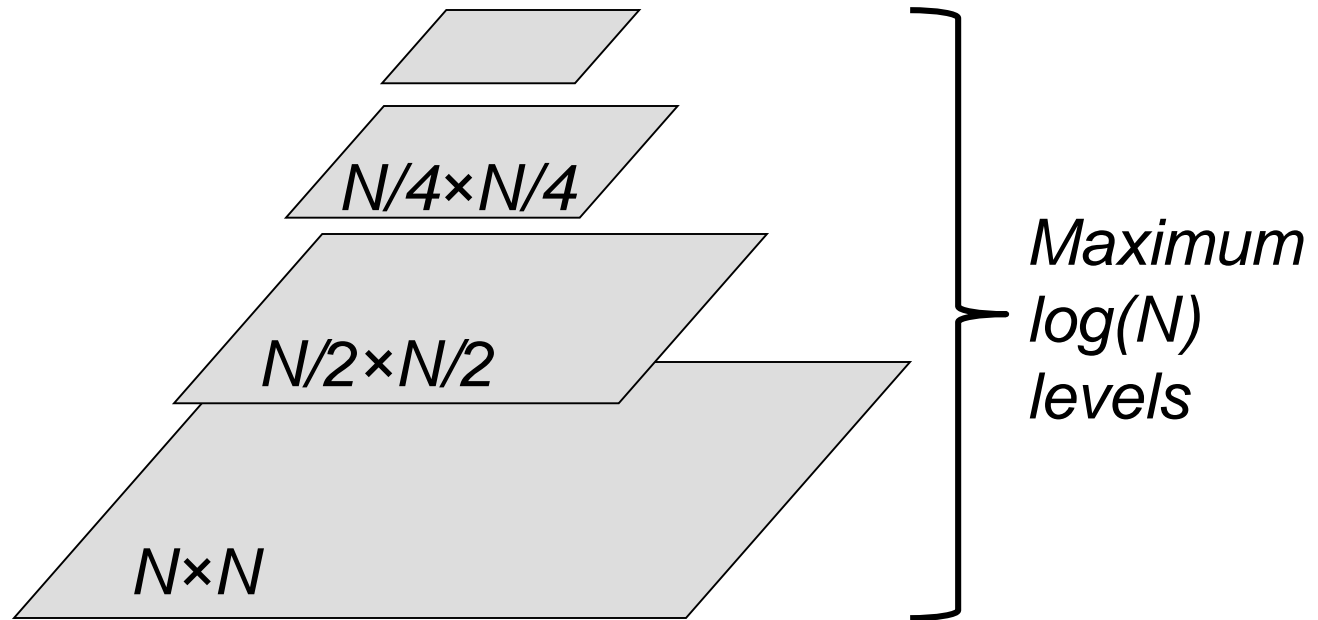
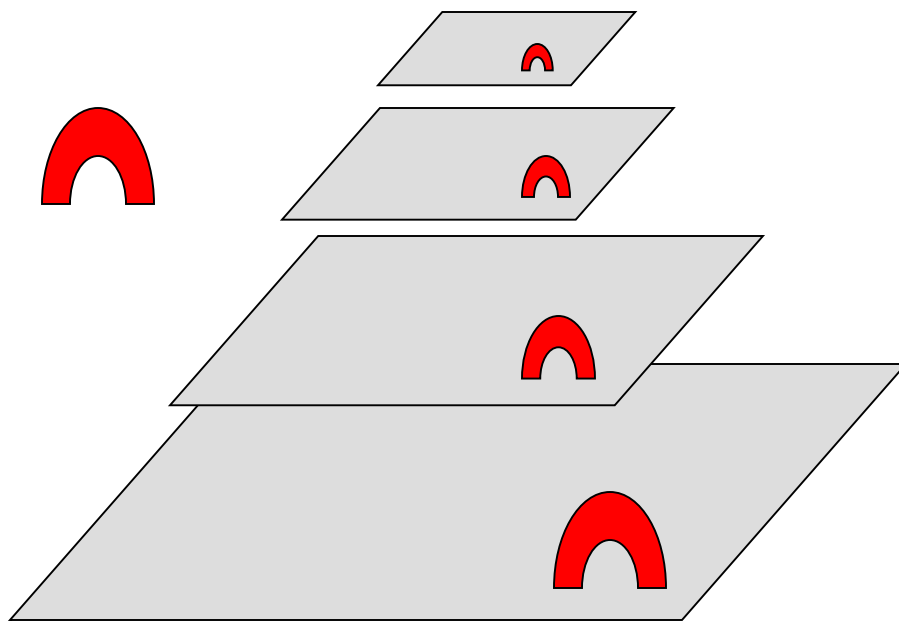


Image Pyramids



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

Efficient Visual Search



Search Area	Pattern Size	Total Oper.
± 16	4^2	2^{14}
± 32	8^2	2^{18}
± 64	16^2	2^{22}
± 128	32^2	2^{26}
$2^8 2^8$	$2^5 2^5$	$= 2^{26}$

- Pyramids: Start the search in a small image
- search area is small in larger levels (e.g. ± 1) using the estimate from the smaller level

Applications for Pyramids

- Detection and Search (Esp. huge images)
- Browsing in Image Databases
- Motion Computation

Google Earth - Products - Mozilla Firefox

http://earth.google.com/earth.html

Open WebMail Getting Started Yahoo! Mail - yaelb... bezeq144 Impr 10bis.co.il

copernic Search Web Search Desktop: Emails

google earth

Google Earth

A 3D interface to the planet

Google Earth Home
Downloads
Products
Industries
Support

Product quick links

- Google Earth
- Google Earth Plus
- Google Earth Pro
- Google Earth Enterprise
- Google Earth Fusion
- Google Earth Server
- Google Earth EC

Google Earth (beta)

"For anyone who has ever dreamed of flying..." - NY Times

The idea is simple. It's a globe that sits inside your PC. You point and zoom to anyplace on the planet that you want to see images and local facts zoom into view. Tap into Google search to show local points of interest and facts. Zoom to a specific check out an apartment or hotel. View driving directions and even fly along your route. We invite you to try it now.

Google Earth is free for personal use. No registration is required. You may (optionally) choose to upgrade to Google Earth

Features:

- Free for personal use.
- Sophisticated streaming technology delivers the data to you as you need it.
- Imagery and 3D data depict the entire earth - Terabytes of aerial and satellite imagery depict cities around the world in detail.

Done

Proxy: www.proxy.cs.huji.ac.il



Camera 1 -

13 events (00:00:20)



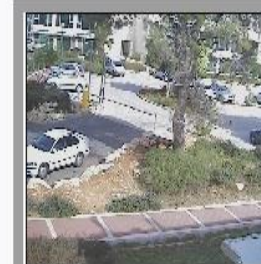
Camera 2 -

16 events (00:00:22)



Camera 3

77 events (00:01:24)



Camera 4

5 events (00:00:03)



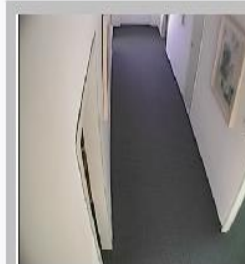
Camera 5

11 events (00:00:15)



Camera 6

3 events (00:00:15)



Camera 7

8 events (00:00:14)



Camera 8

30 events (00:00:46)

Image Resizing

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Reduce:

1. Blur

- E.g. Convolve with a 3×3 filter $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4}) \times (\frac{1}{4}, \frac{1}{2}, \frac{1}{4})^T$,
or a 5×5 filter $\frac{1}{16}(1, 4, 6, 4, 1) \times \dots$ or larger

2. Sub-sample

- Select only every 2nd pixel in every 2nd row

Expand:

1. Zero Padding ($a_1, 0, a_2, 0, a_3, 0, \dots$)

2. Blur

- Note: Expand blur needs different normalizations!
- Is zero padding followed by blur with $(\frac{1}{2}, 1, \frac{1}{2})$ OK?

