Computer Vision and Image Processing

Gil Ben-Artzi

3

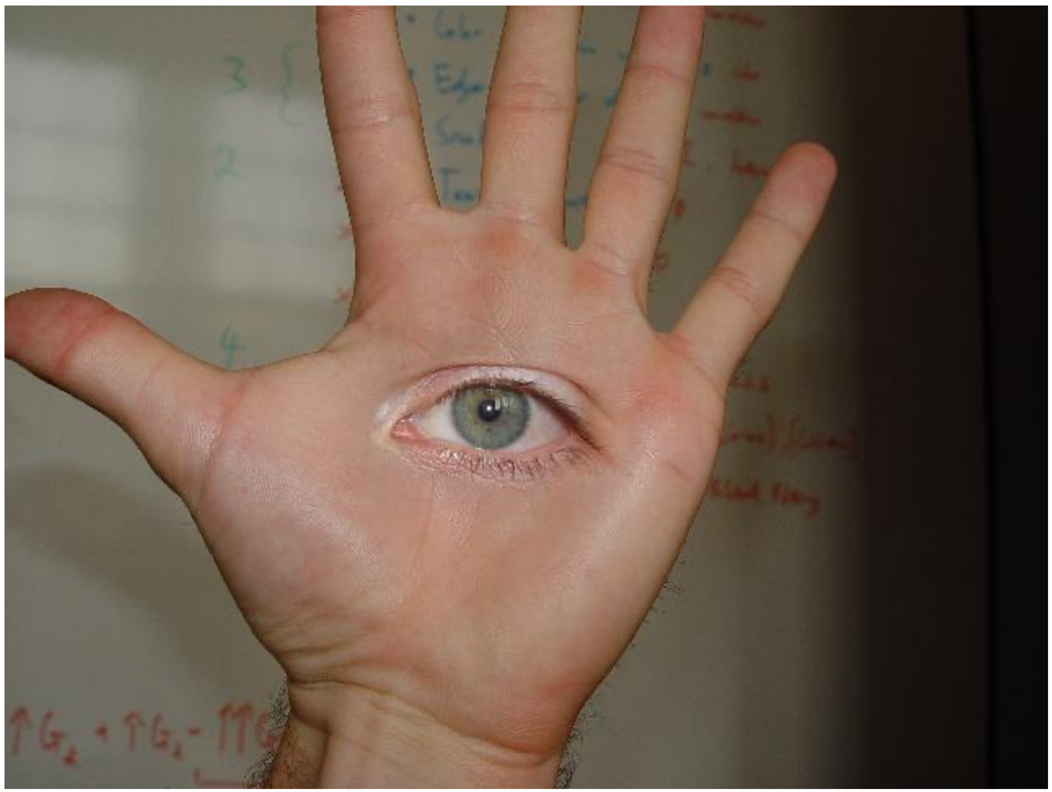
Agenda

- Week 1
  - Image Enhancement: histogram, quantization
- Week 2
  - Filtering: smoothing, median filtering, sharpening
  - Low level detection: Template matching, Edges, Line, Circles
- Week 3
  - Image Blending and Pyramids, Optical Flow

Image Blending

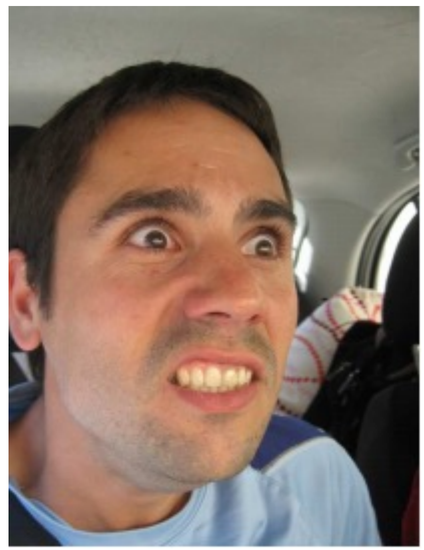
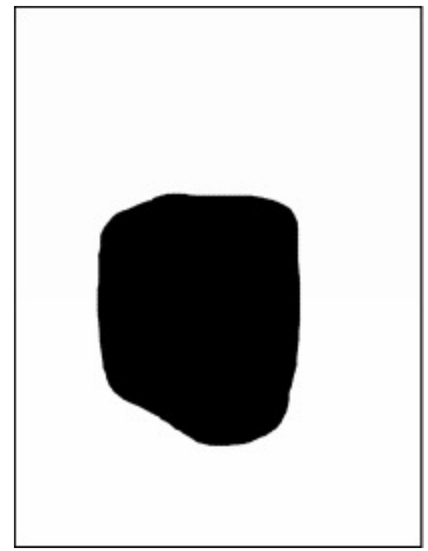
What is Blending?

Merge two image in a seamless way



What is Blending?

Blending: Input Images and Mask

Output: New Image

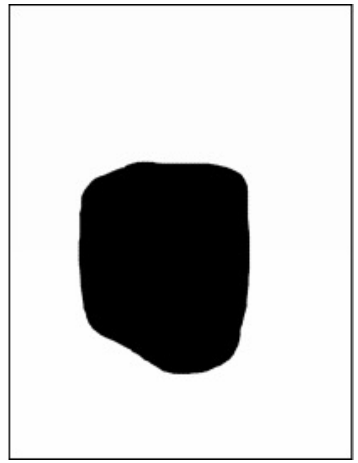
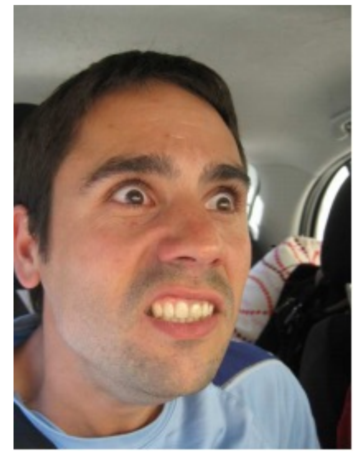
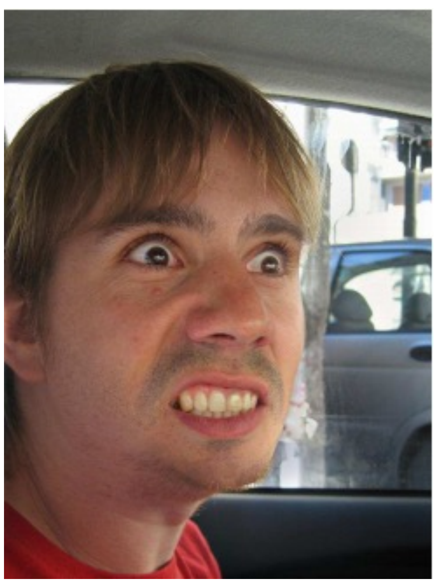
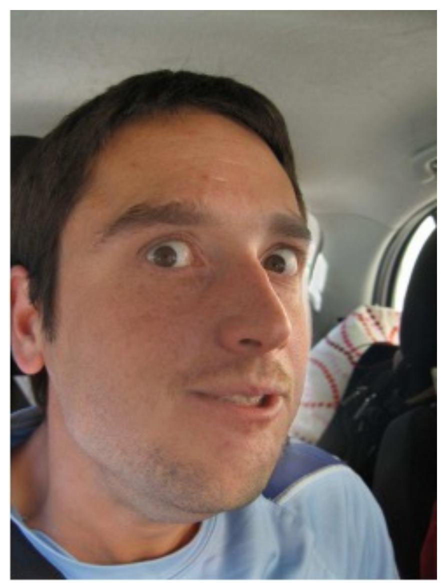






