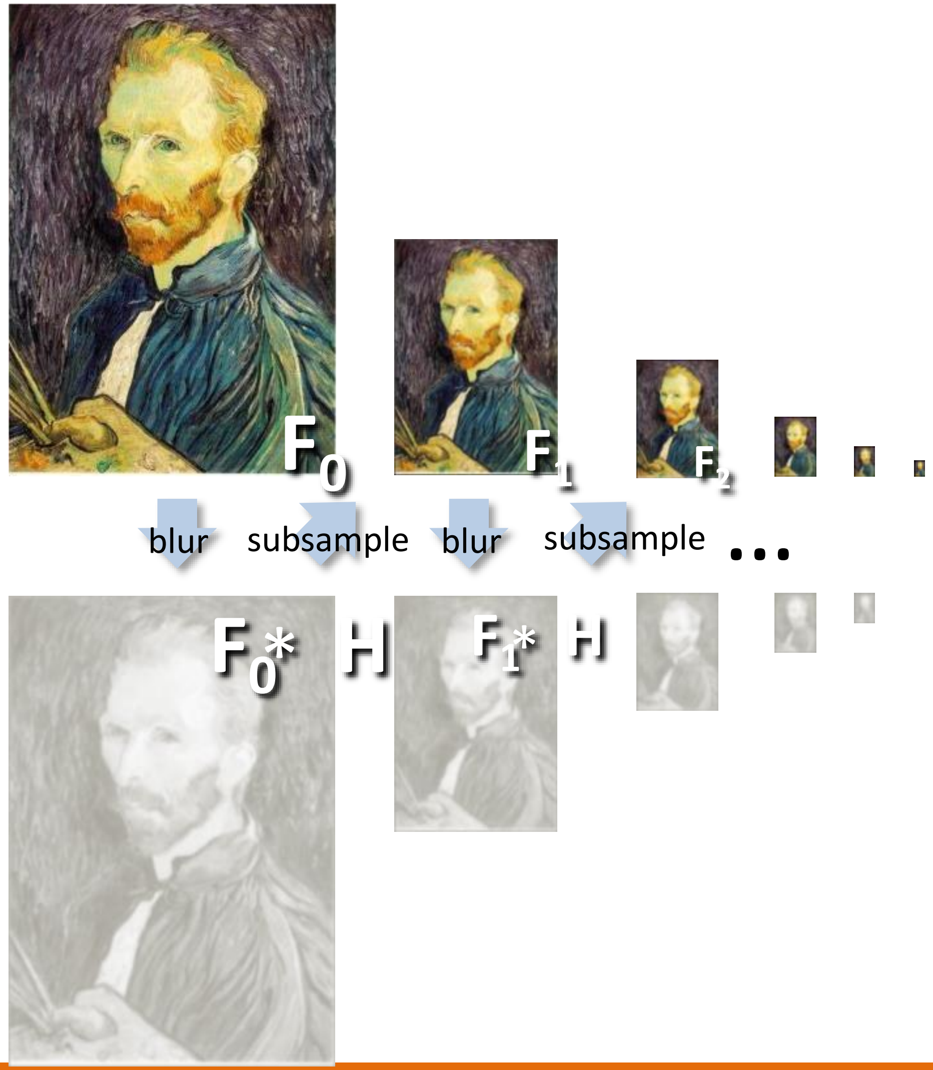
1/8 (4x zoom)

Down-sampling with Gaussian pre-filtering

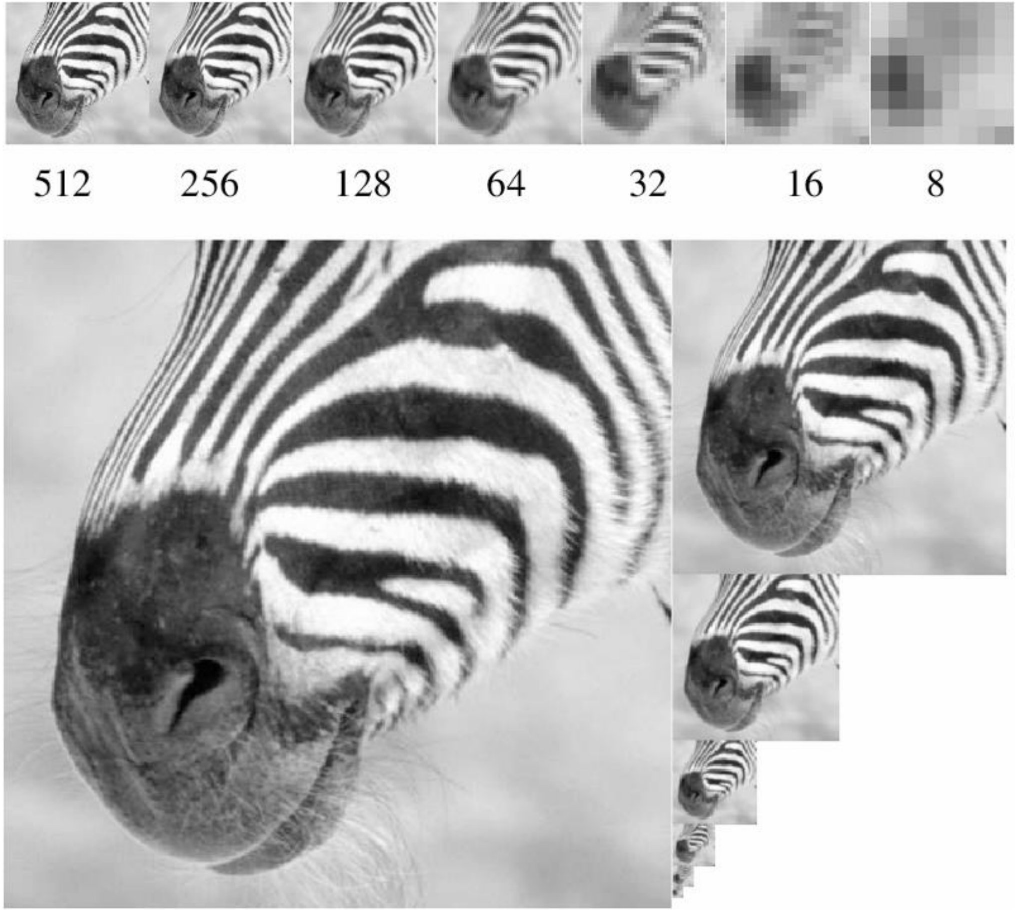
Smoothing removes high frequency components!



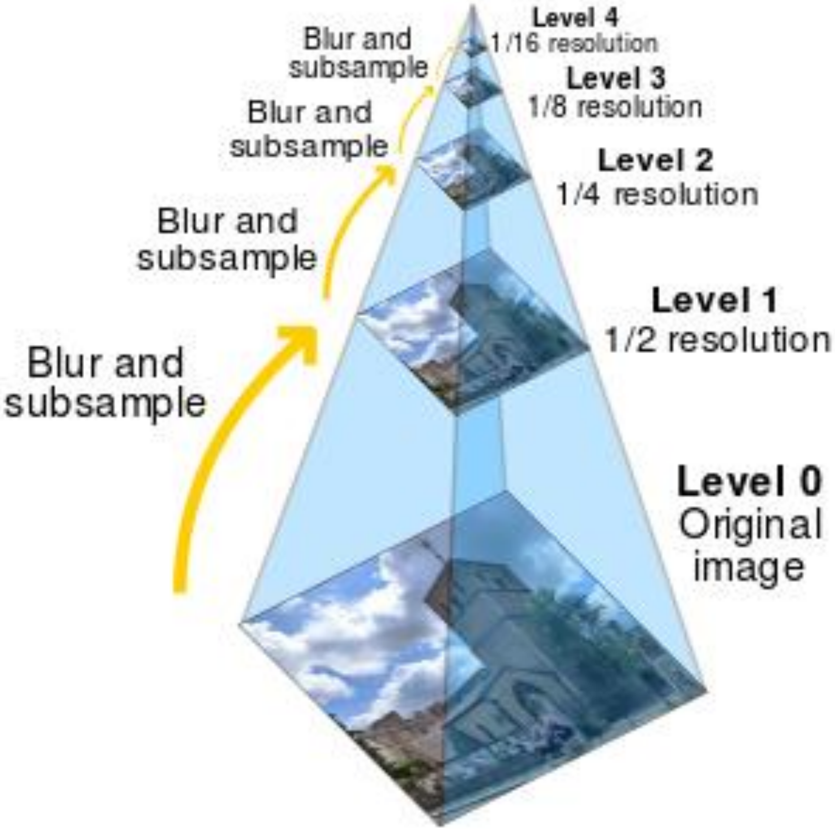
Gaussian
Pyramid



Gaussian pyramid



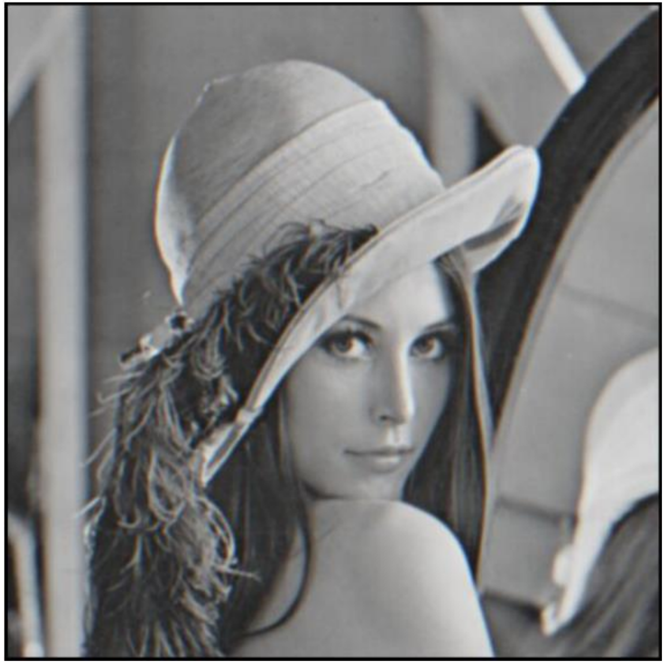
Gaussian pyramid



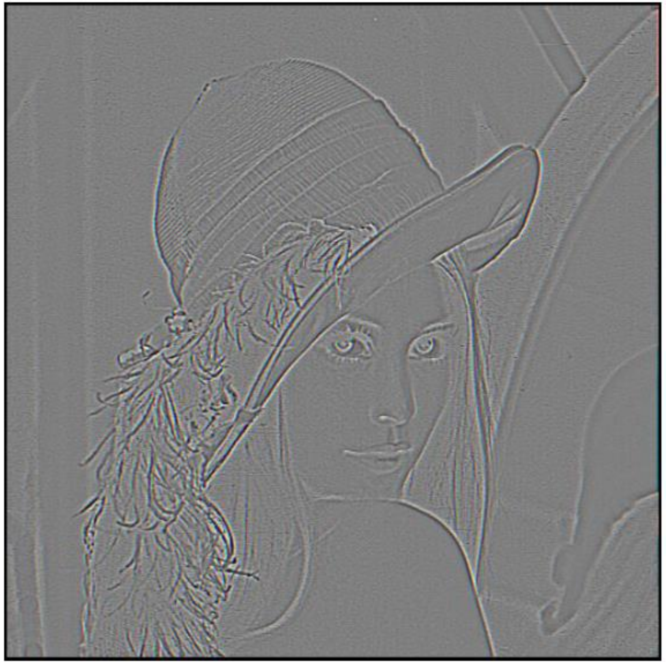
Courtesy: wikipedia

What does smoothing takes away?