Blur Kernels

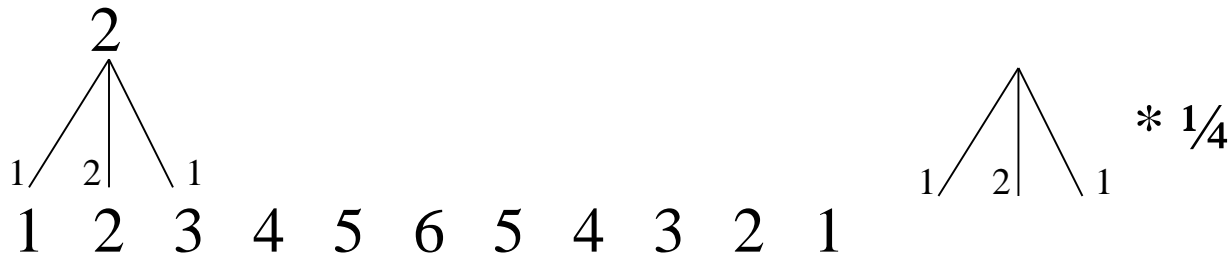
Commonly Used - Binomial Coefficients

- Odd number of coefficients (have a center)
- Sum of coefficients normalized to 1
- Fast to compute (using shift and add)
- Asymptotically similar to a Gaussian

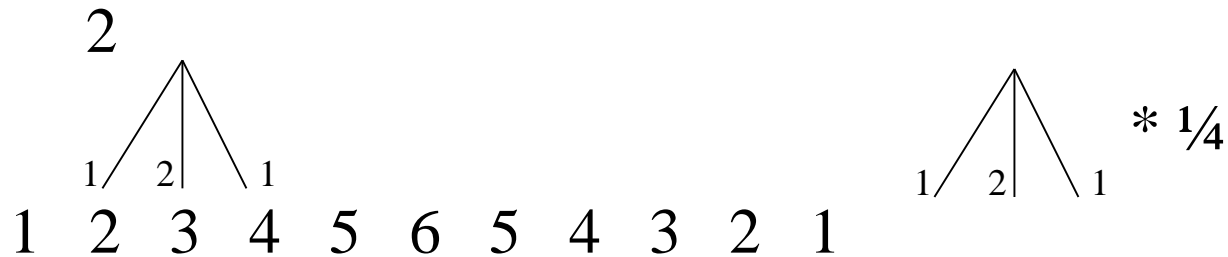
$$\begin{array}{ccccccccc} & & & 1 & 2 & 1 & & & / 4 \\ & & 1 & 4 & 6 & 4 & 1 & & / 16 \\ 1 & 6 & 15 & 20 & 15 & 6 & 1 & & / 64 \end{array}$$

$$(1 \ 1) * \dots^{2k} \dots * (1 \ 1) / 2^{2k}$$

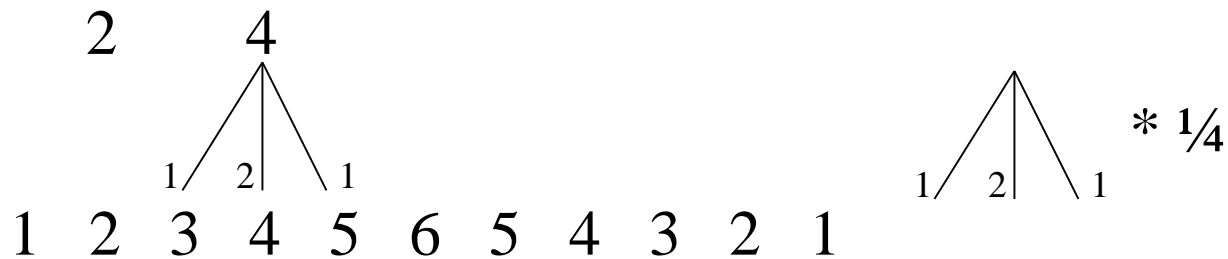
Blur & Sub-sample (*Reduce*)



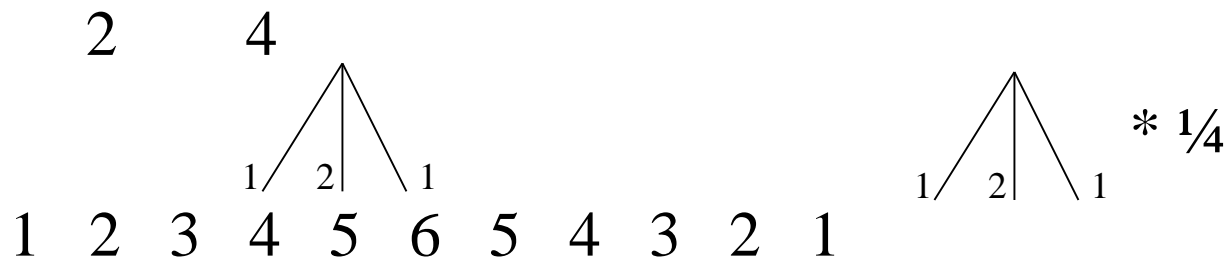
Blur & Sub-sample



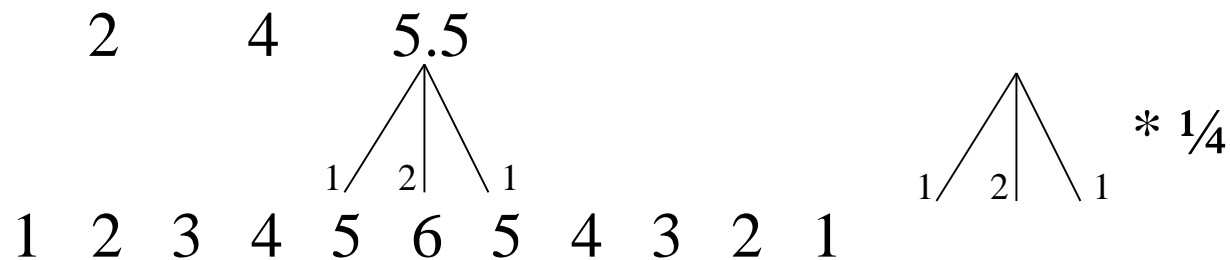
Blur & Sub-sample



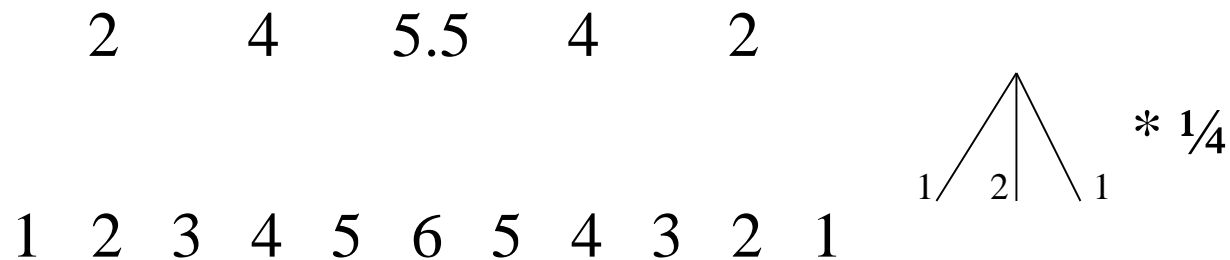
Blur & Sub-sample



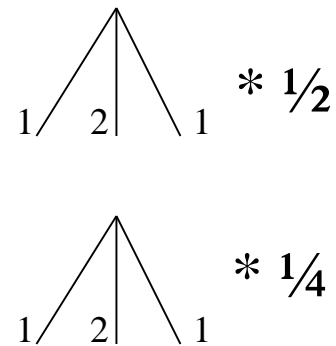
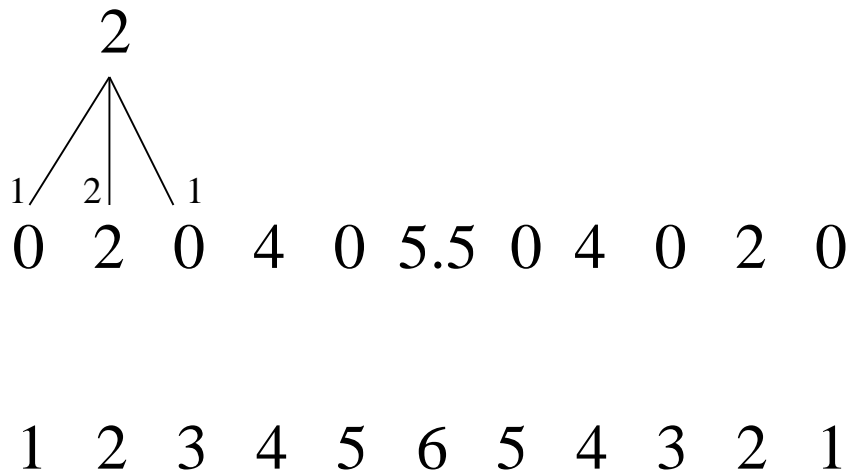
Blur & Sub-sample