Image Blending

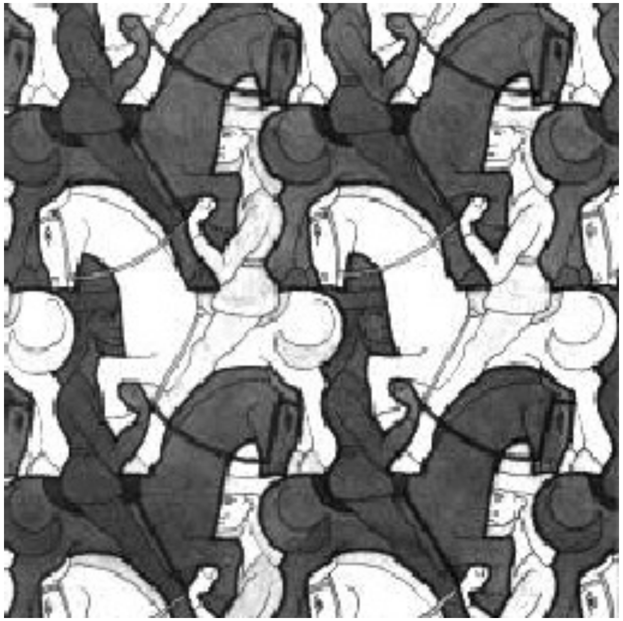
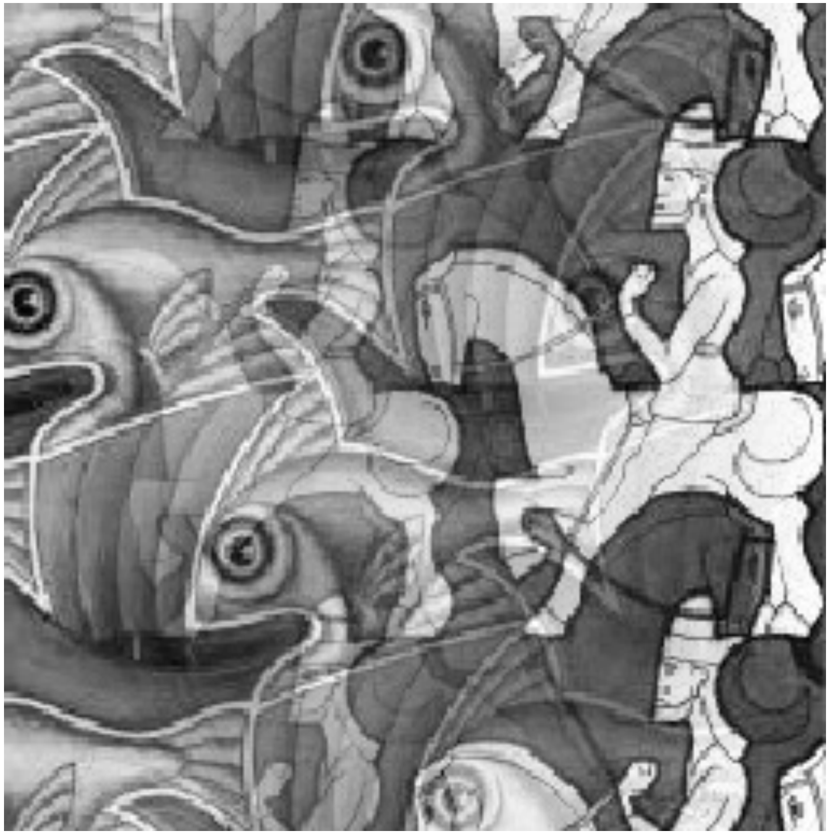
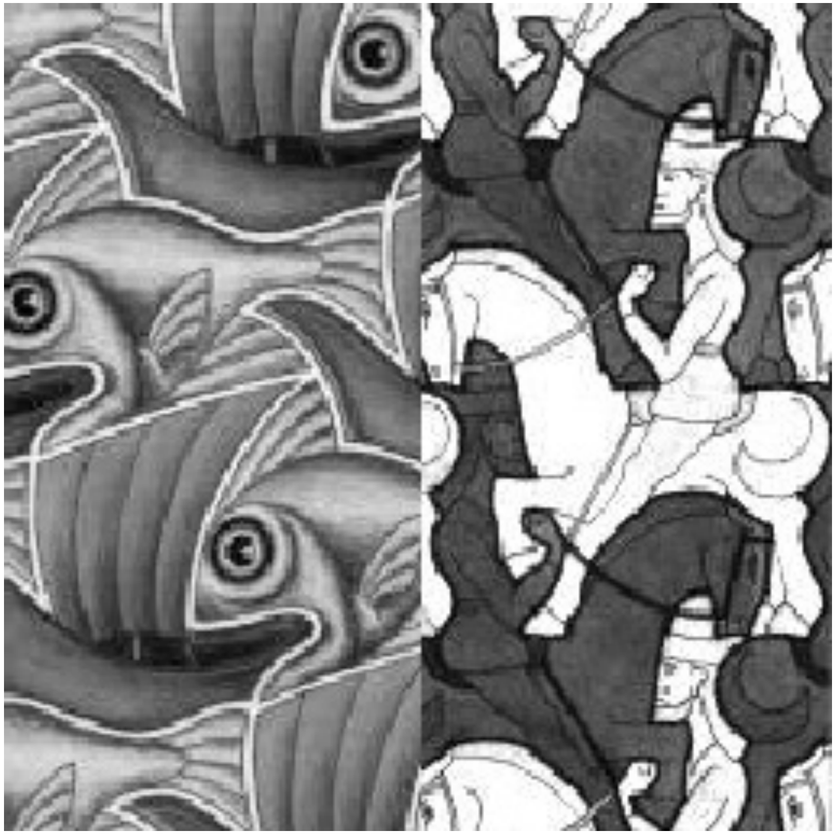