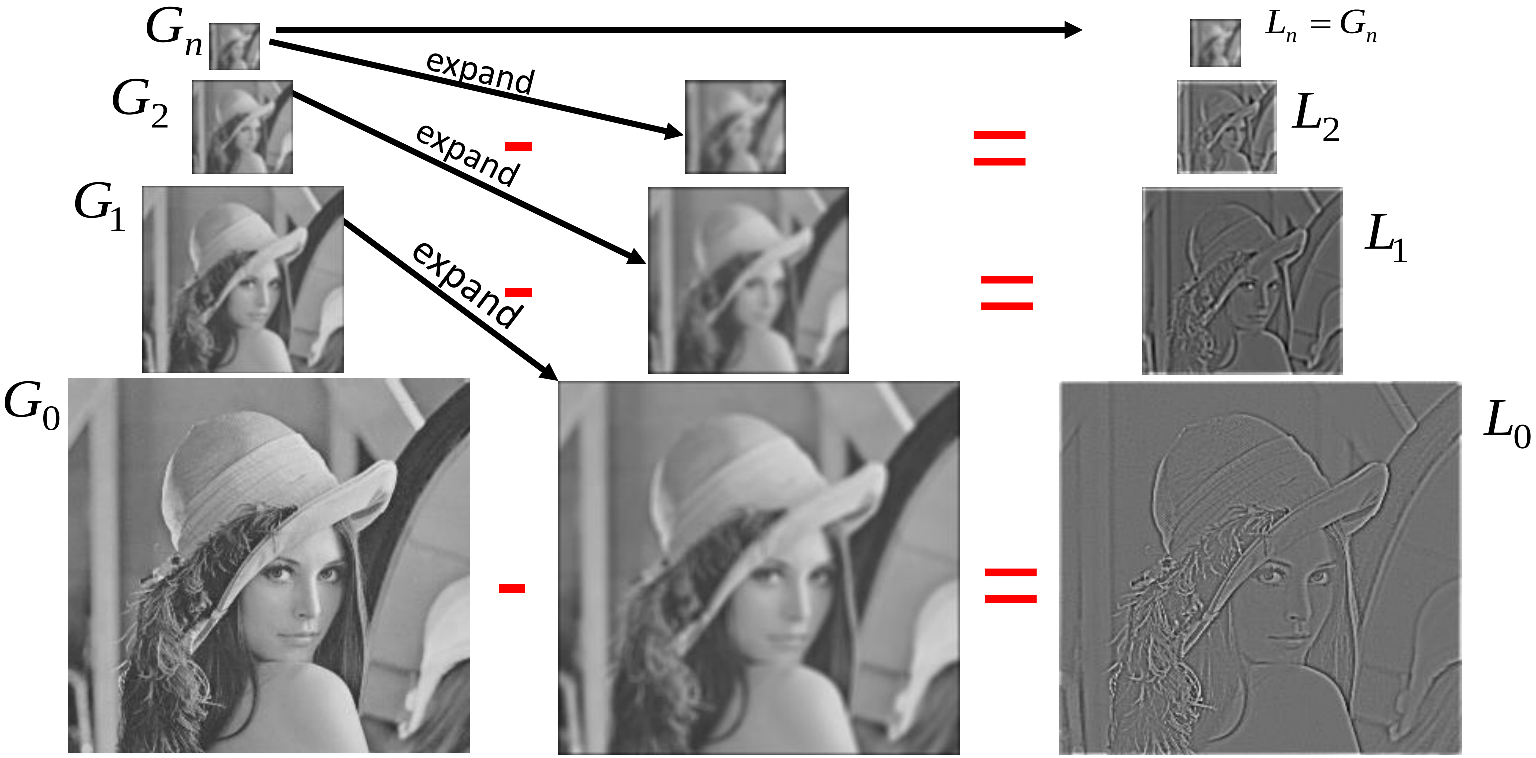
What does smoothing takes away?



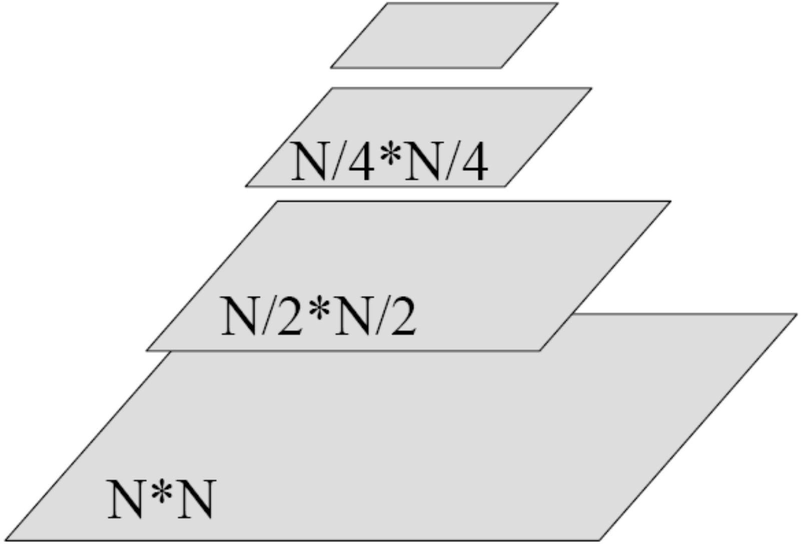
What does smoothing takes away?



Laplacian pyramid



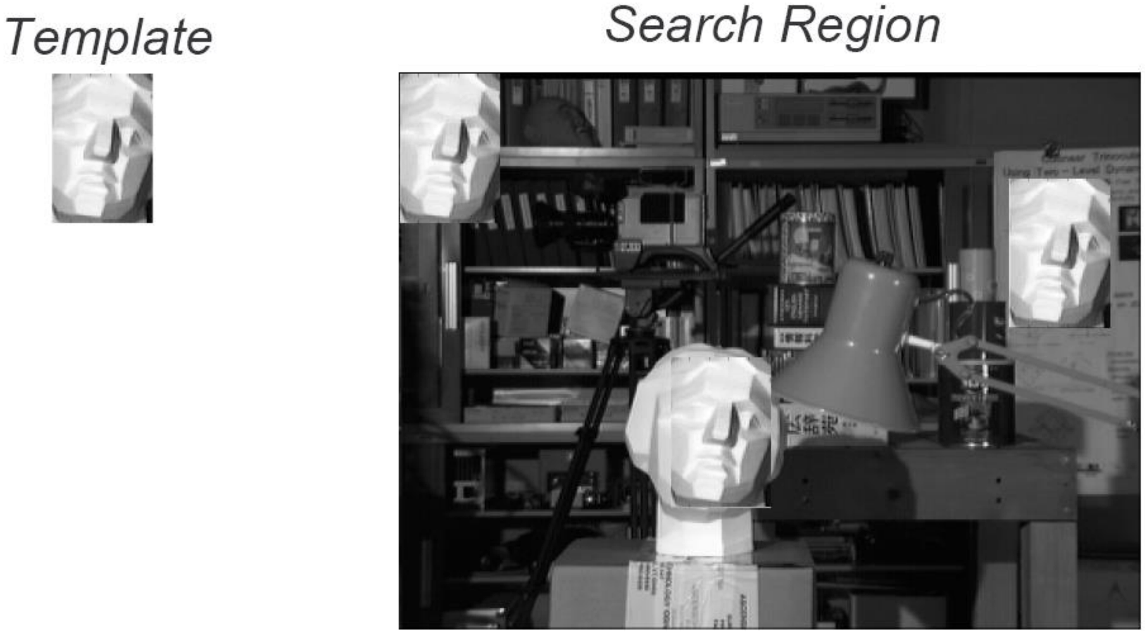
Space required for image pyramid



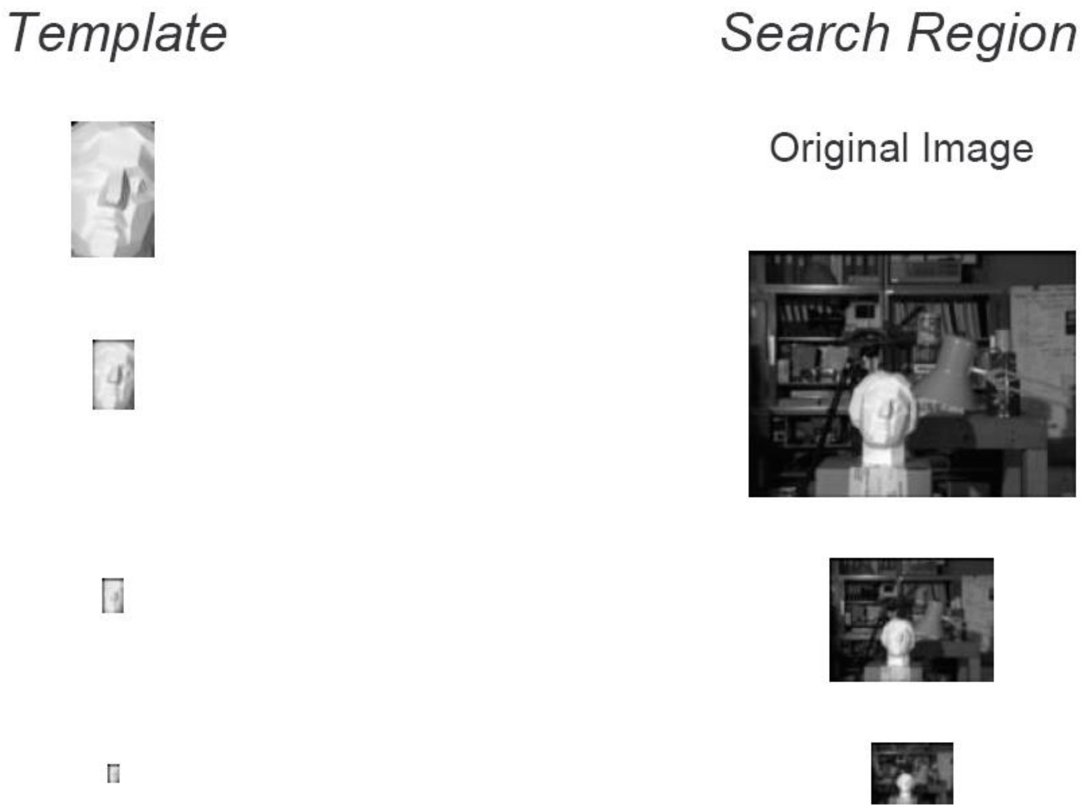
$$N^2 + \frac{1}{4}N^2 + \frac{1}{16} + \cdots = 1\frac{1}{3}N^2$$