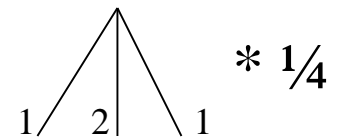
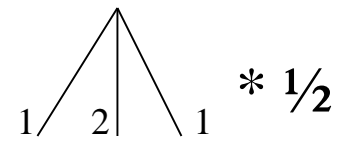
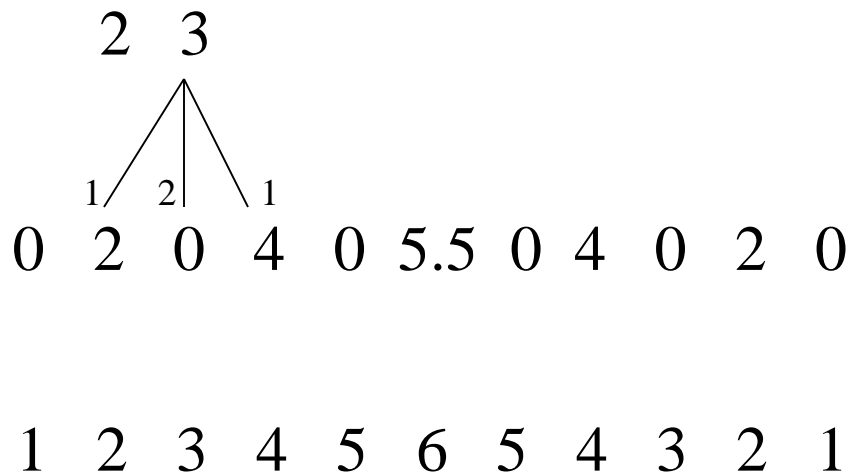
Blur & Sub-sample



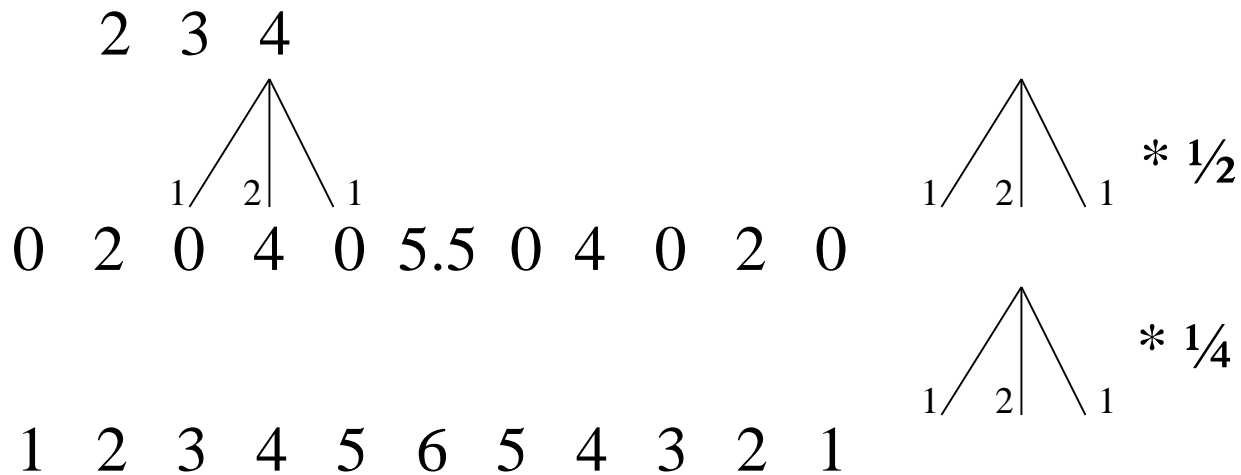
Zero-Pad & Blur (Expand)