Image Blending: Challenges

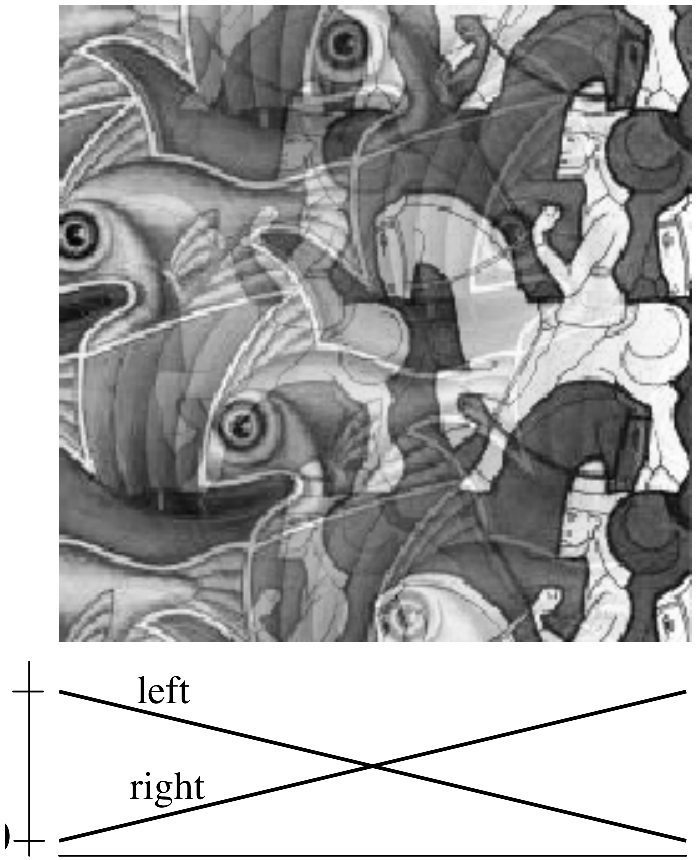


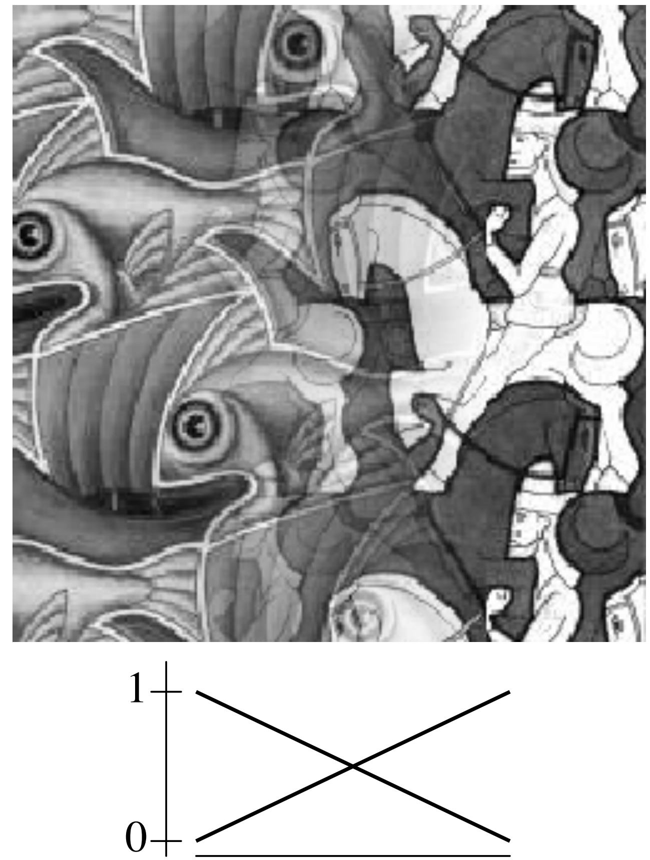
Ghosting



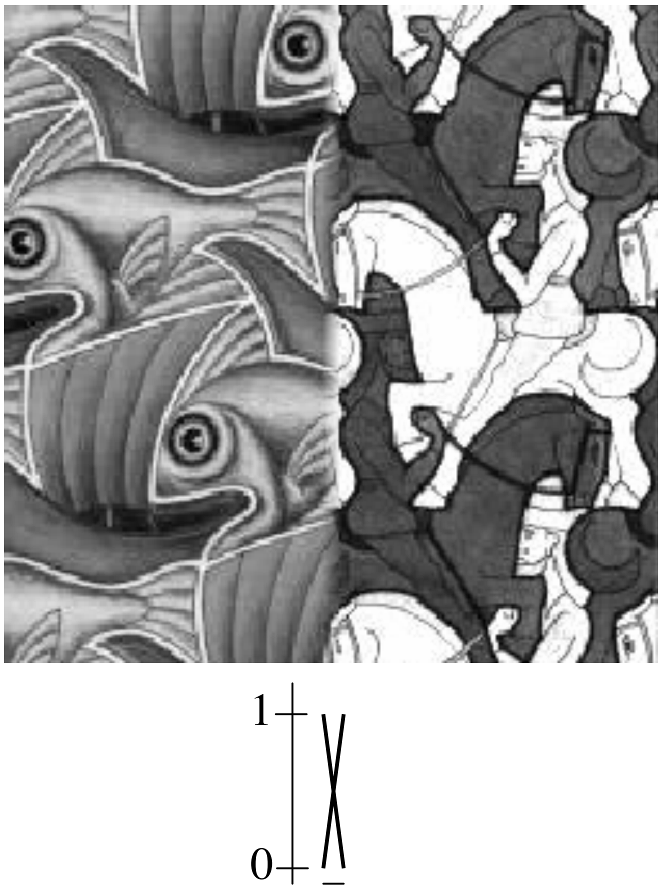
Visible Seams

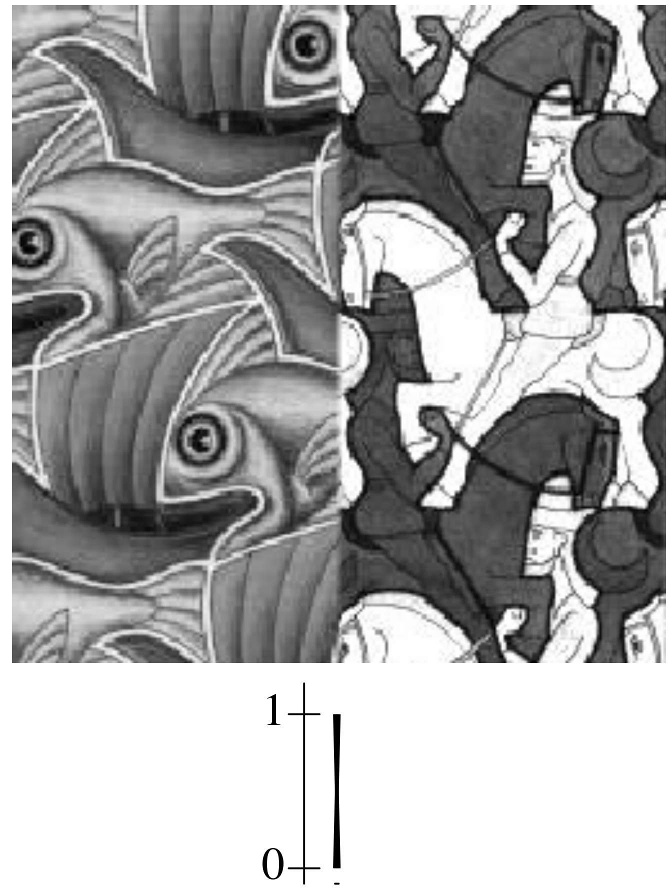
Ghosting: Window size



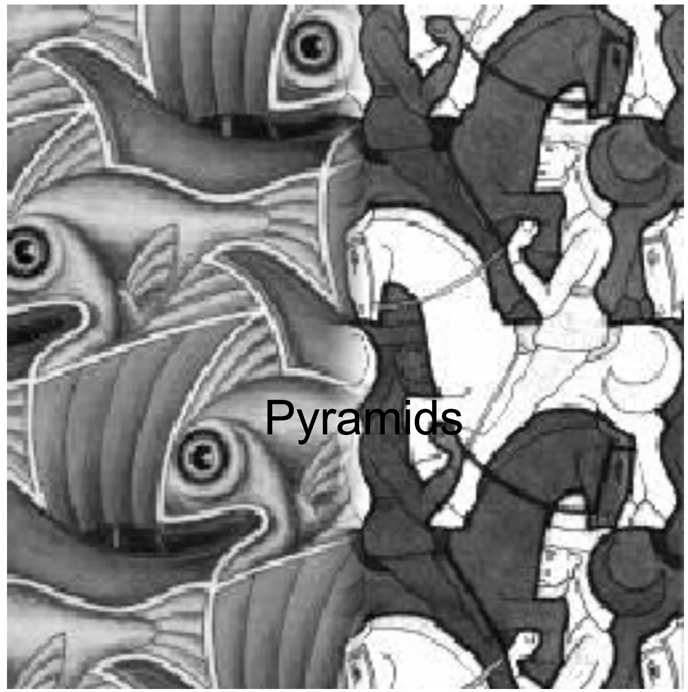


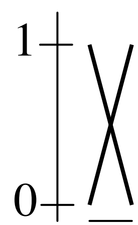
Visible Seam: Window Size





Optimal Window Size





Optimal Window Size

- To avoid seams
  - window >= size of largest prominent feature
- To avoid ghosting
  - window <= 2*size of smallest prominent feature
- Depends on the changes (frequencies) in the image
  - hard to find
- We will use <u>image pyramids</u> to avoid the need to find window size

Image Pyramids

What is Blending?

Merge two image in a seamless way

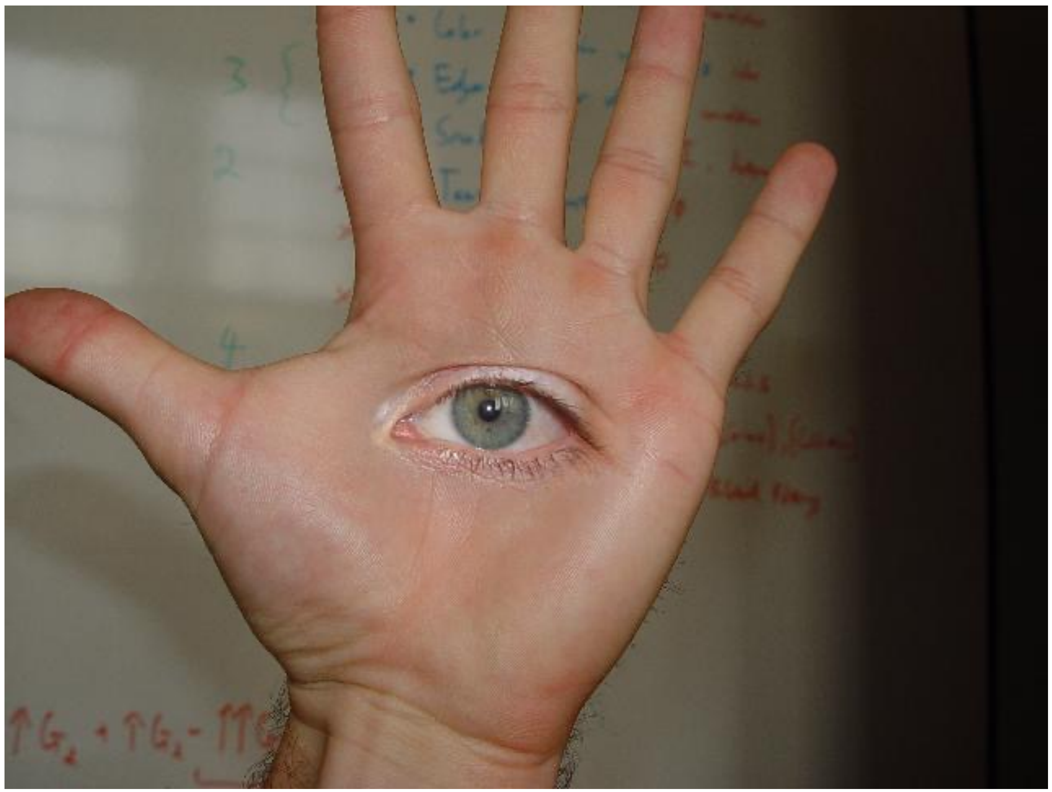
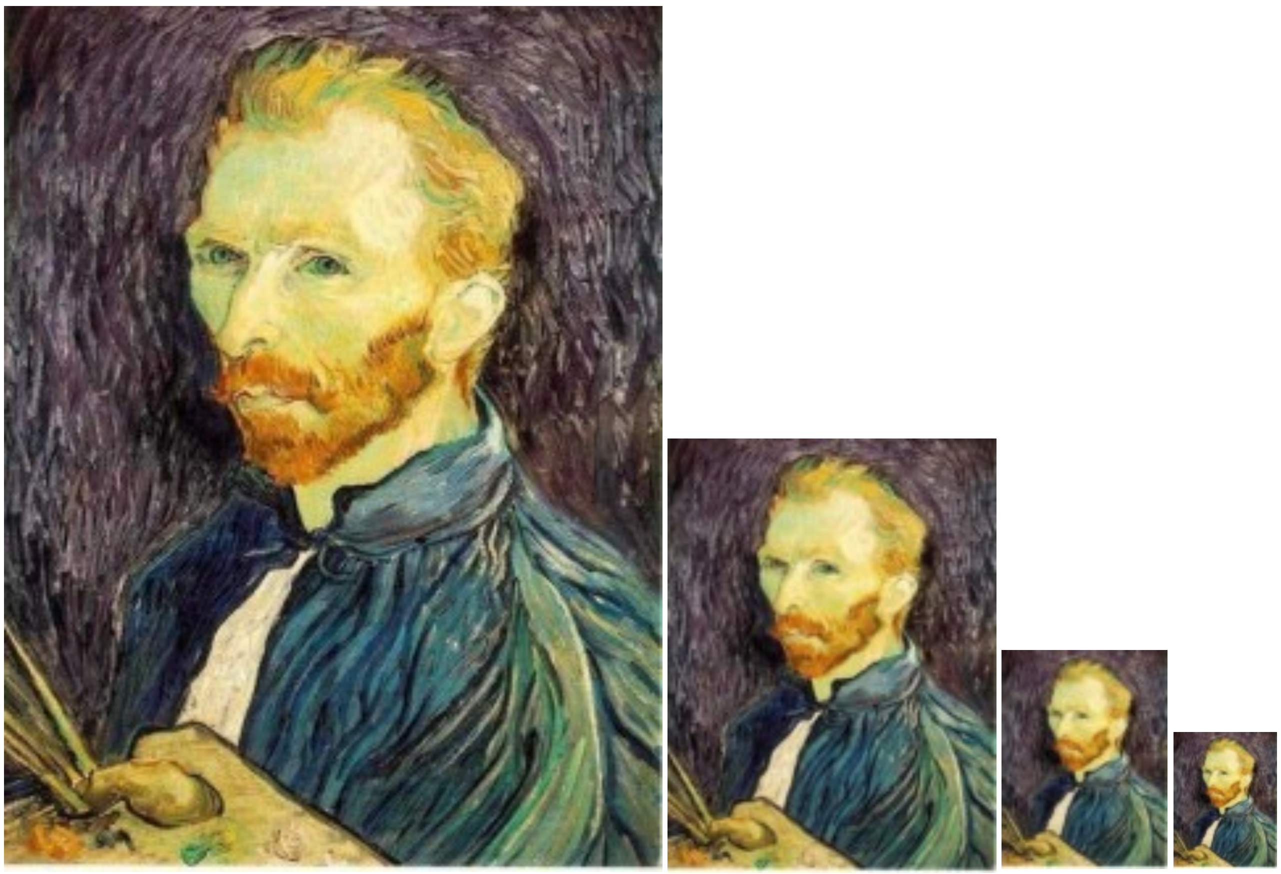