Application Gaussian pyramid

- Efficient multi-scale detection

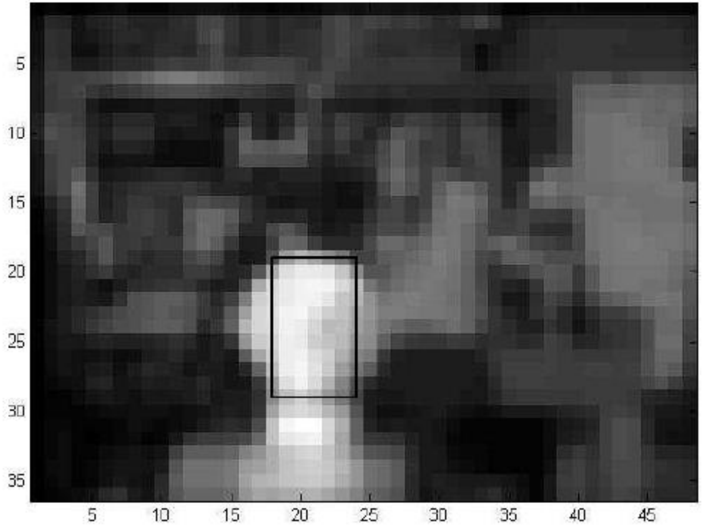


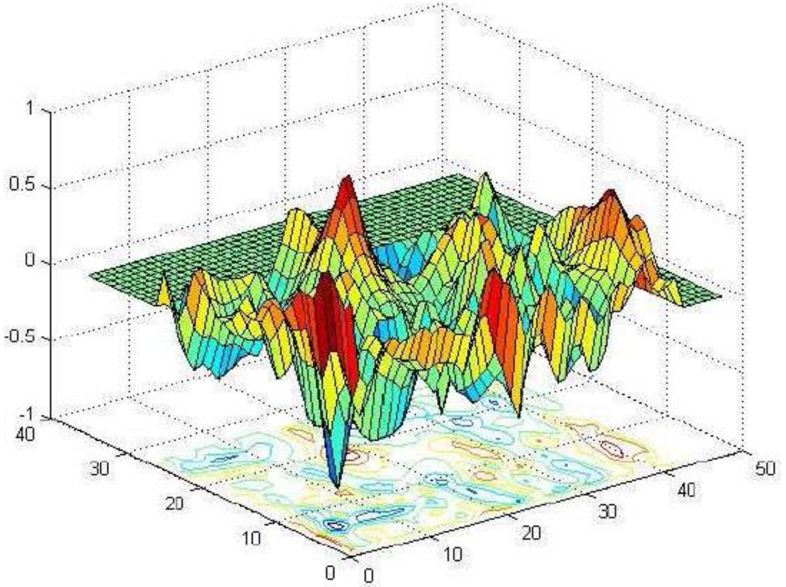
Efficient multi-scale detection



Efficient multi-scale detection

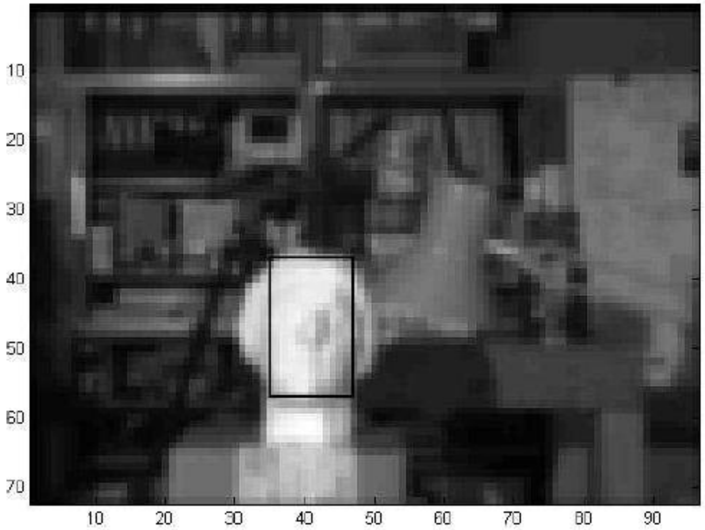
- Level 3 search: at the lowest level we search the entire image with correlation template

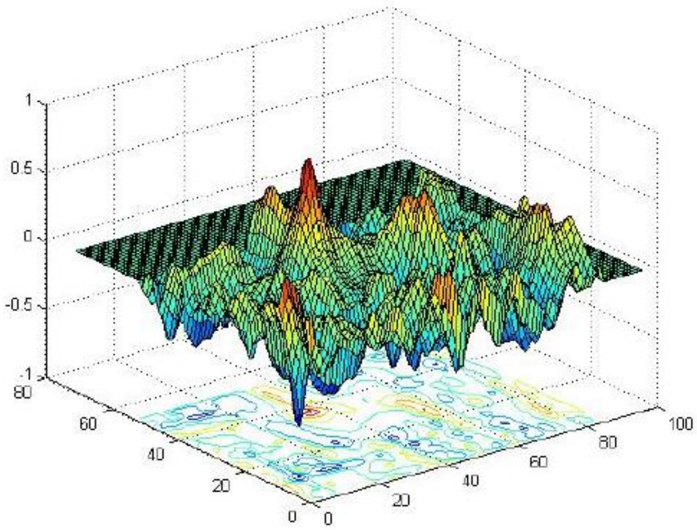




Efficient multi-scale detection

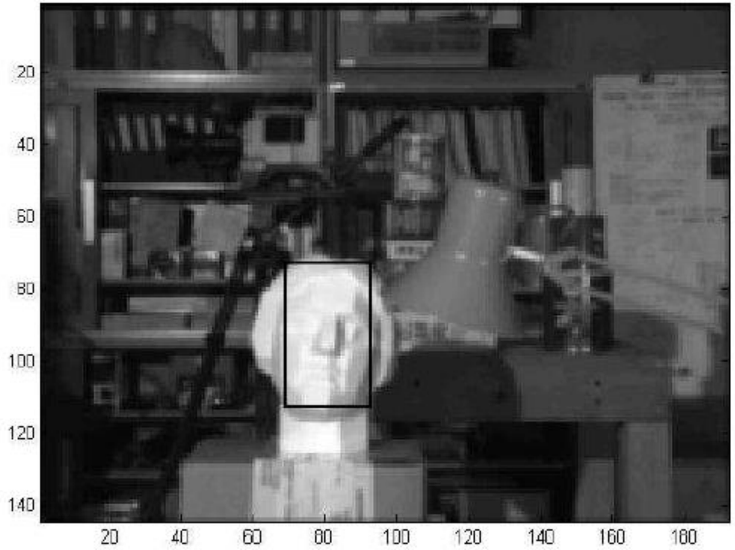
- Level 2 search: constrained to a neighbourhood of high response centers in the previous level

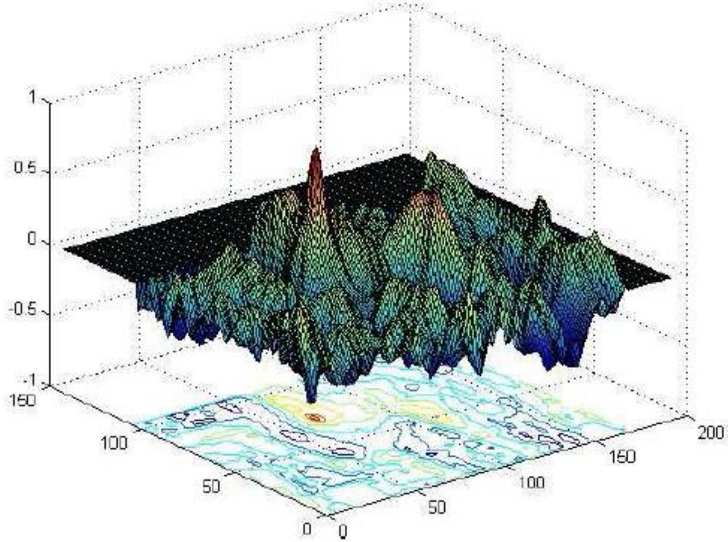




Efficient multi-scale detection

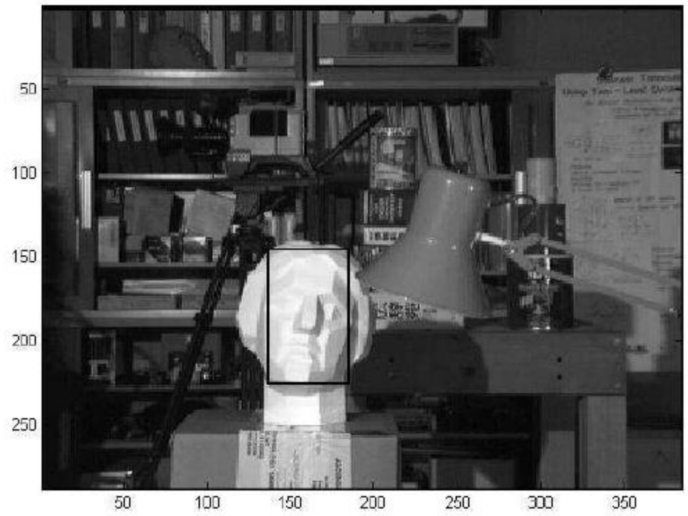
- Level 1 search: again constrained based on results of level 1