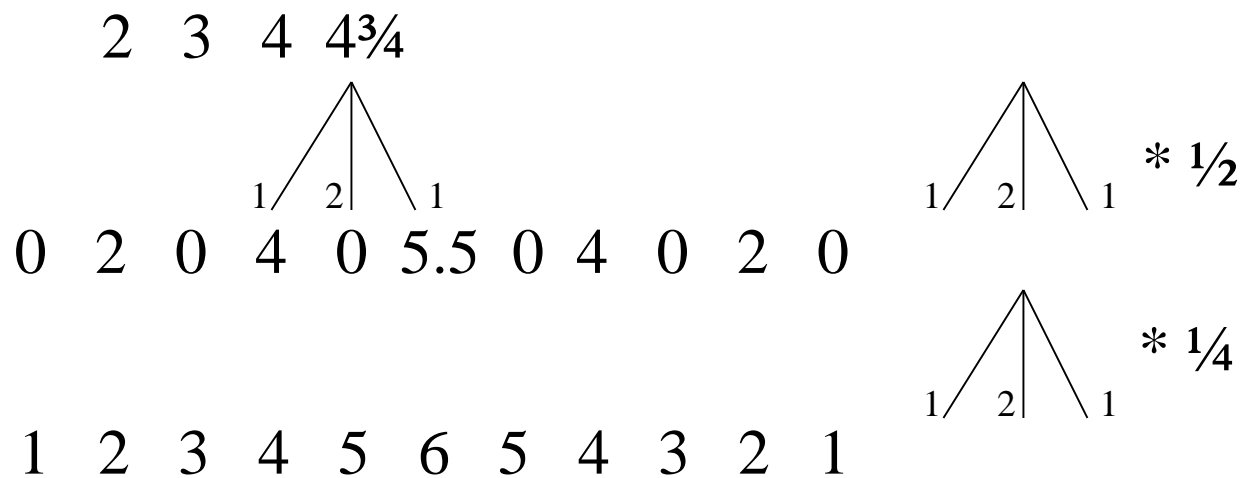
Zero-Pad & Blur



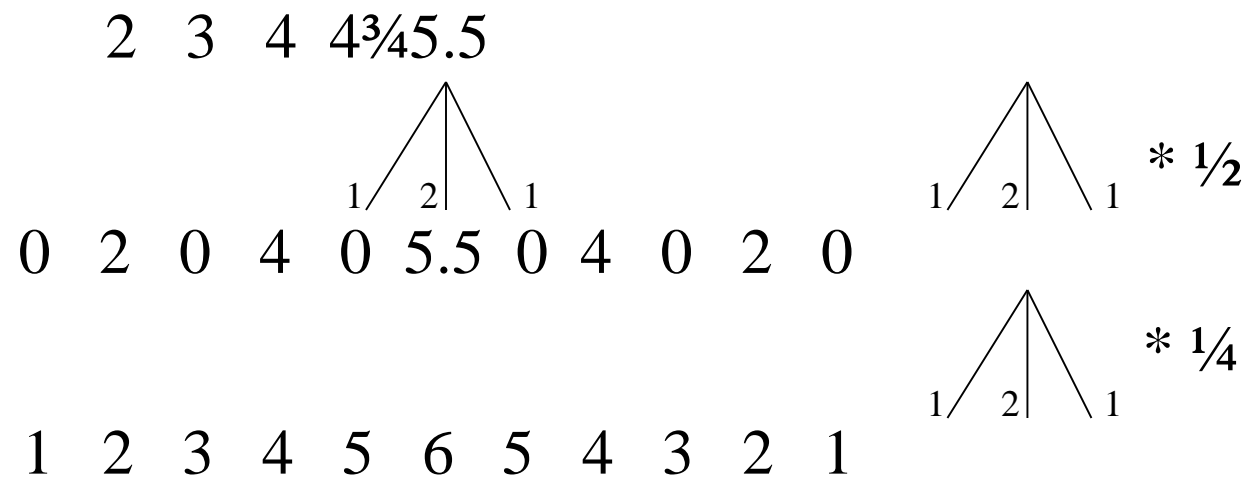
Zero-Pad & Blur



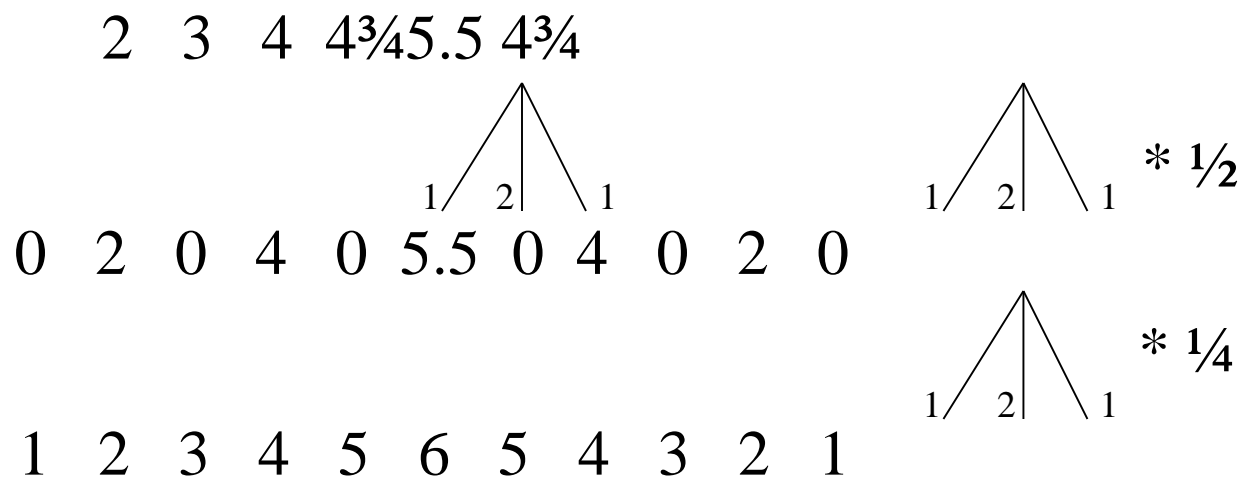
Zero-Pad & Blur



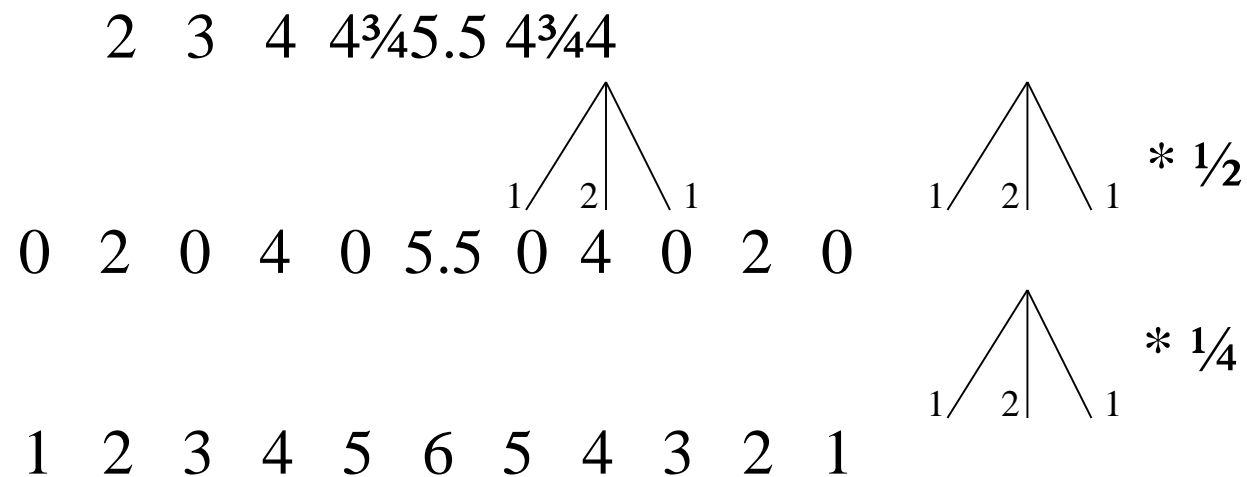
Zero-Pad & Blur



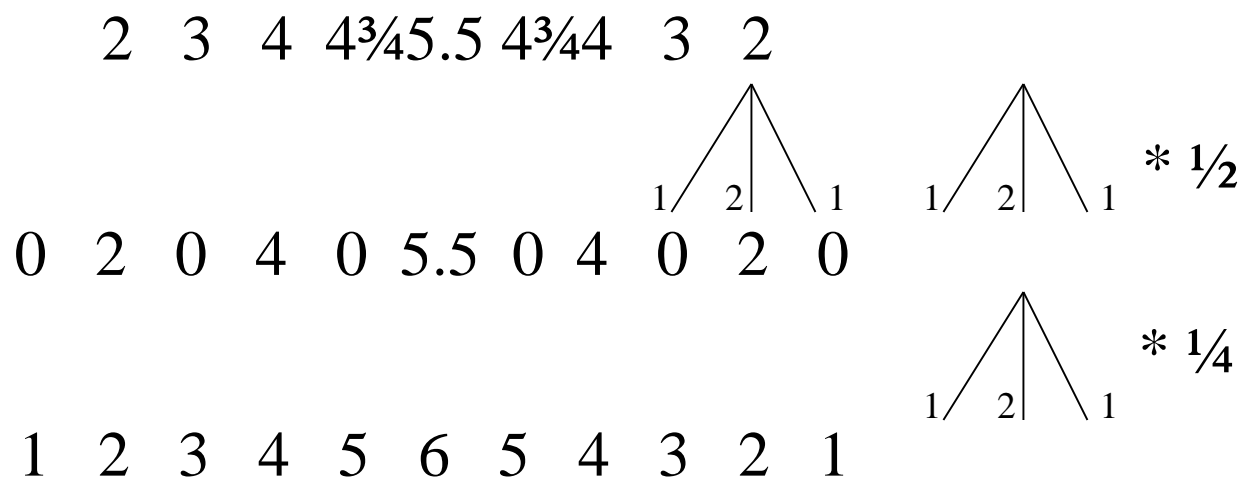
Zero-Pad & Blur



Zero-Pad & Blur

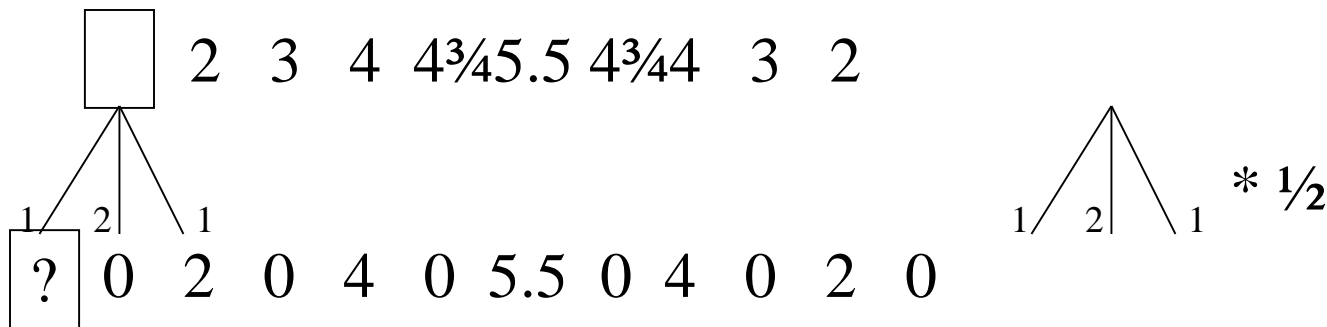


Zero-Pad & Blur



Handling Image Boundaries

Never Cyclic...