Image Pyramids

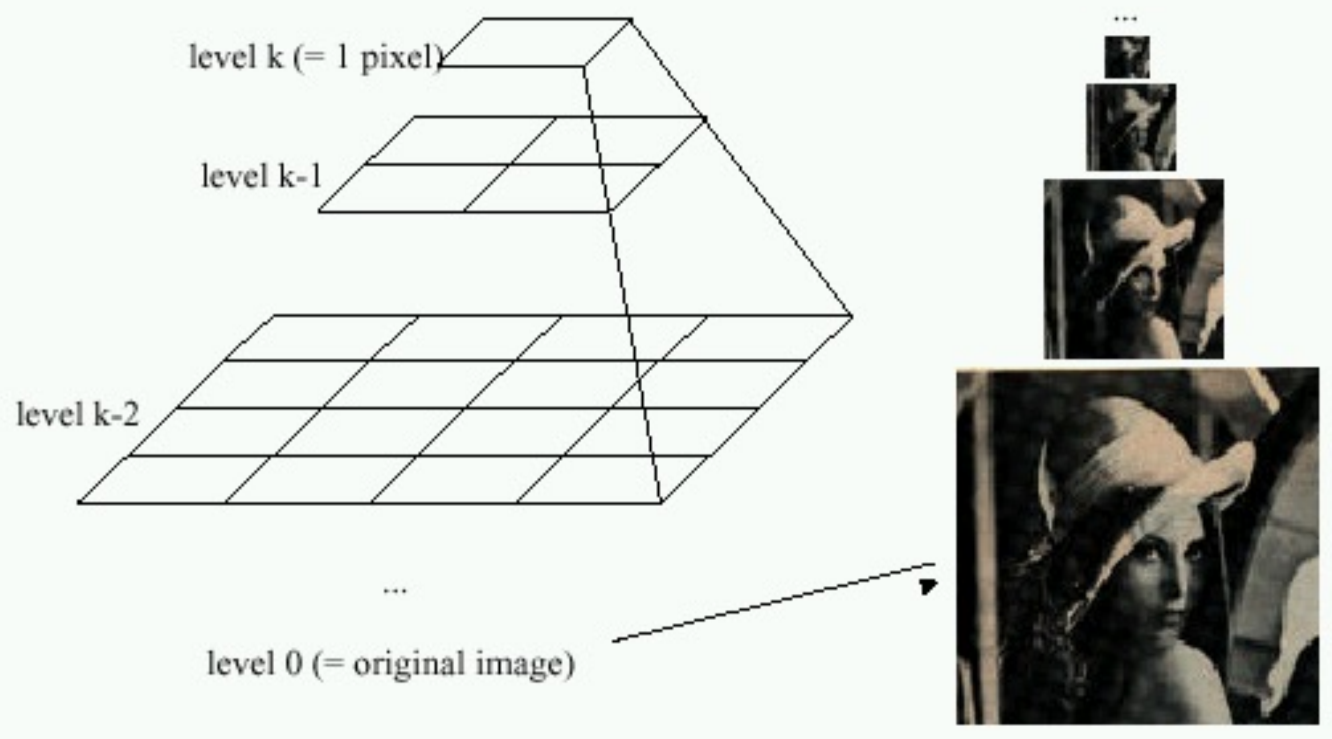
- Pyramid of an Image: A collection of the same image, reduced size by half at each level



Size:1x1 Size:1/2x1/2 1/4x1/4 1/8x1/8

Why Pyramids?

Idea: Represent NxN image as a “pyramid” of $1x1, 2x2, 4x4, \ldots, 2^k x 2^k$ images (assuming $N=2^k$)

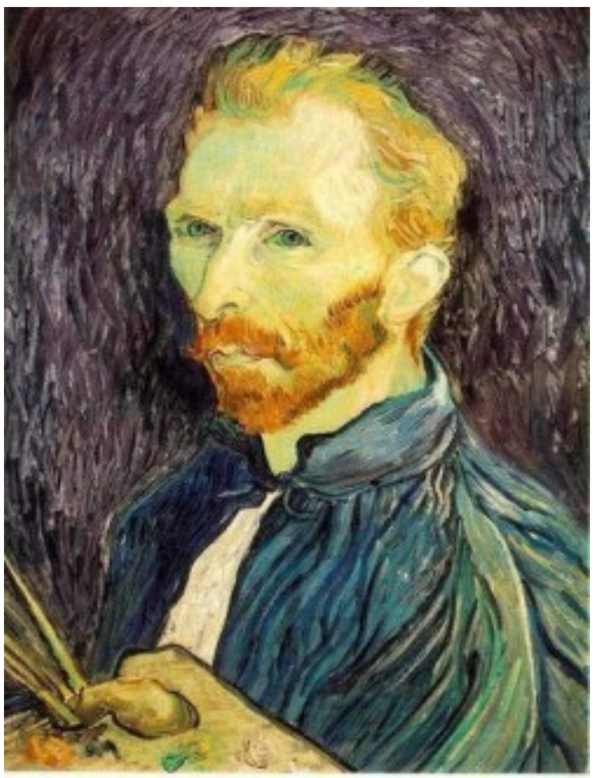


Gaussian Pyramid [Burt and Adelson, 1983]

Image Pyramids: How?

- Naïve Solution?