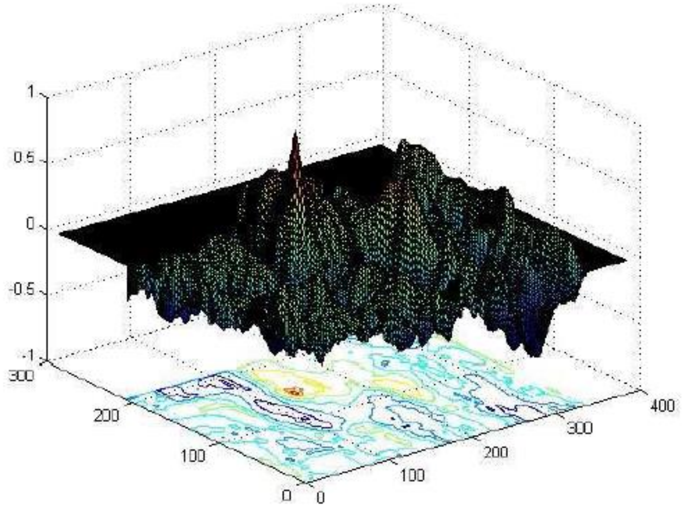




Efficient multi-scale detection

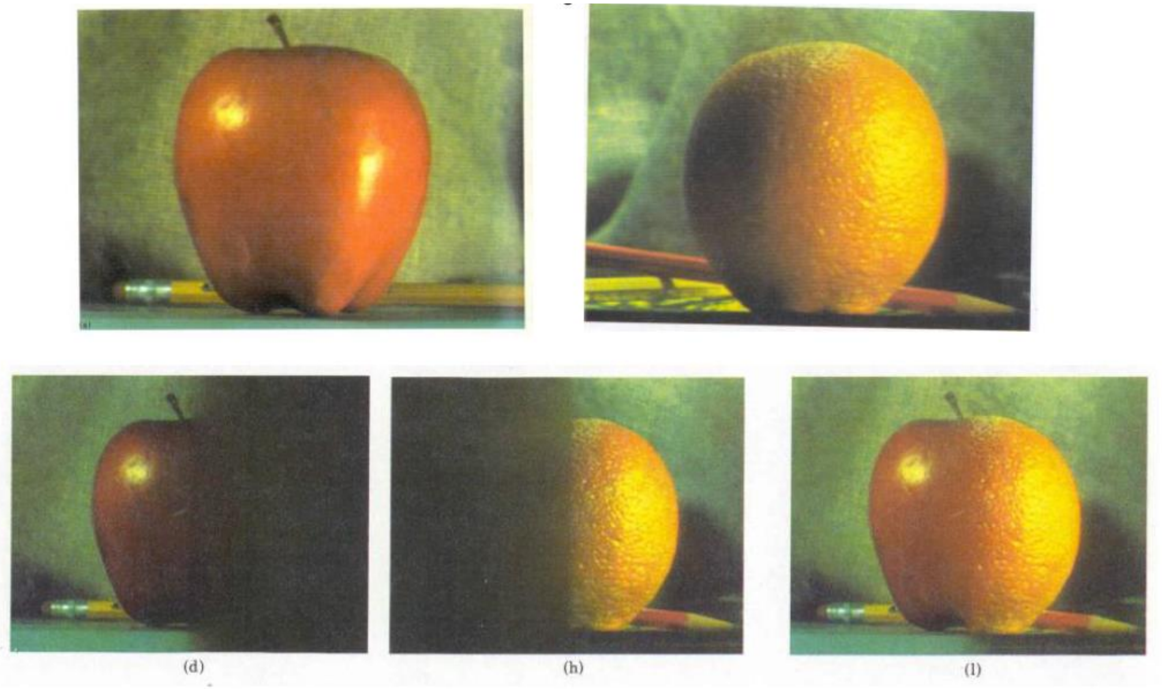
- Level 0 search: total time reduced to 0.5 second from 31 seconds