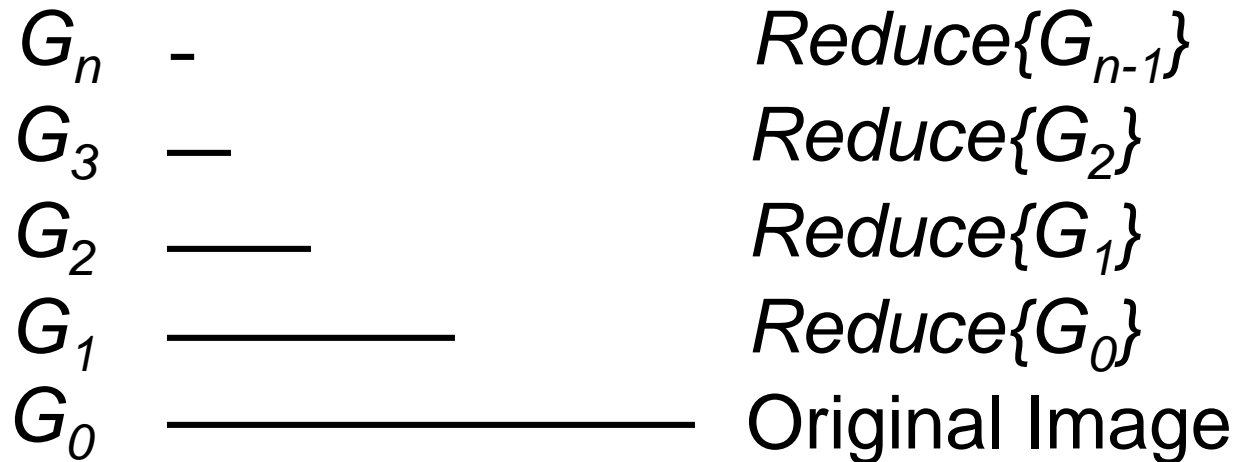


Mirror on last pixel. ?=2

Mirror after last pixel. ?=0

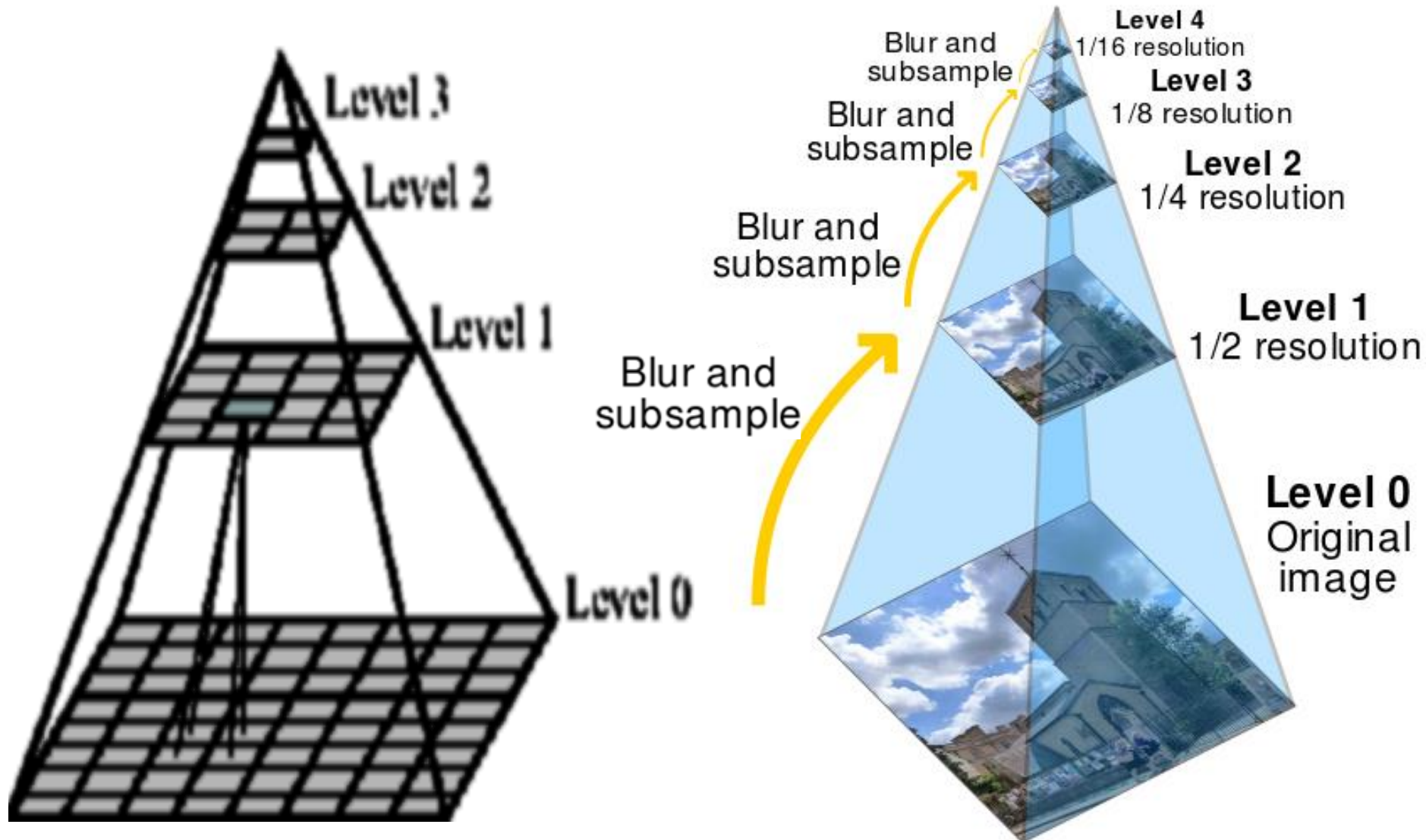
Duplicate last pixel. ?=0

Gaussian Pyramid



- 2D Picture
 - Reduce Rows, Reduce Columns

5-Level Gaussian Pyramid (Wikipedia)





G_0

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.



G_1



G_2



G_3



G_4



$G_{0.5}$



$G_{1.5}$



$G_{2.5}$

Laplacian Pyramid

Gaussian

Laplacian

G_n	L_n -	G_n
G_3	L_3 —	$G_3 - \text{Expand}\{\text{Reduce}\{G_3\}\}$
G_2	L_2 ——	$G_2 - \text{Expand}\{\text{Reduce}\{G_2\}\}$
G_1	L_1 —————	$G_1 - \text{Expand}\{G_2\}$
G_0	L_0 —————	$G_0 - \text{Expand}\{G_1\}$

$$\begin{aligned}
 L_n + L_{n-1} &= \text{Expand}\{L_n\} + L_{n-1} = \\
 &= \text{Expand}\{G_n\} + (G_{n-1} - \text{Expand}\{G_n\}) = G_{n-1}
 \end{aligned}$$