Subsampling: take the 2nd pixel from each row/col

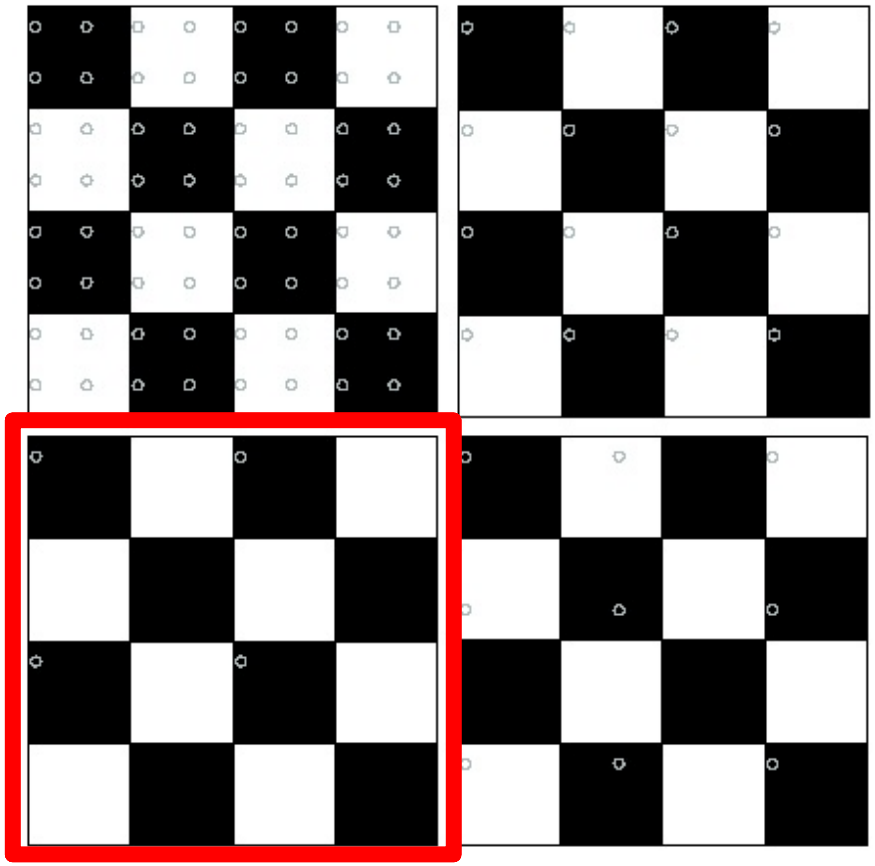


Size:1x1

Size:1/2x1/2

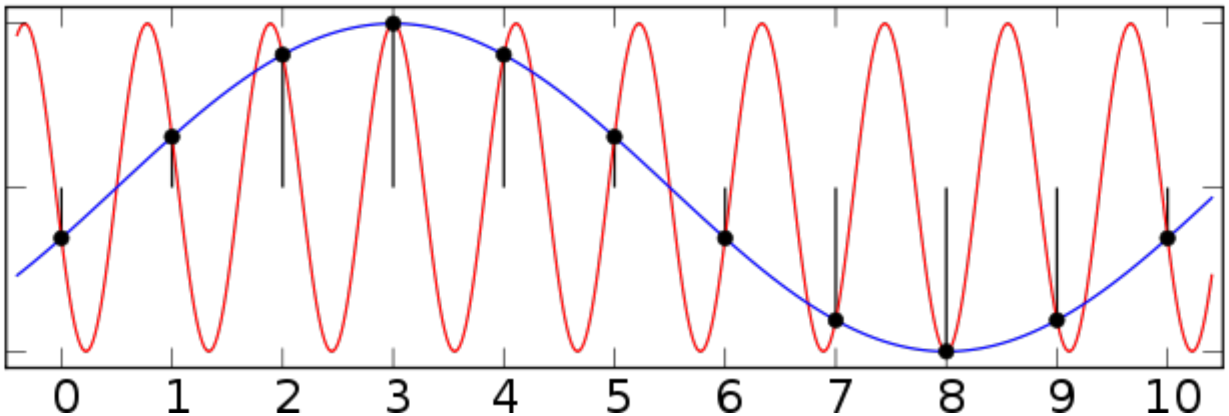
Image Pyramids: Sub-sampling

Simple subsampling results in aliasing



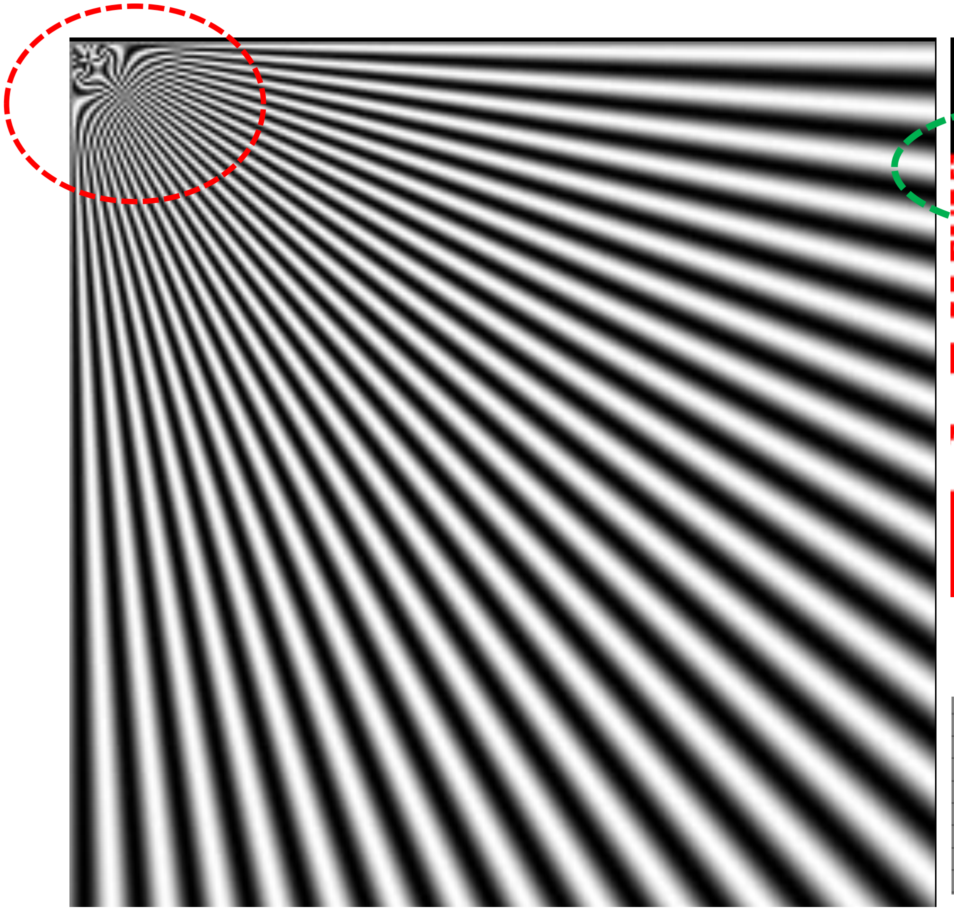
Forsyth and Ponce

Aliasing



- When sub-sample, the rate should be adjust to capture all the changes in the signal
- The sampling rate <u>at each level </u>in image pyramids is ½ x ½
- The sampling rate after 3 levels with respect to the original image is 1/8 x 1/8
- After several levels, aliasing is almost inevitable

Aliasing



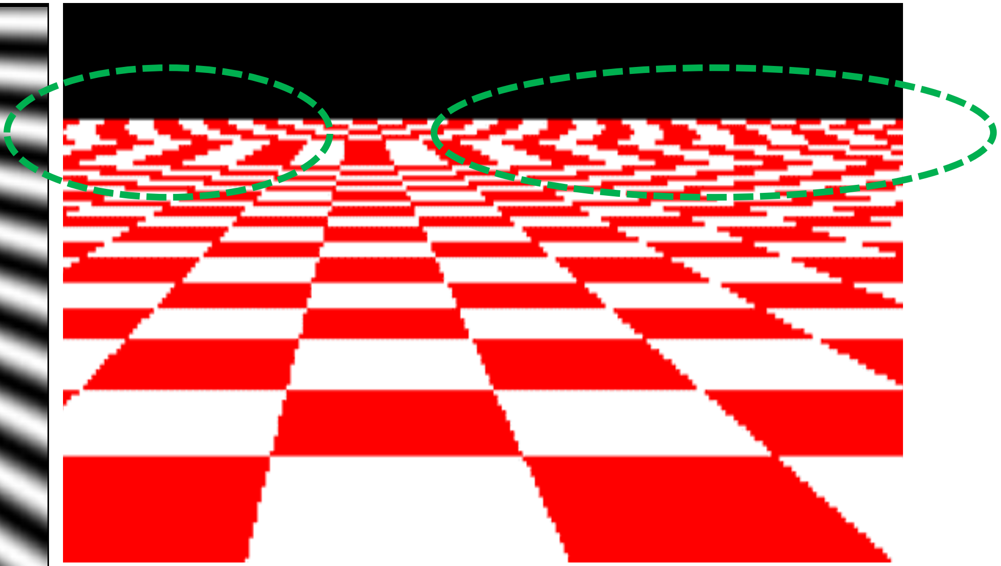
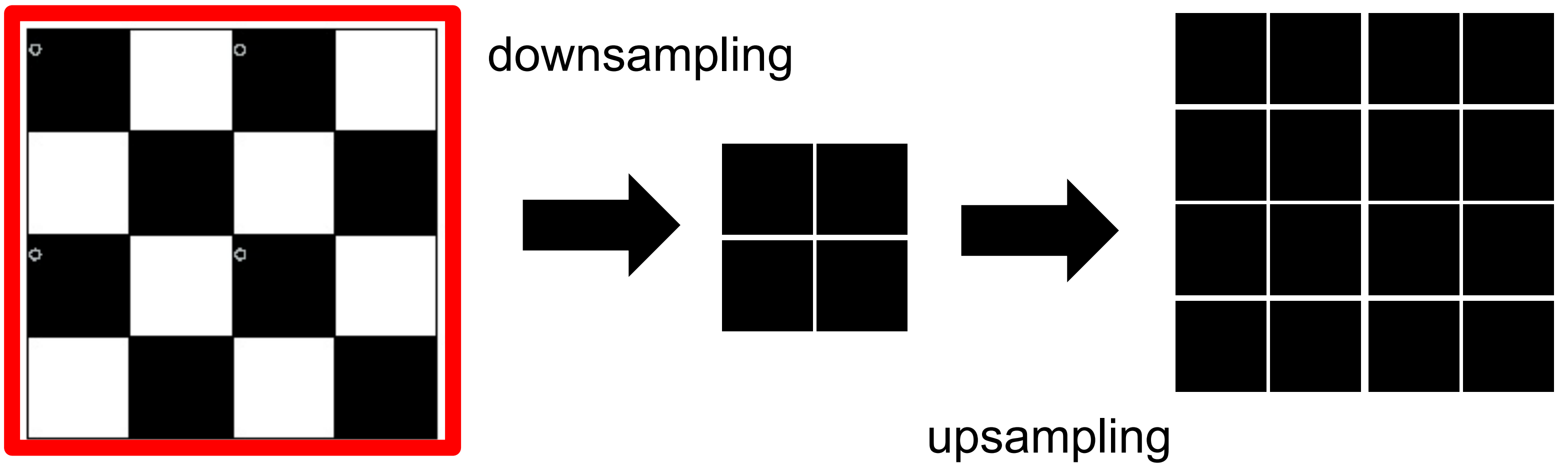