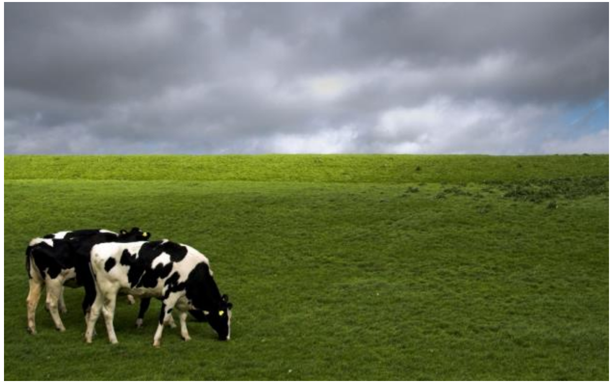
Application Gaussian pyramid

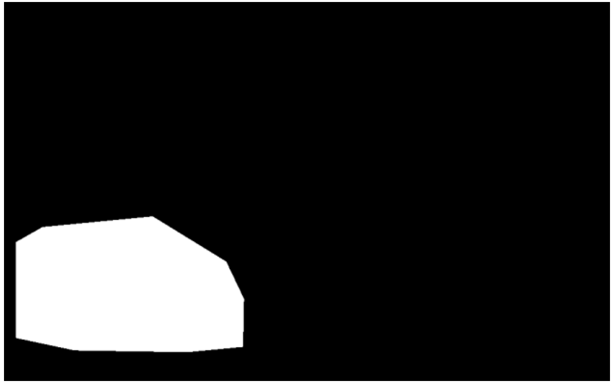
- Blending Apples and Oranges

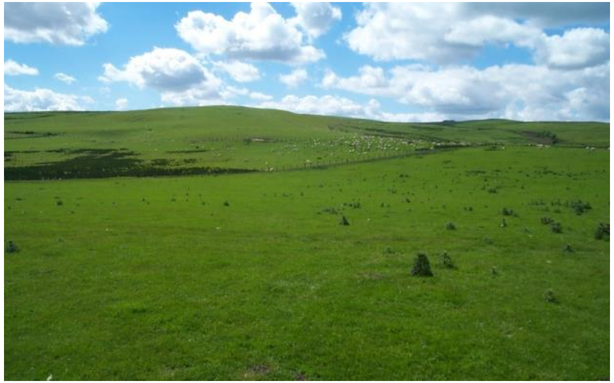


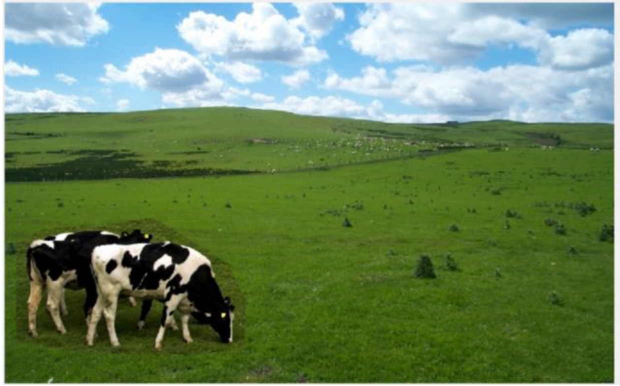
Burt and Adelson 1983

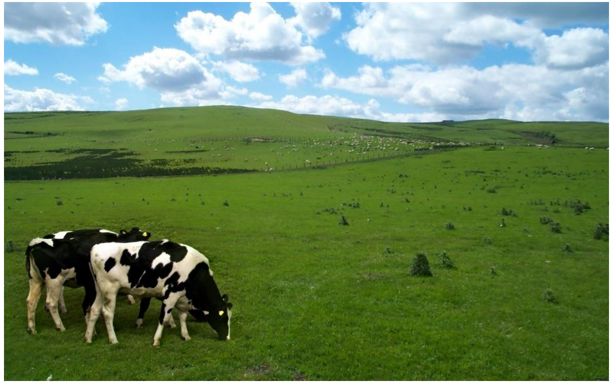
Application Gaussian pyramid



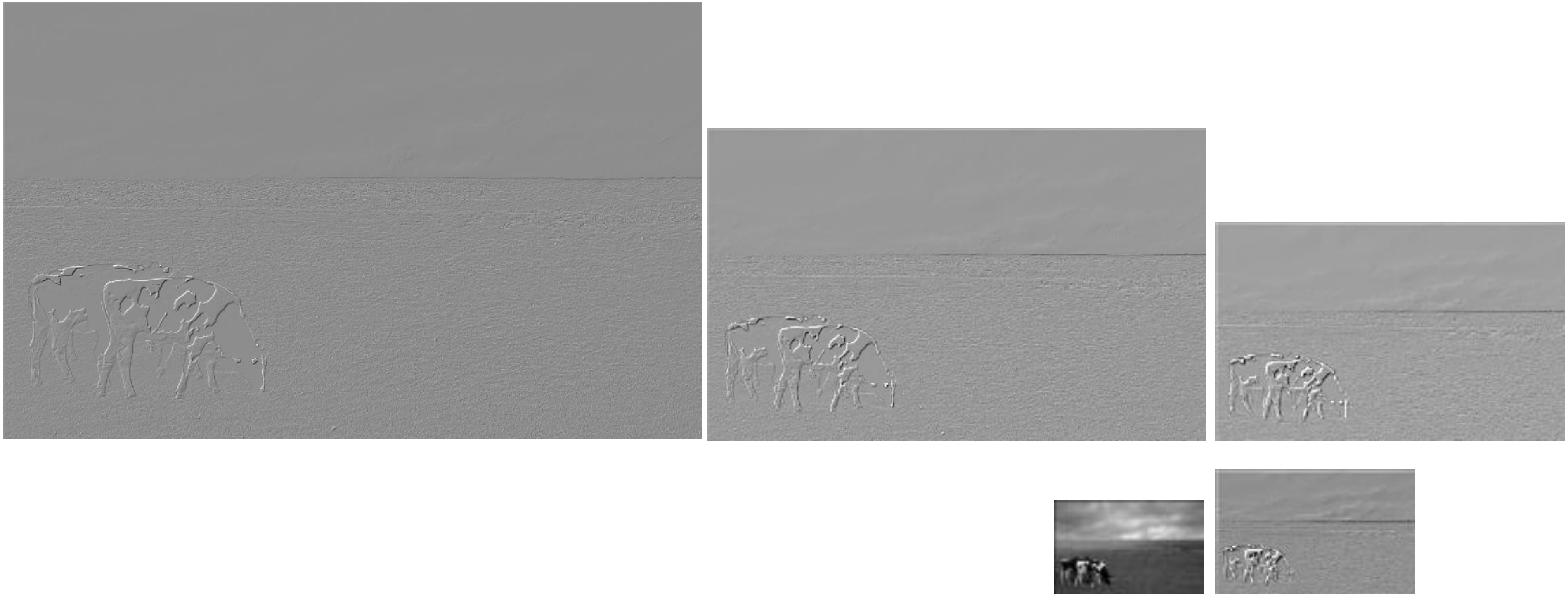