$$\sum_{i=k}^n (L_i) = G_k$$

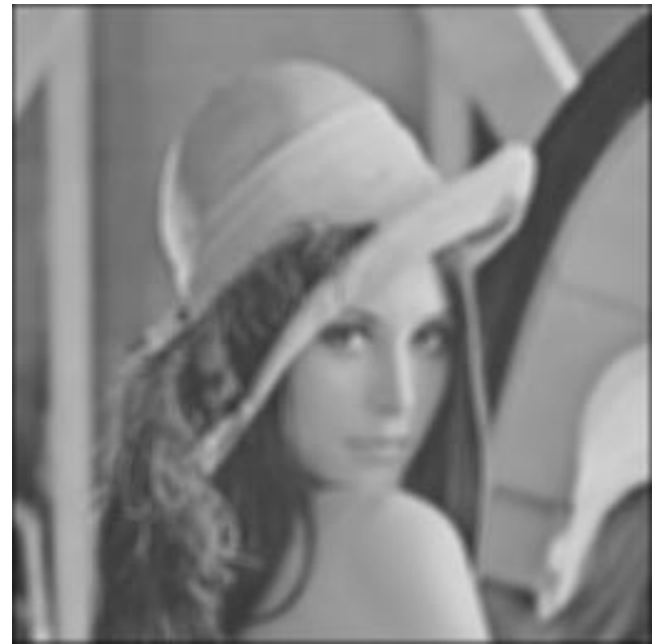
$$\sum_{i=0}^n (L_i) = G_0$$



G_0



G_1



$\text{Expand}\{G_1\}$

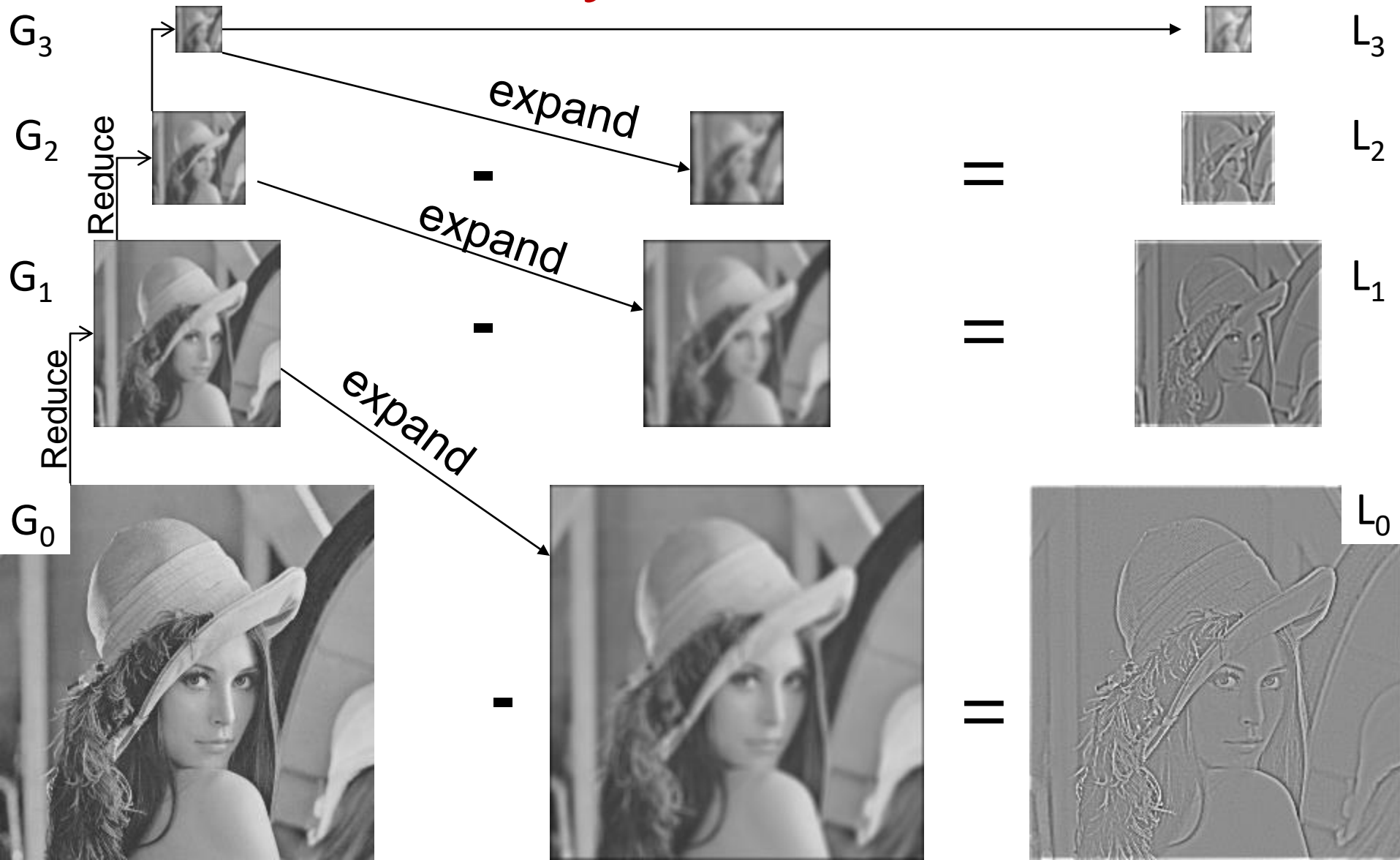


$L_0 = G_0 - \text{Expand}\{G_1\}$

Gaussian - Laplacian Pyramids

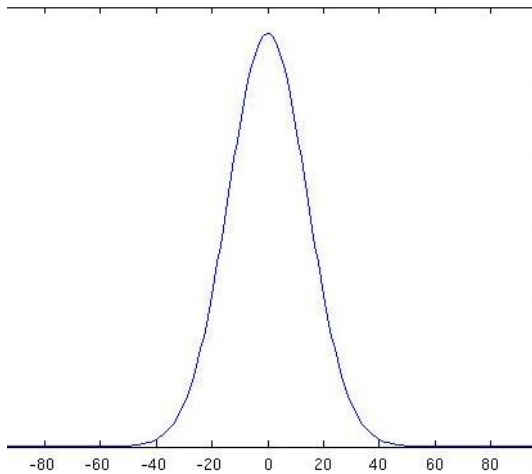
Gaussian Pyramid

Laplacian Pyramid

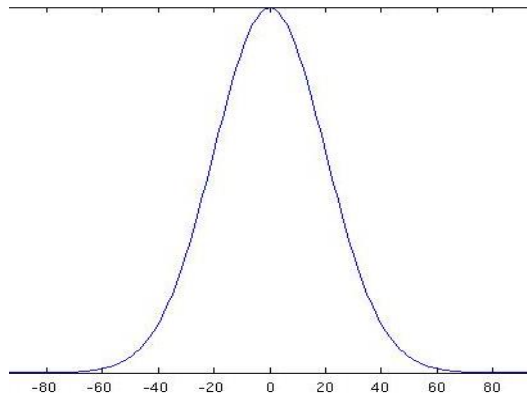


Laplacian Pyramid as a Band-Pass Filter

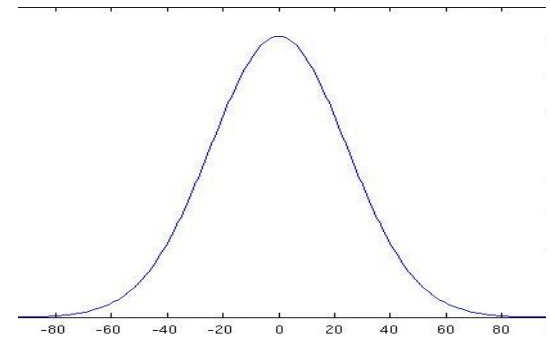
Gaussian Pyramid –
Convolution with a Gaussian filter