Image Pyramids: Sub-sampling

Aliasing means that we can not reconstruct something similar



Aliasing

Gaussian Pyramids

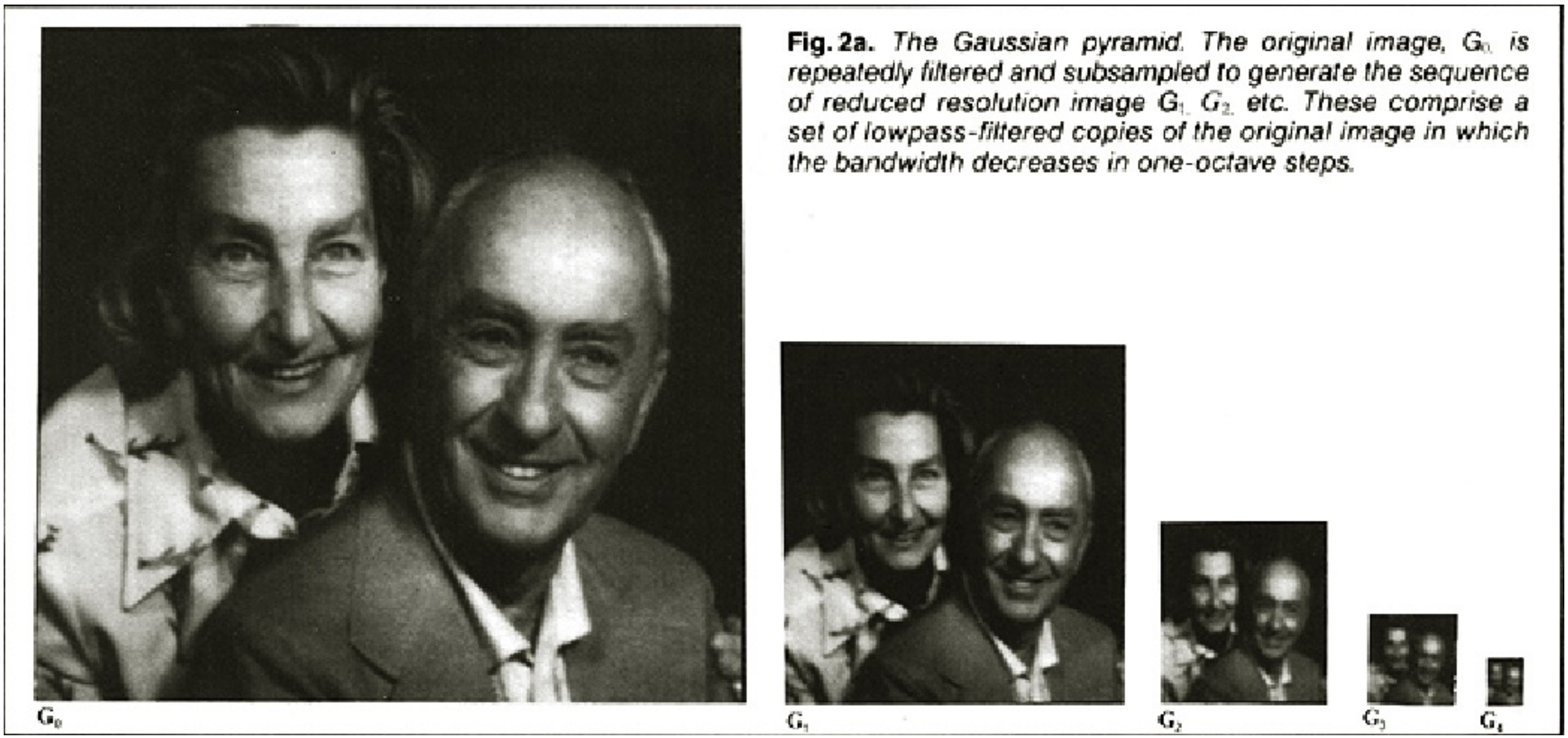


Fig. 2a. *The Gaussian pyramid. The original image, G_0, is repeatedly filtered and subsampled to generate the sequence of reduced resolution image G_1, G_2, etc. These comprise a set of lowpass-filtered copies of the original image in which the bandwidth decreases in one-octave steps.*

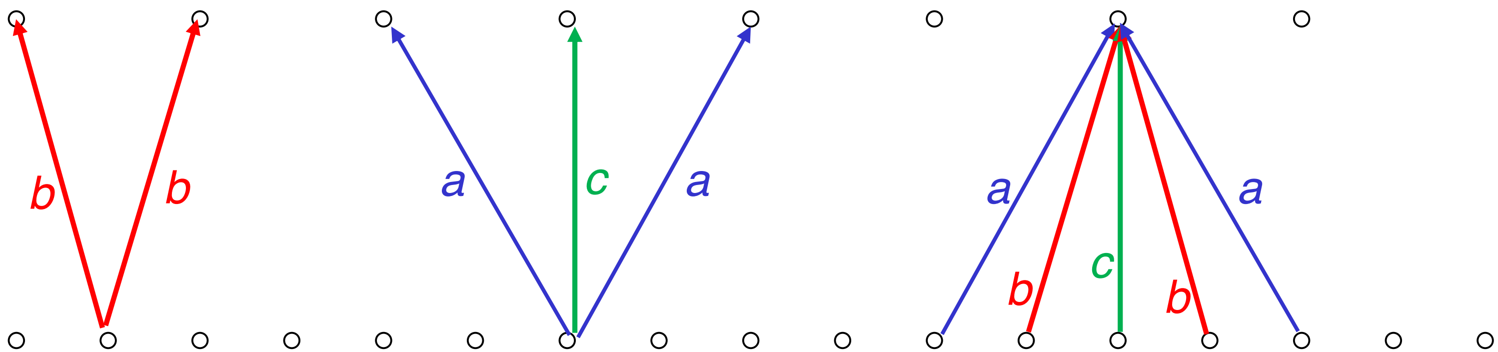
Gaussian Pyramids: How

- We define the Reduce operation
- Input: Image NxM, Output: Image N/2 x M/2
- The Reduce Operation:
  - Blur
    - Convolve with a 5×5 gaussian filter
  - Sub-sample
    - Select only every 2nd pixel in every 2nd raw

Gaussian Smoothing Kernel

The contribution of each pixel for the pixels in the next level

Example: 5 Weights: (a, b, c, b, a)



Conditions:

c>b>a (Unimodal)

2a+2b+c=1 (Receiving weights)

c+2a=2b (Contributing weights)

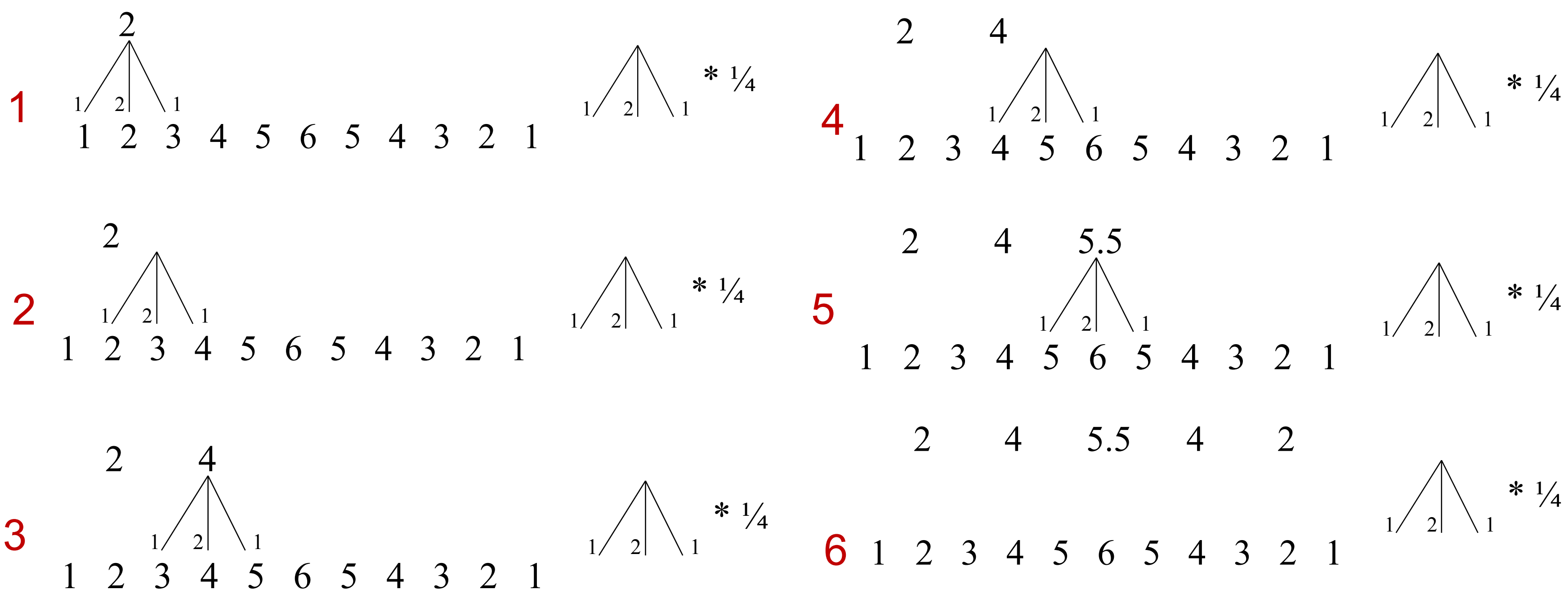
Commonly Used - Binomial Coefficients

1 2 1

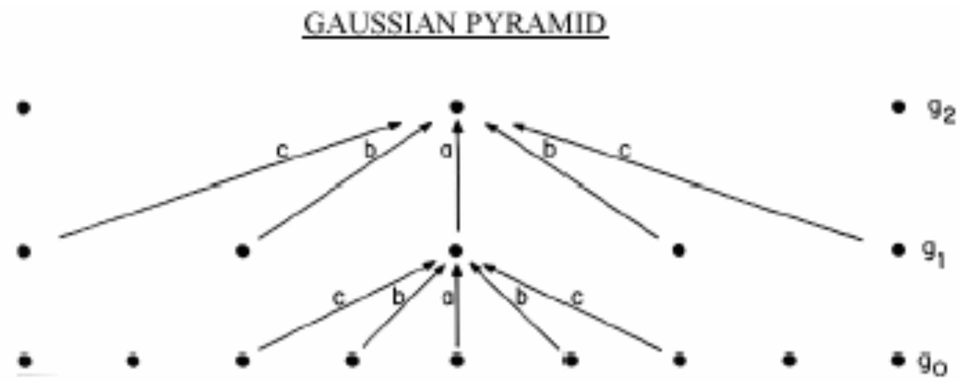
1 4 6 4 1

1 6 15 20 15 6 1

Reduce Operation



Pixel Contributions



g_0 = IMAGE

g_L = REDUCE $[g_{L-1}]$

Fig 1. A one-dimensional graphic representation of the process which generates a Gaussian pyramid Each row of dots represents nodes within a level of the pyramid. The value of each node in the zero level is just the gray level of a corresponding image pixel. The value of each node in a high level is the weighted average of node values in the next lower level. Note that node spacing doubles from level to level, while the same weighting pattern or “generating kernel" is used to generate all levels.