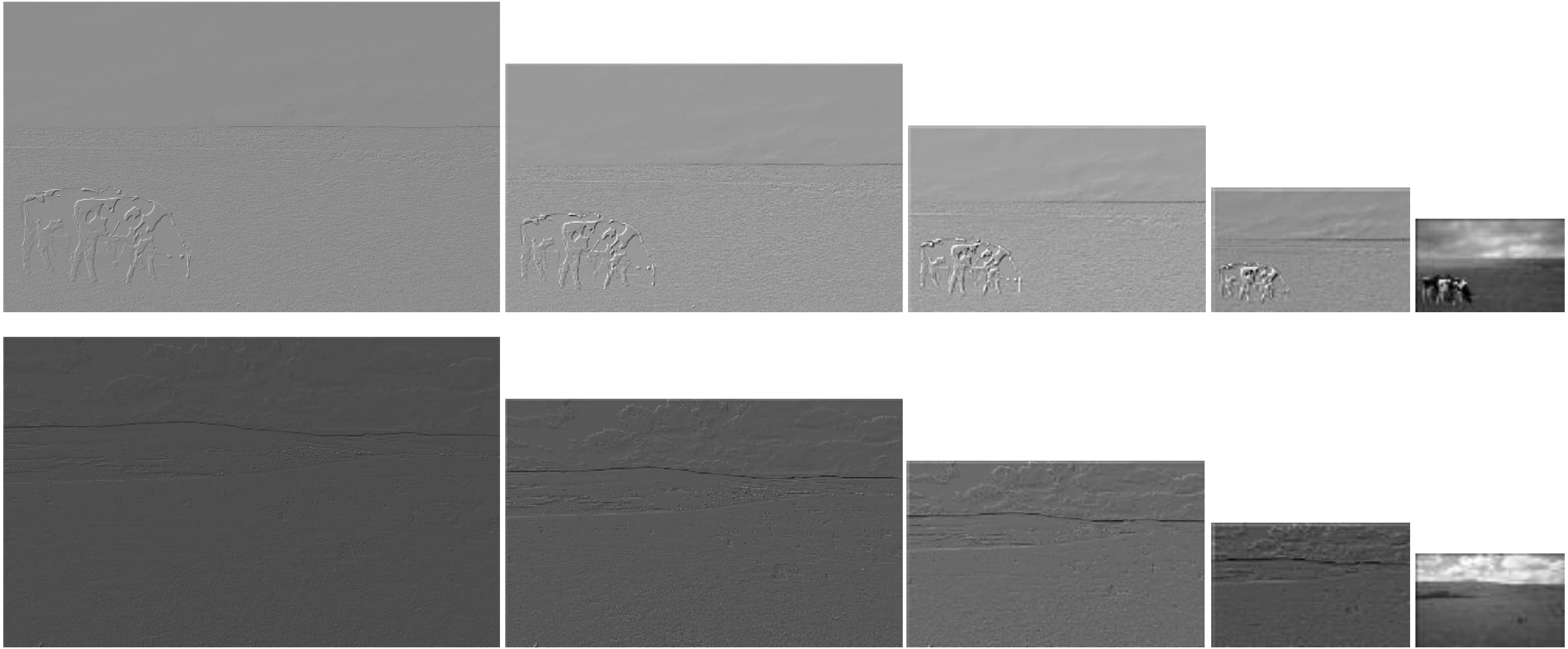




Application Gaussian pyramid



Application Gaussian pyramid



Application Gaussian pyramid

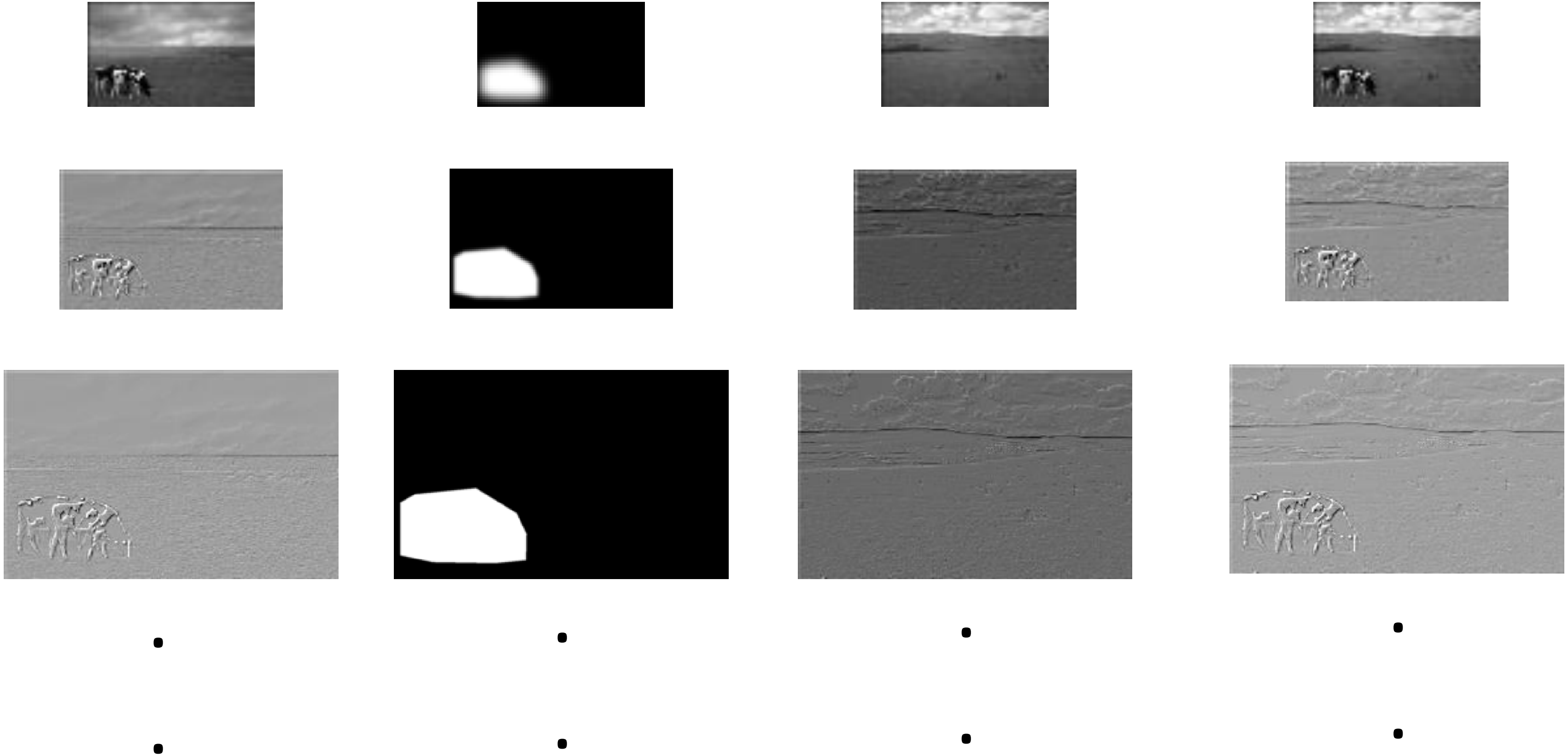


Image Blending

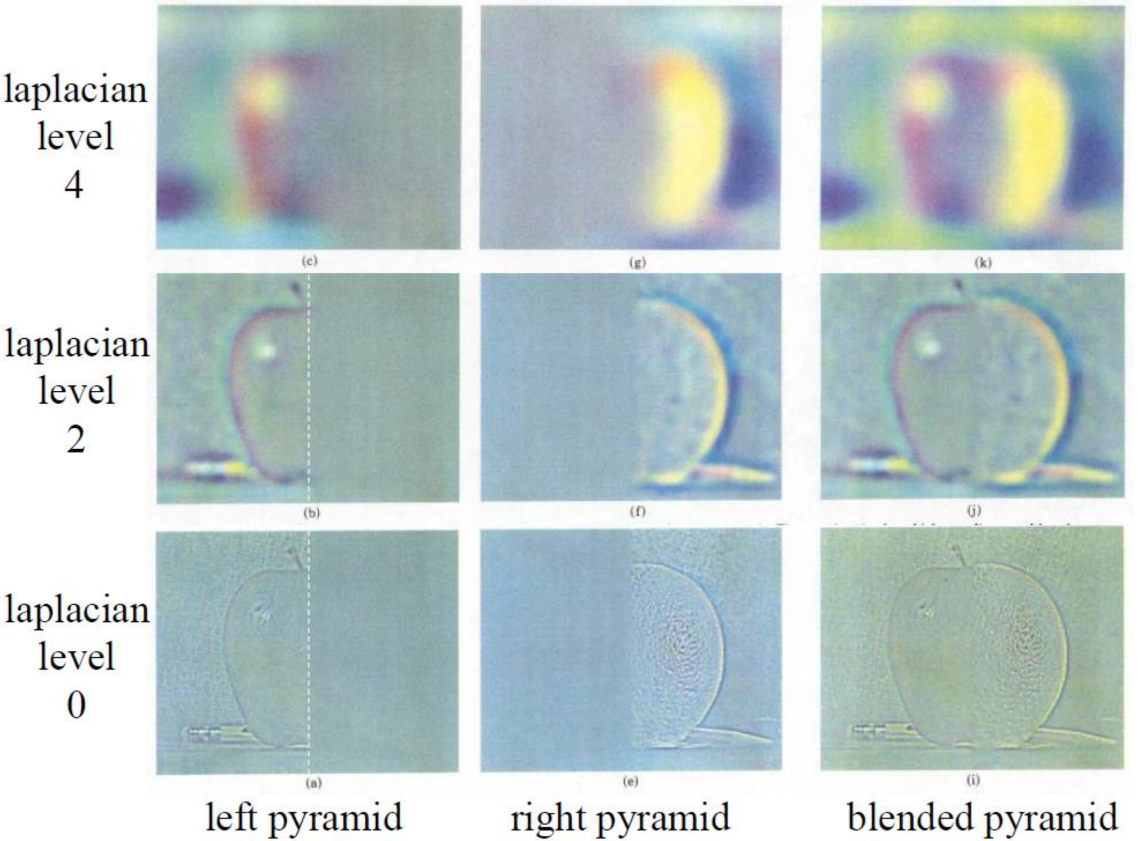
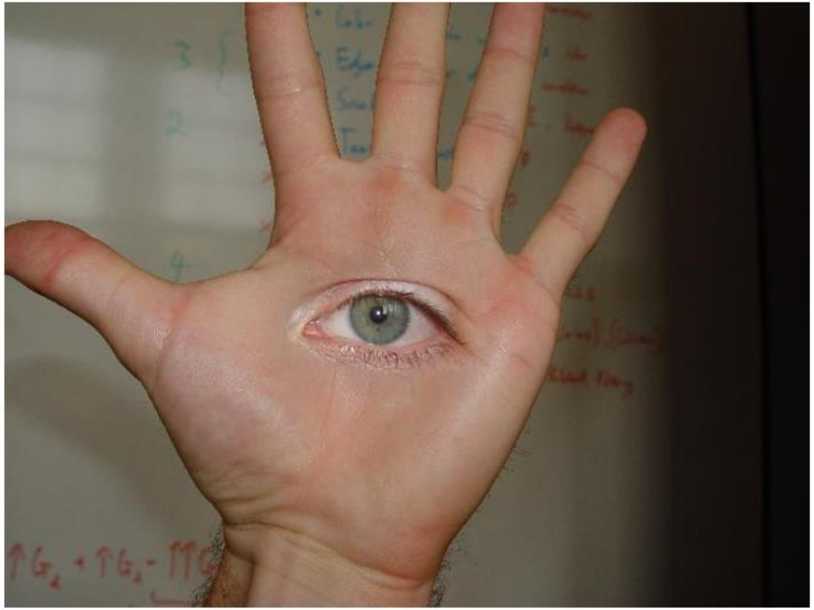
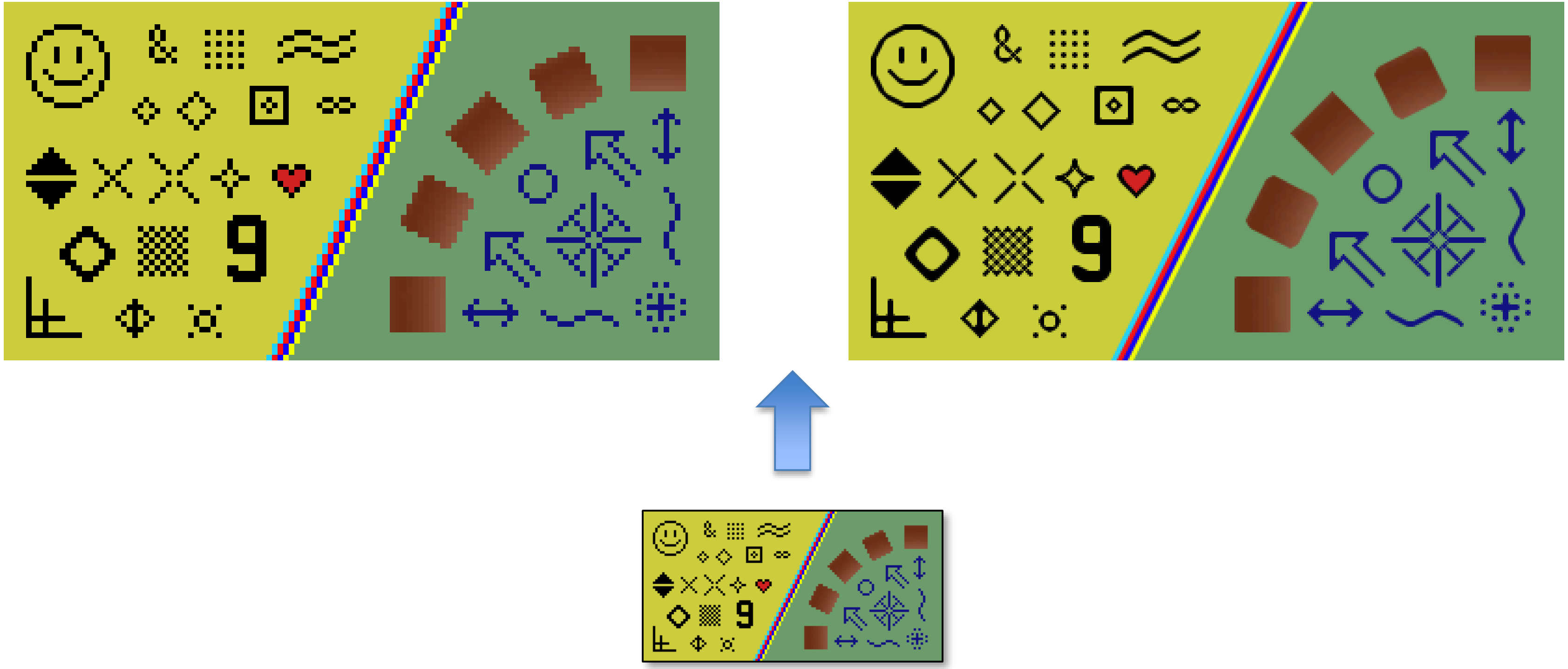


Image Blending

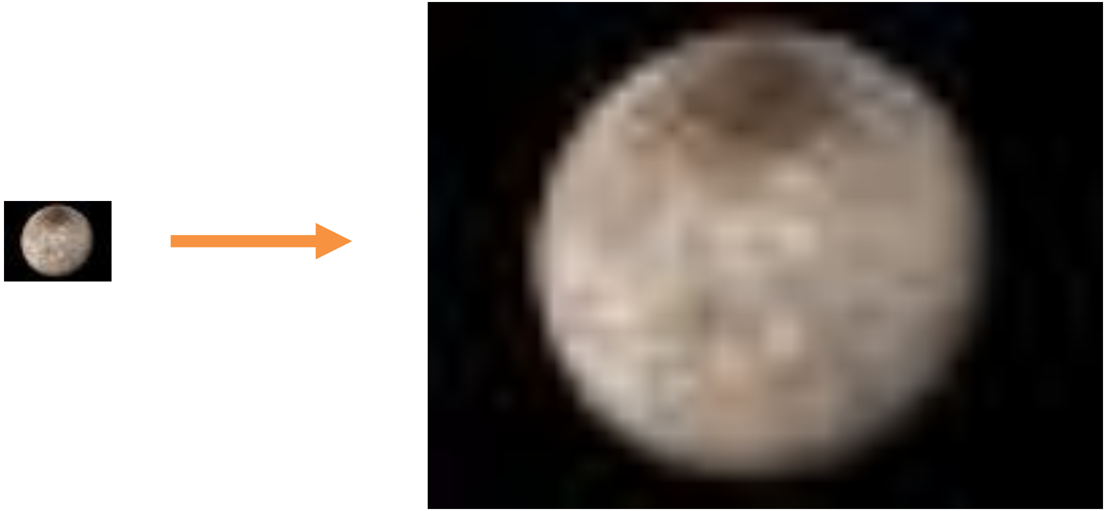


© prof. dmartin

Questions?