Level 1
Narrow



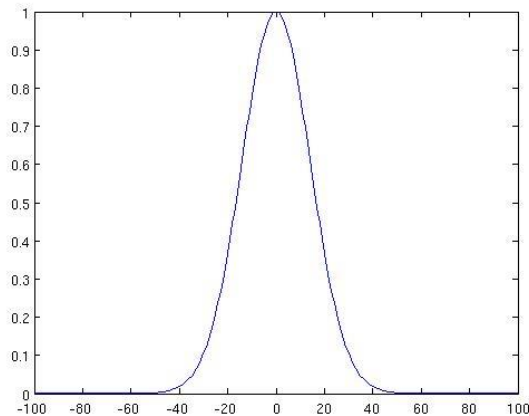
Level 2



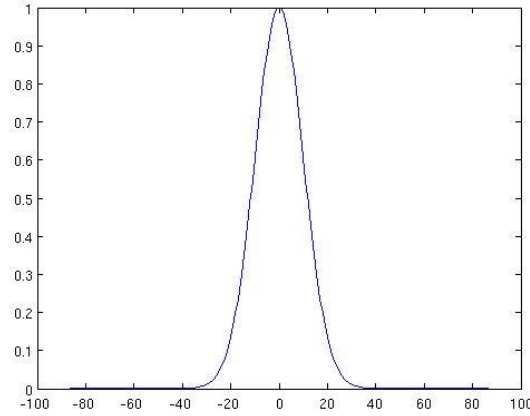
Level 3
Wide

Laplacian Pyramid as a Band-Pass Filter

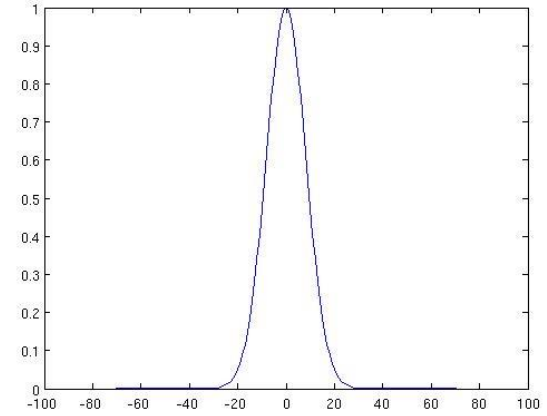
Gaussian Pyramid (in the Fourier domain) –
Multiplication with a Gaussian kernel



Level 1
Wide



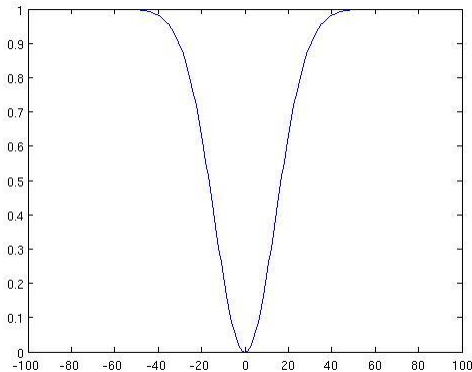
Level 2



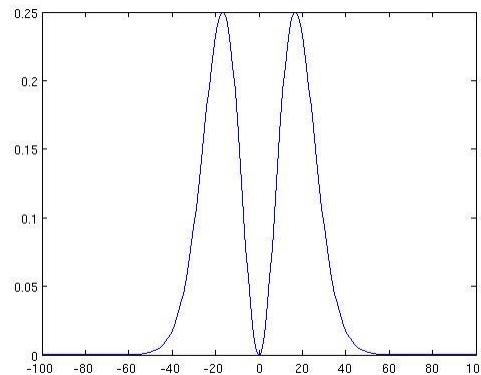
Level 3
Narrow

Laplacian Pyramid as a Band-Pass Filter

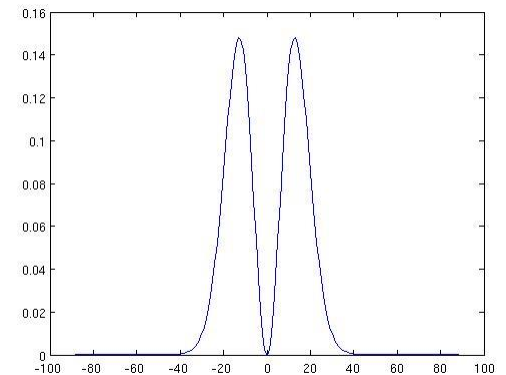
- In the Fourier domain, the Laplacian is the difference between two powers of Gaussian kernels:



Level 1



Level 2



Level 3

Pyramid Compression

- Build a Laplacian Pyramid
- Quantize pyramid values to 3-5 values
 - Optimal Quantization
- Compress using Entropy Compression
 - (Huffman, Lempel-Ziv)
- Reconstruct normally
- Next Generation: Wavelet Compression

Pyramid Compression

8 bits/pixel

1 bits/pixel

0.5 bits/pixel



(a)



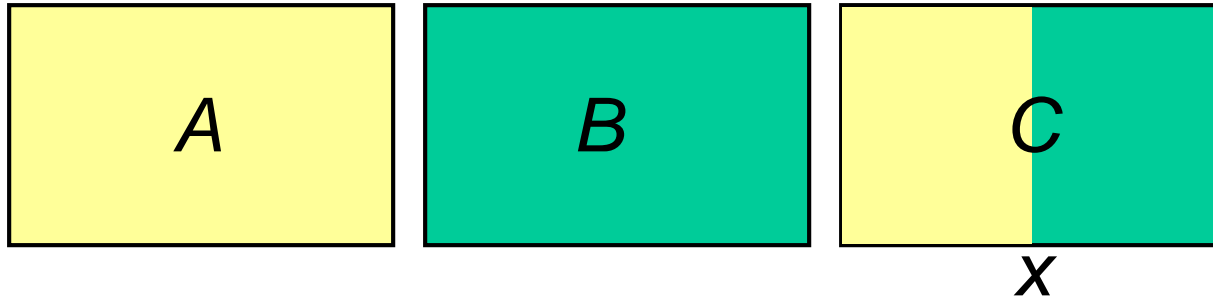
(b)



(c)

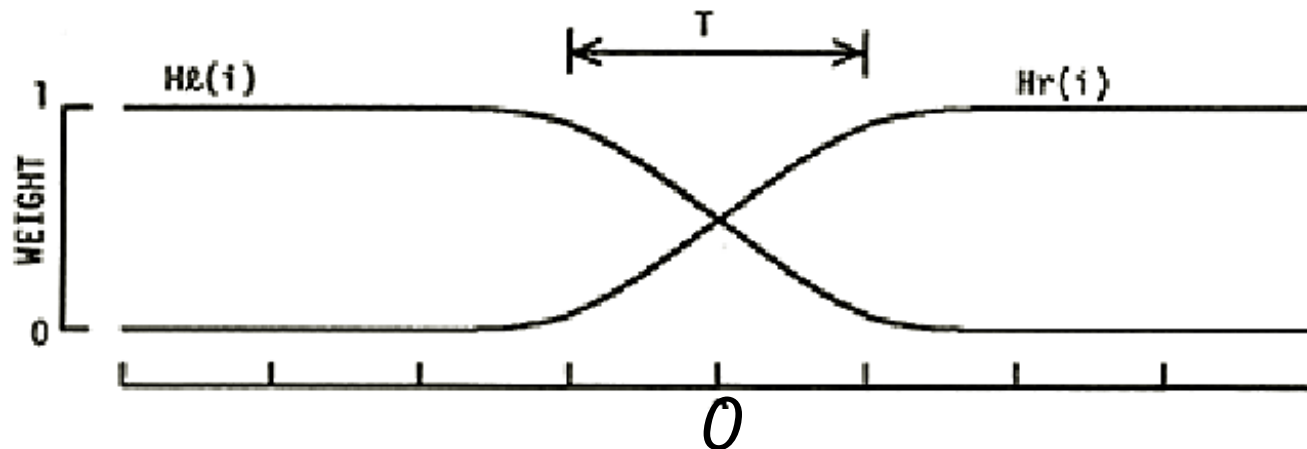
Fig. 5. Pyramid data compression. The original image represented at 8 bits per-pixel is shown in (a). The node values of the Laplacian pyramid representation of this image were quantitized to obtain effective data rates of 1 b/p and 1/2 b/p. Reconstructed images (b) and (c) show relatively little degradation.