Computational Complexity

- Memory
  - NxN (1+1/4+1/16+...)= 4/3 * NxN
- Each level can be computed with a single convolution

Gaussian Pyramid

G0

G1=Reduce(G0)

G2=Reduce(G1)

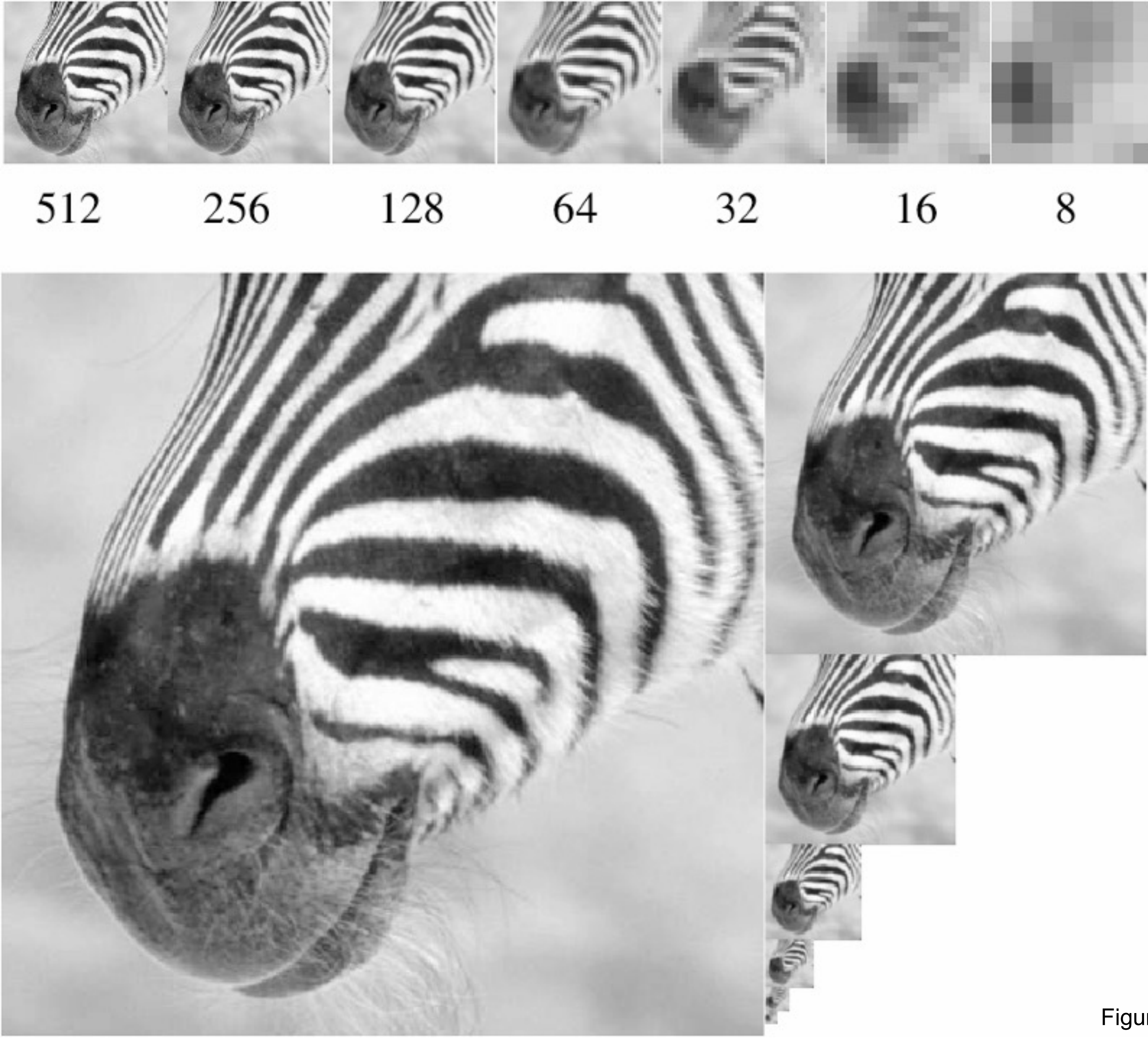


Figure from David Forsyth

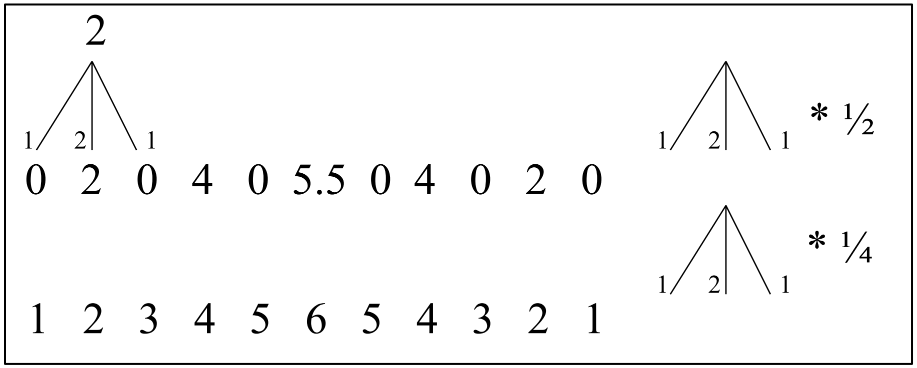
Expand

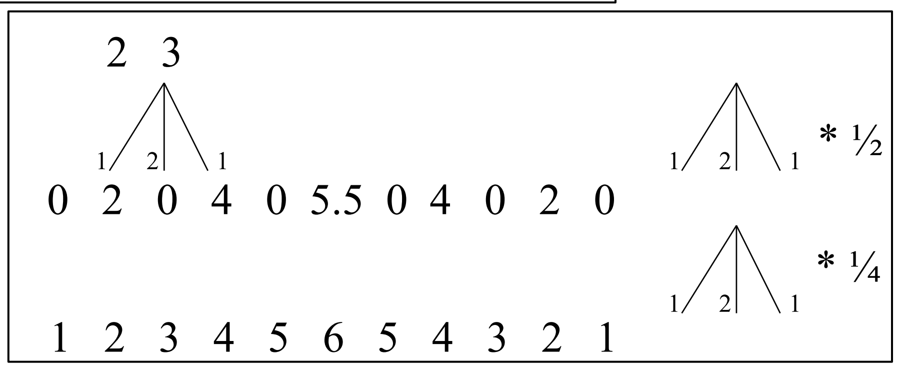
- Given a Pyramid level K, can we reconstruct level K-1?
- First problem: level k-1 is not the same size as level K
- The Expand Operation:
  - Zero Padding (a1, 0, a2, 0, a3, 0, . . .)
  - Blur

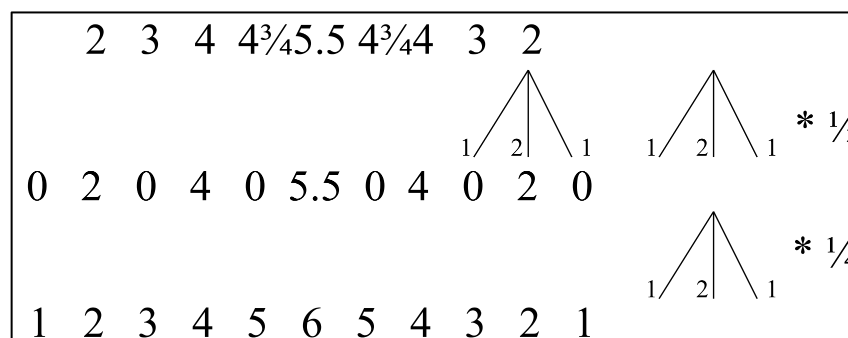
    - Note: Blur needs different normalizations!

    - What is zero padding followed by blur with $\frac{1}{2}(1,2,1)$

Reconstruct: Expand







Reduce and Expand

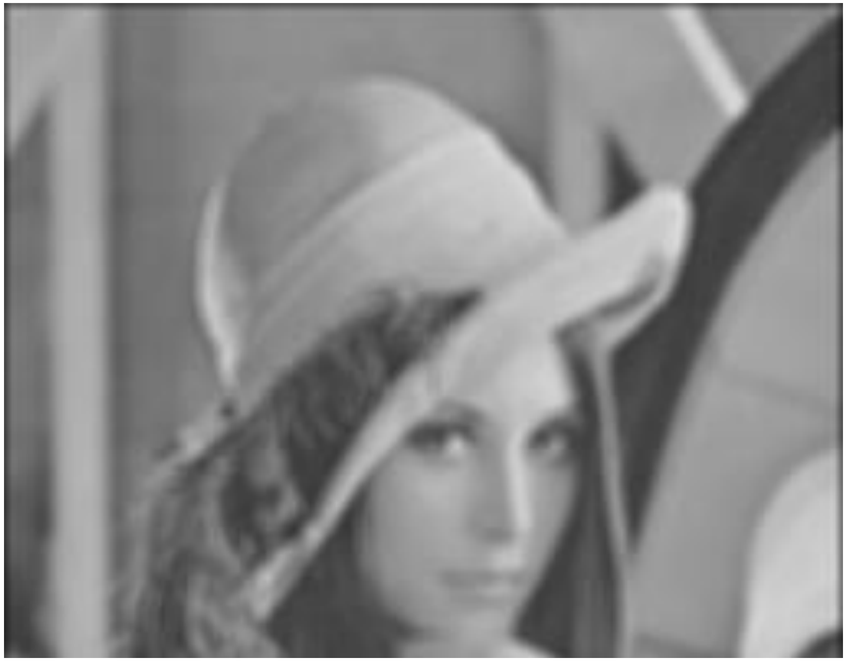
G0

G1
Reduce(G0)