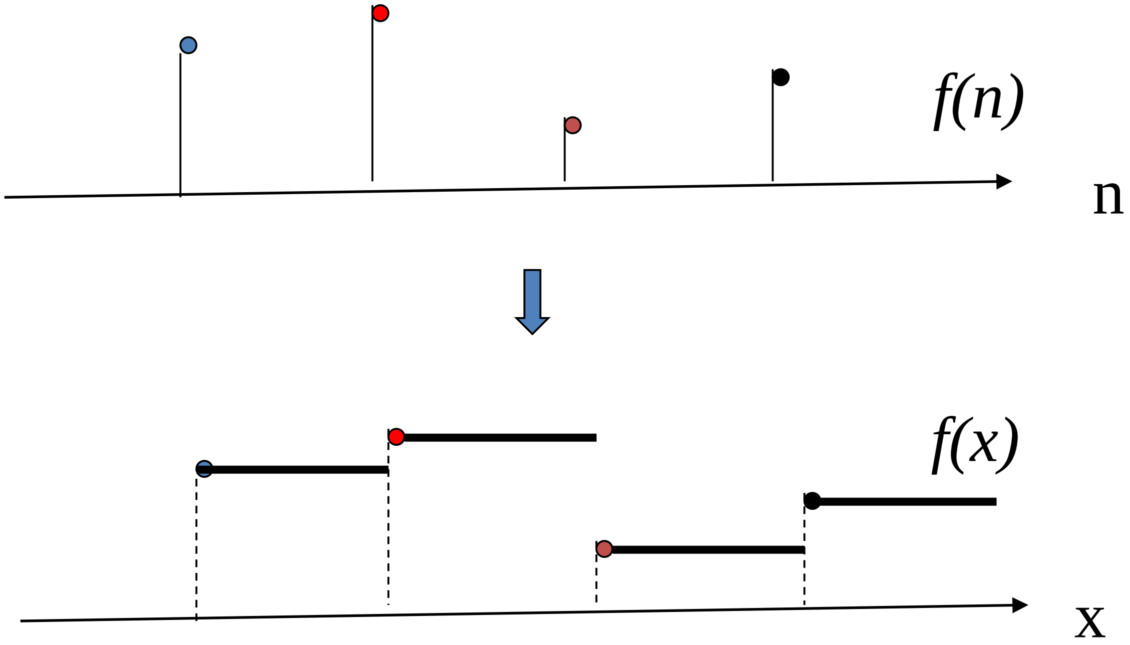


Up-sampling



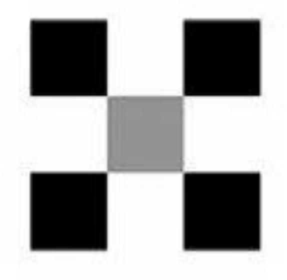
Nearest neighbor interpolation

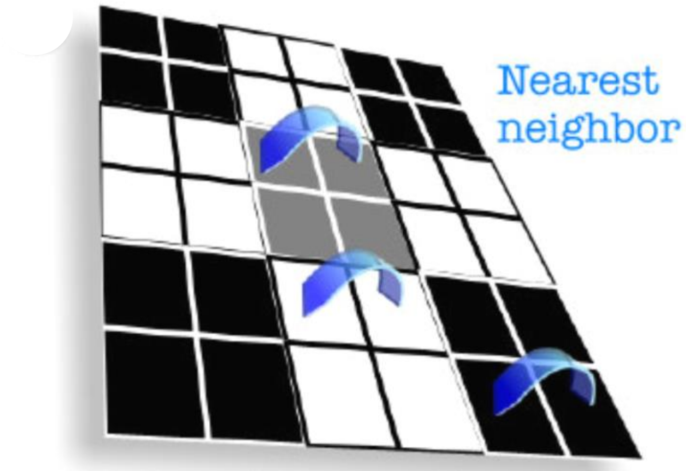
- Just repeat elements



Nearest Neighbour Interpolation

- Just repeat elements

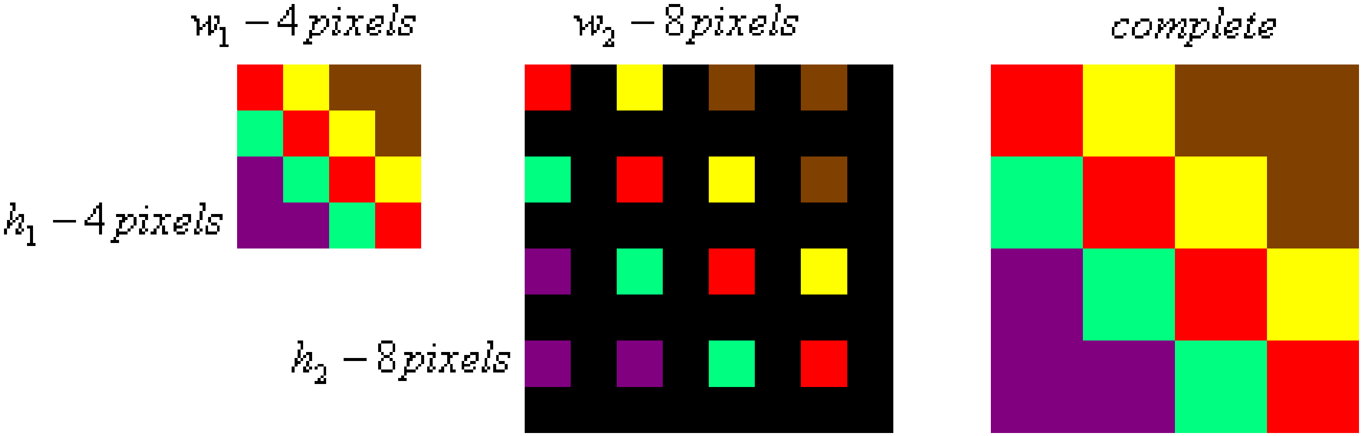




Nearest Neigbor calculation for new pixels uses the simplest of math for reassigning picture elements.

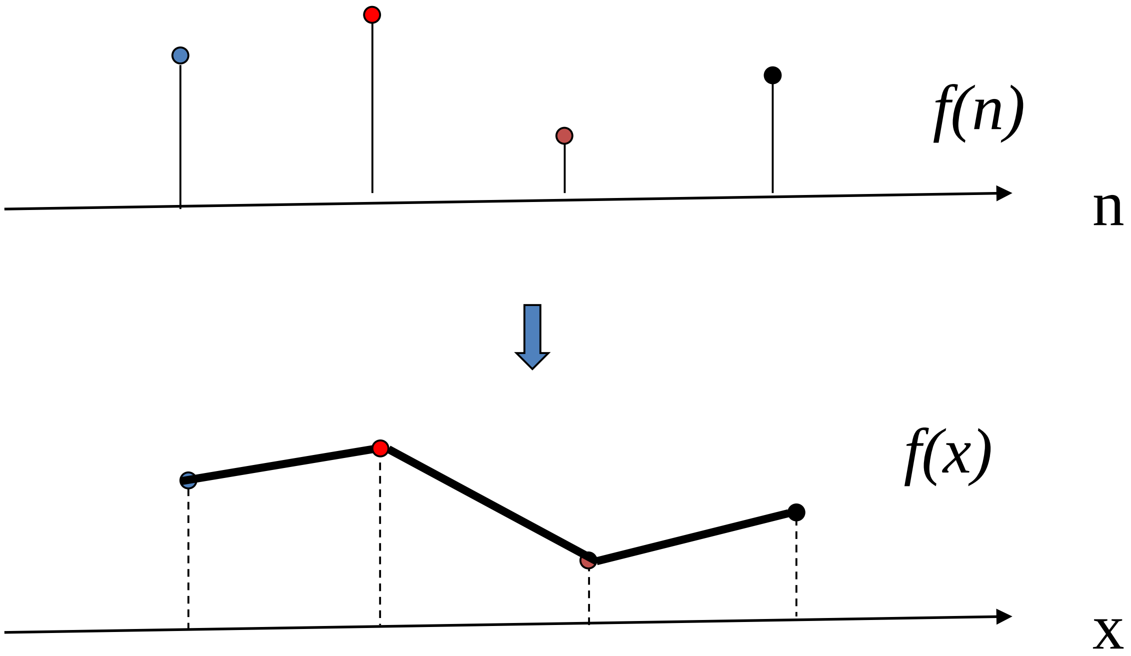
Nearest Neighbour Interpolation

- Just repeat elements



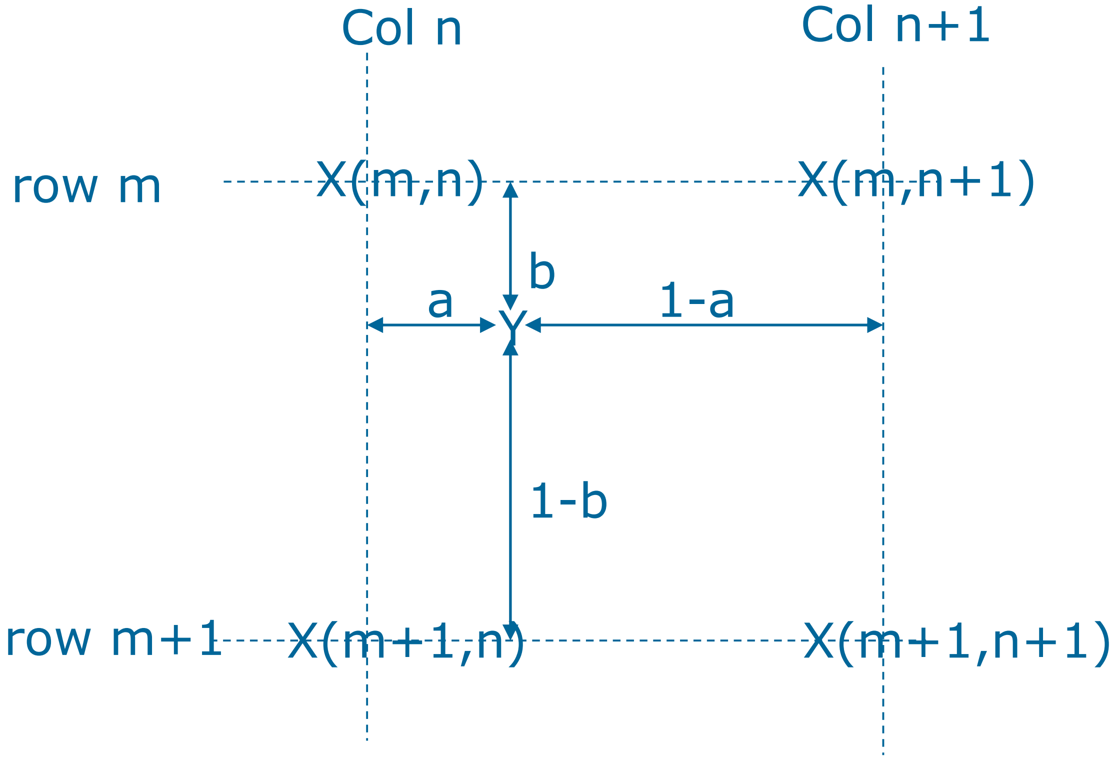
Linear interpolation

- Linear combination



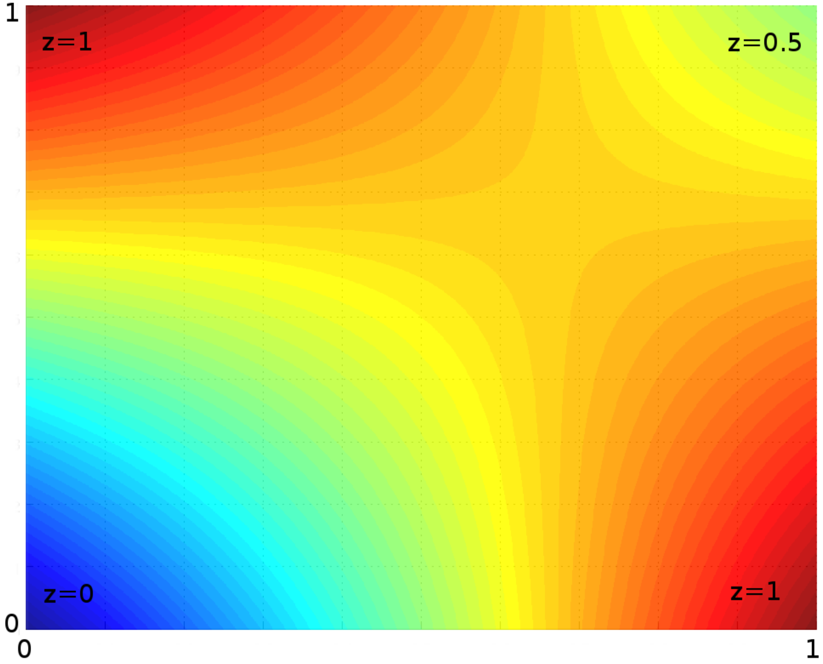
Bilinear interpolation (2D)

- Divide and conquer



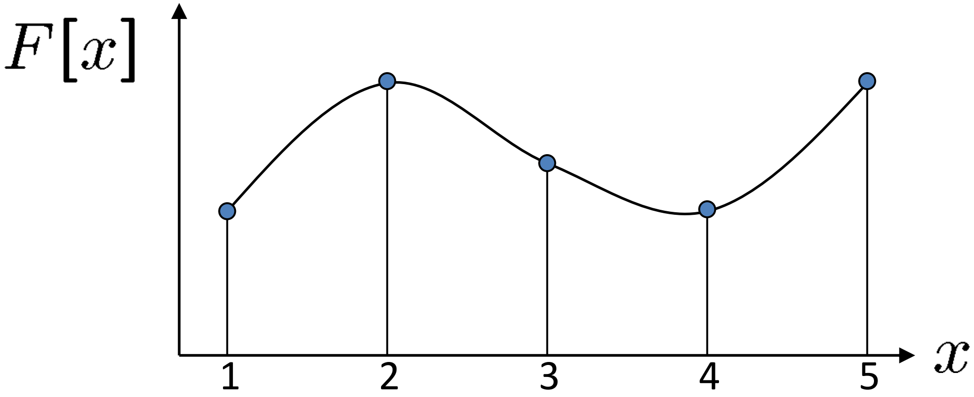
Bilinear interpolation (2D)

- 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.