Expand(G1)

Pyramid Level Difference

G0



Expand(G1)

Pyramid Level Difference

- L0=G0 – Expand(G1)

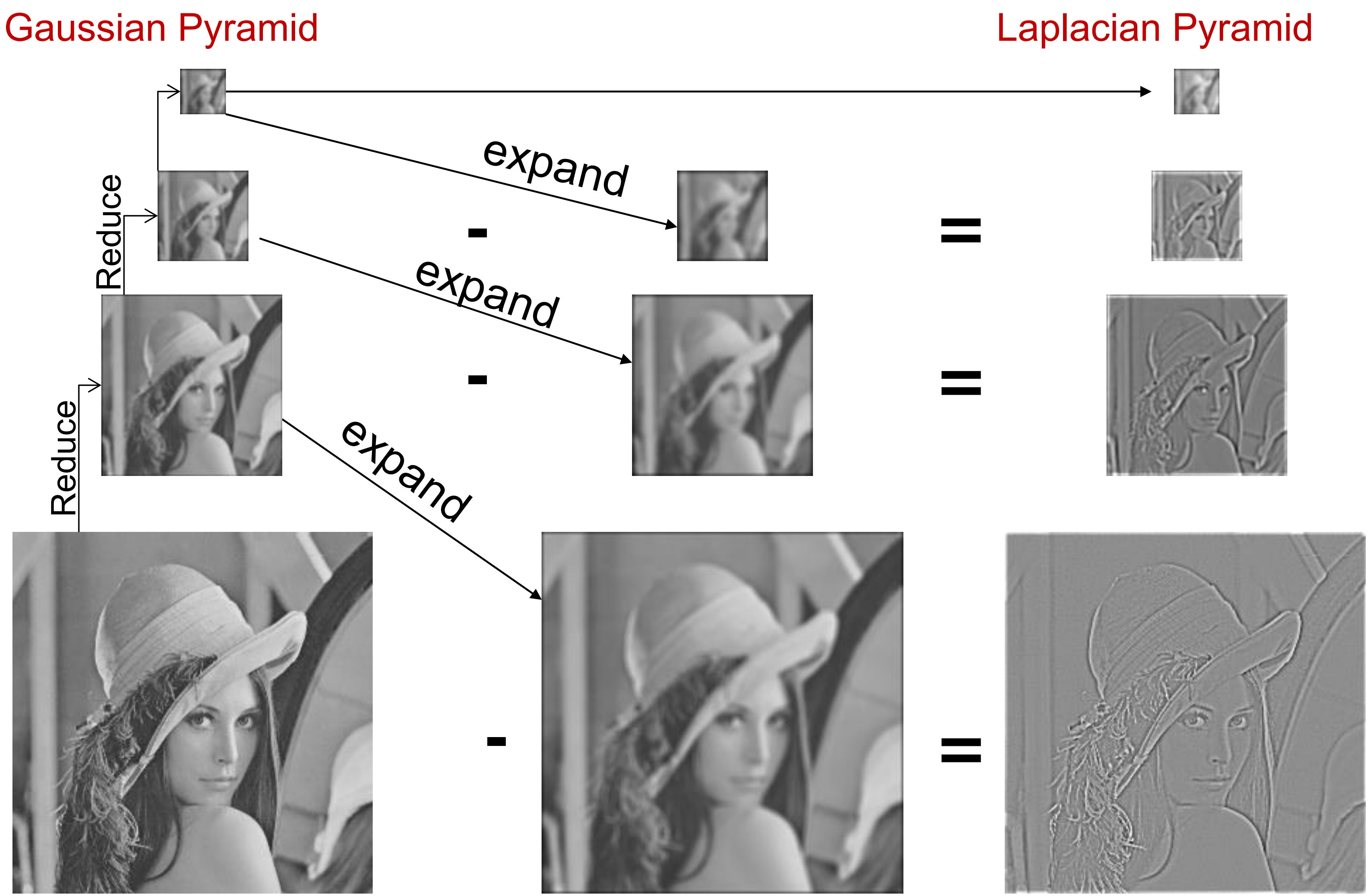
Reconstruction

- The difference stores all the details that were lost by the blurring and upsampling operations

=

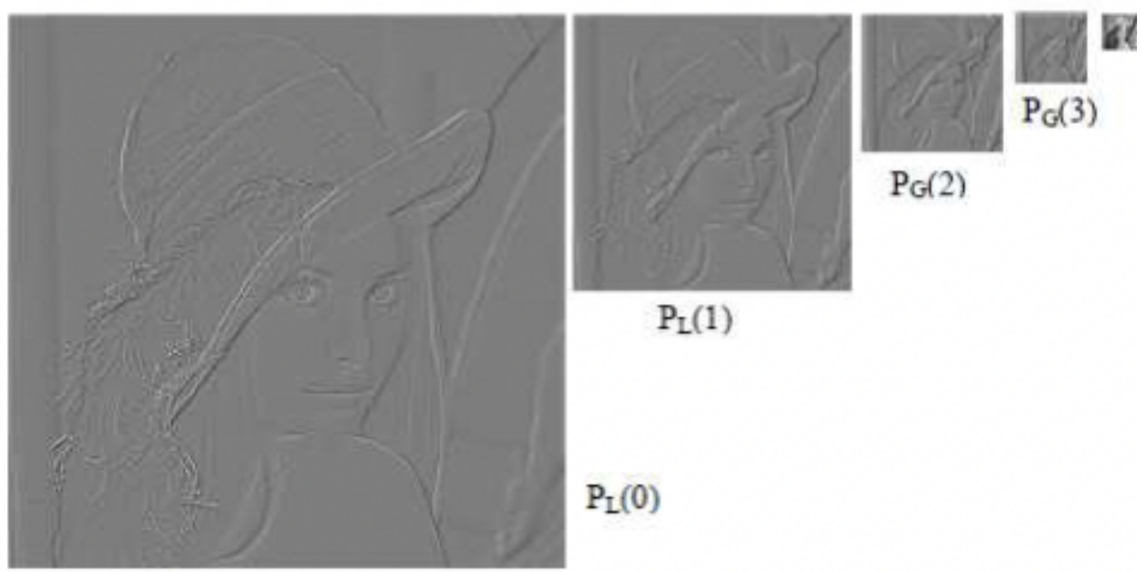
+

Gaussian and Laplacian Pyramid



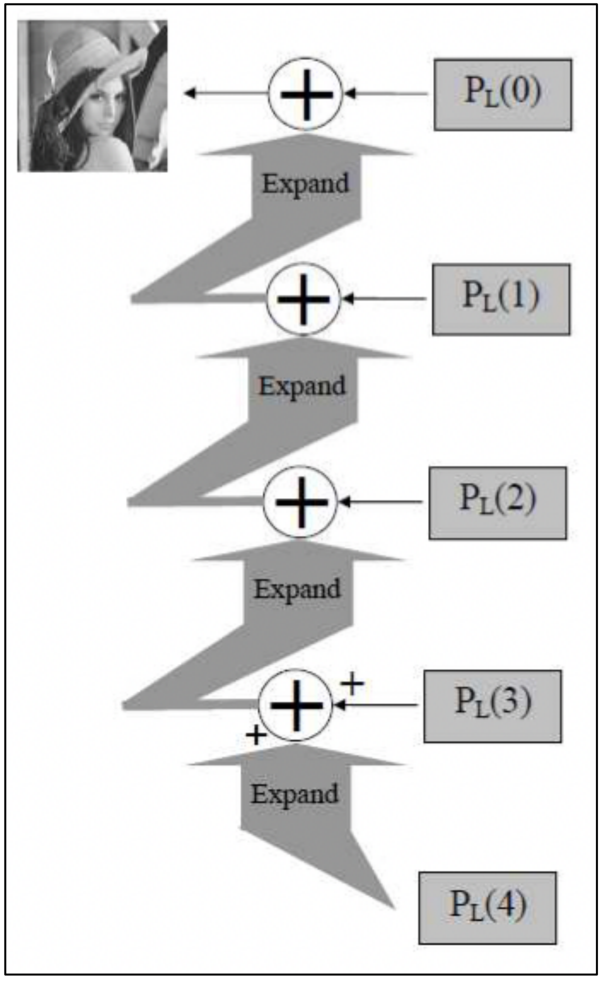
Reconstruction: Laplacian and Gaussian pyramids

- The last level in both Laplacian and Gaussian are equal
  - So we can reconstruct the original image
  - How?



Reconstruction

We can reconstruct the original image by
Storing only the Laplacian pyramid, which
Is more compressed than the original image



Credit: Shai Avidan

Recall:Optimal Window Size

- To avoid seams
  - window >= size of largest prominent feature
- To avoid ghosting
  - window <= 2*size of smallest prominent feature

Depends on the changes (frequencies) in the image, hard to find

Pyramid to rescue

- The key idea
  - For a given image, we do not know what are the frequencies in the image
  - But for the pyramid, we can approximate the optimal window size

Pyramid Blending Arbitrary Shape

- Given two images *A* and *B*, and a binary image mask *M*
- Construct Laplacian Pyramids L_a and L_b
- Construct a Gaussian Pyramid G_m
- Create a third Laplacian Pyramid L_c where for each level *k*

$$L_c(i,j) = G_m(i,j)L_a(i,j) + (1 - G_m(i,j))L_b(i,j)$$

- Reconstruct all levels L_c in to get the blended image

Input

Output