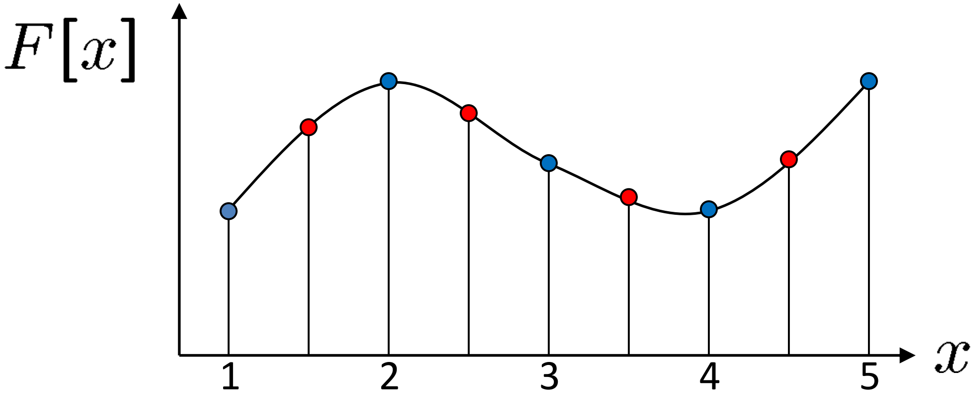
Principled approach to interpolation

- Estimate the function from quantized values



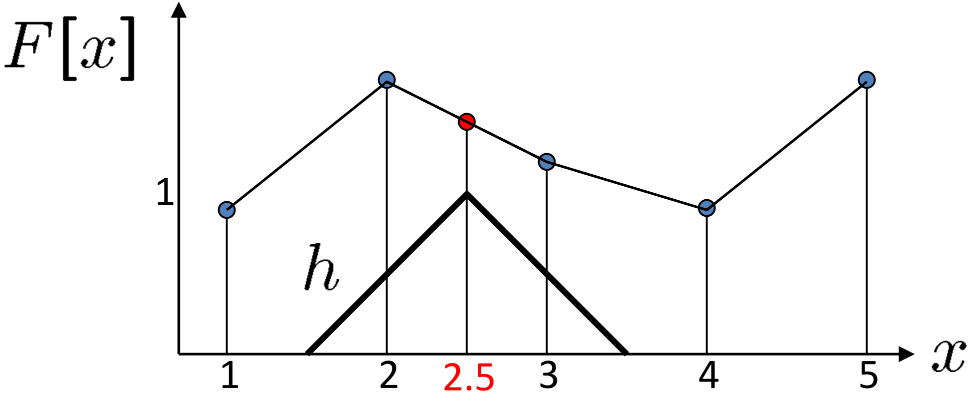
Principled approach to interpolation

- Estimate the function from quantized values

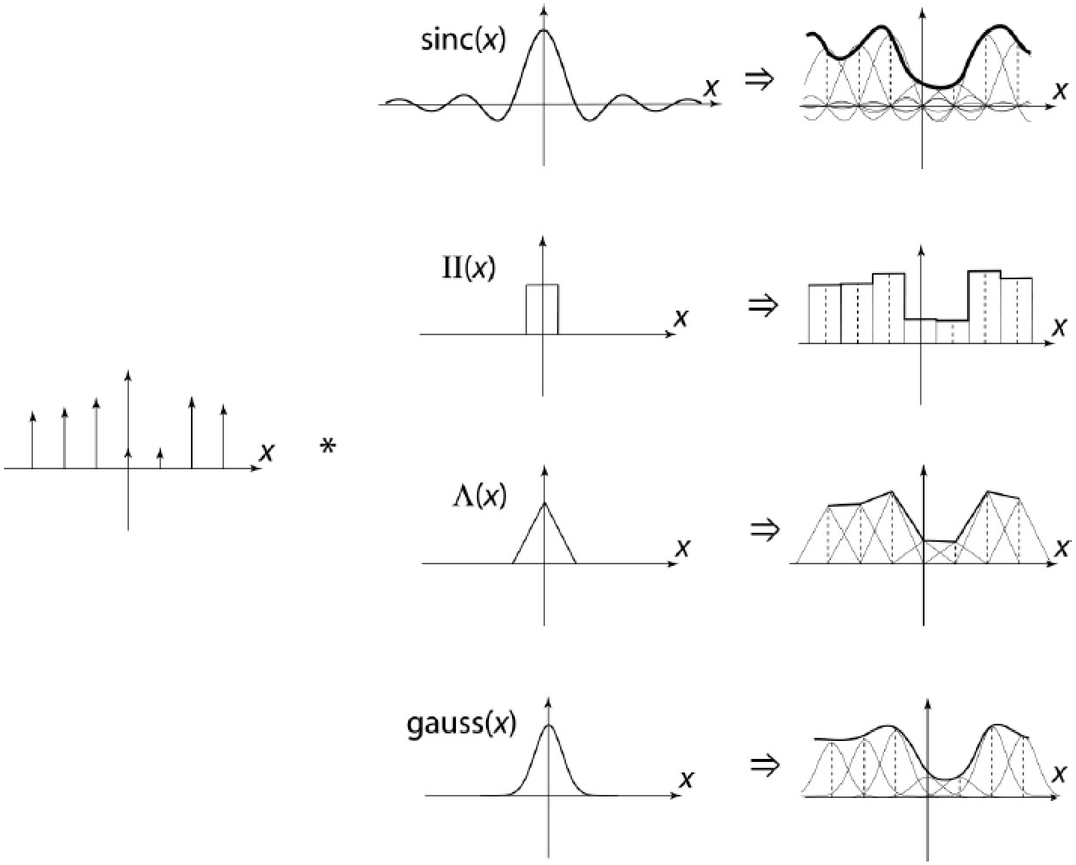


Principled approach to interpolation

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



Interpolation as filtering



Sinc interpolation

Nearest-neighbor interpolation

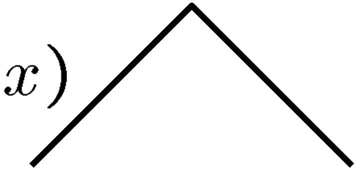
Linear interpolation

Gaussian reconstruction

source: B. Curless

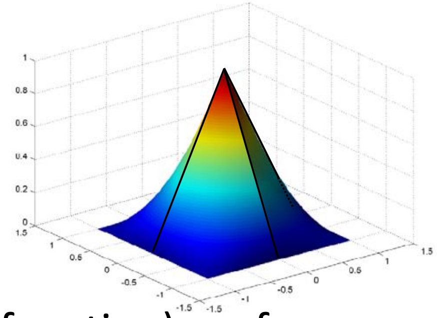
From 1D to 2D

$h(x)$



performs
linear interpolation

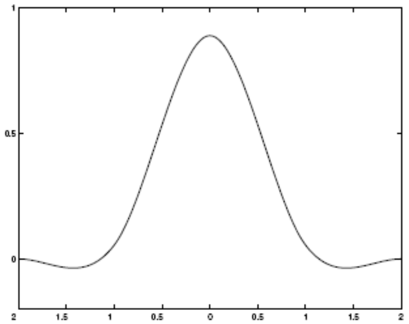
$h(x, y)$



(tent function) performs
bilinear interpolation

Bicubic filter

- Commonly used

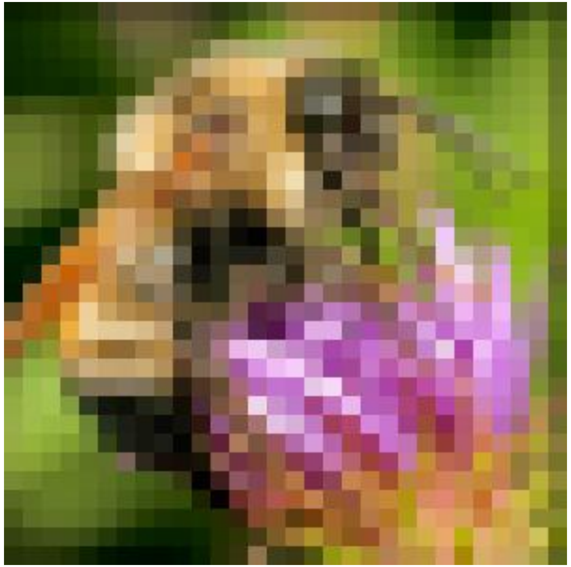


Cubic reconstruction filter

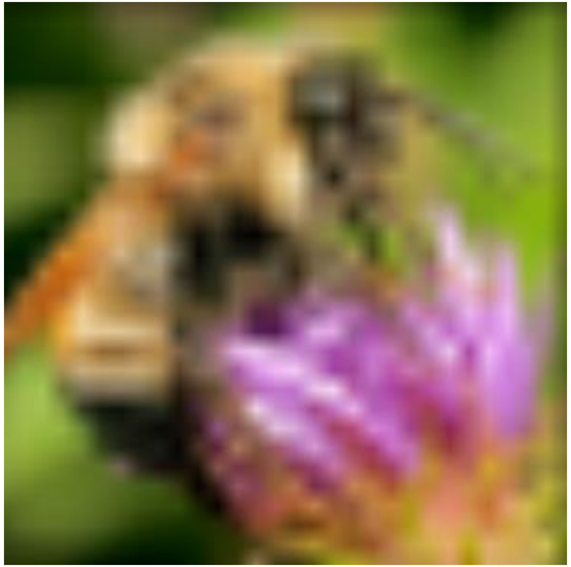
More advanced interpolation are adaptive, for example edge sensitive interpolation!

Image interpolation

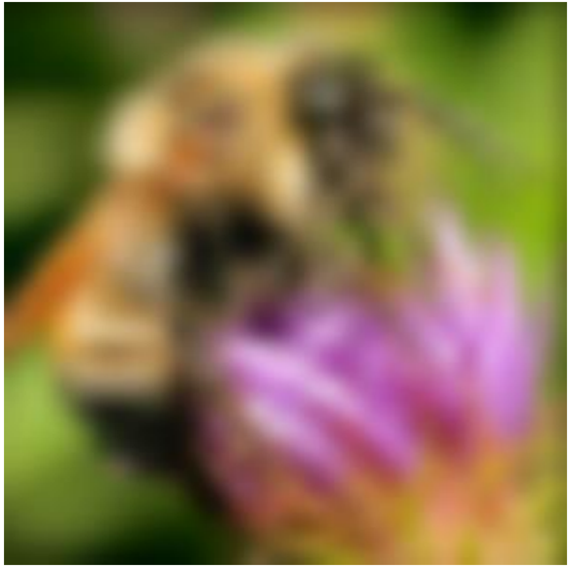
Original image: x 10



Nearest-neighbor interpolation



Bilinear interpolation



Bicubic interpolation

Image interpolation

Also used for *resampling*