Picture Merging with Spline



For every Row:

$$C(i) = H_l(i-x)A(i) + H_r(i-x)B(i)$$



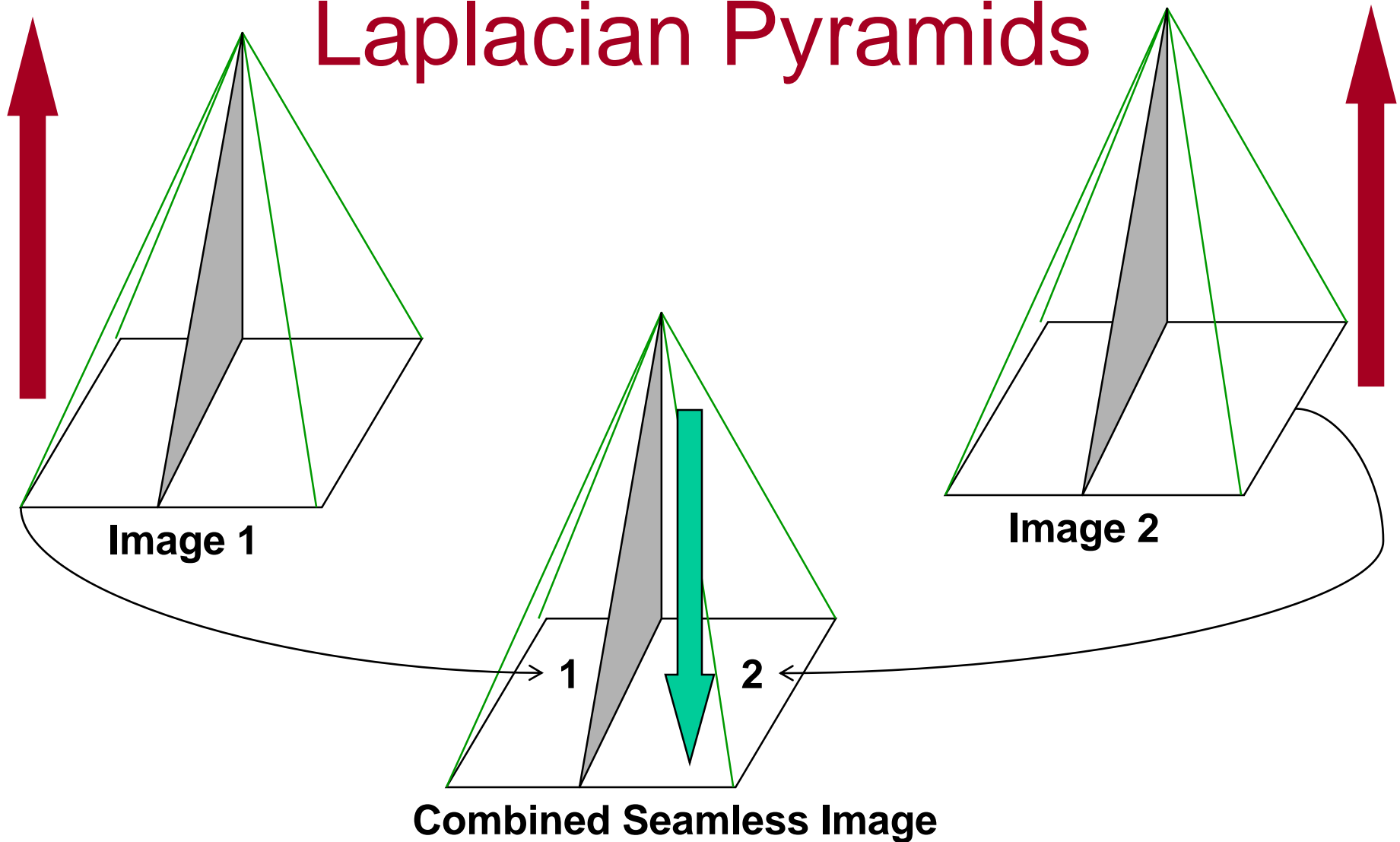
Multiresolution Spline

- Given two images A and B to be splined in middle
- Construct Laplacian Pyramid L_a and L_b
- Create a third Laplacian Pyramid L_c where for each level k

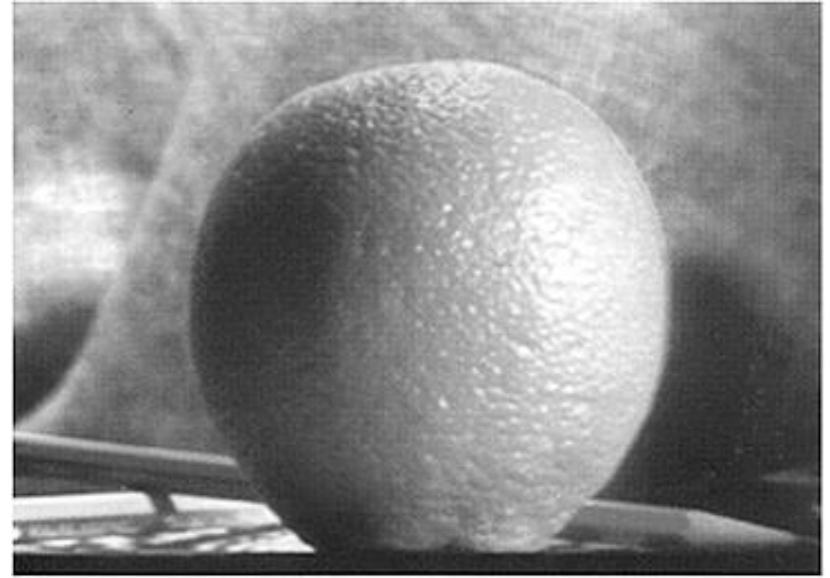
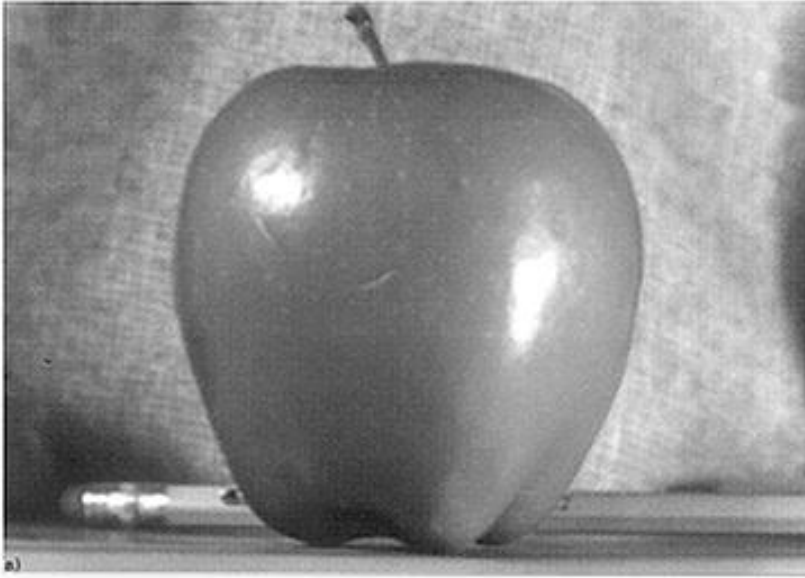
$$L_c(i, j) = \begin{cases} L_a(i, j) & \text{if } i < width / 2 \\ (L_a(i, j) + L_b(i, j)) / 2 & \text{if } i = width / 2 \\ L_b(i, j) & \text{if } i > width / 2 \end{cases}$$

- Sum all levels in L_c to get the blended image

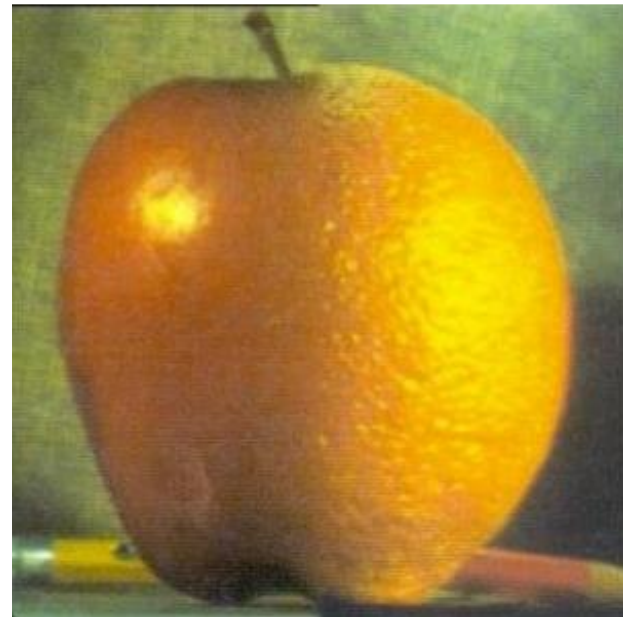
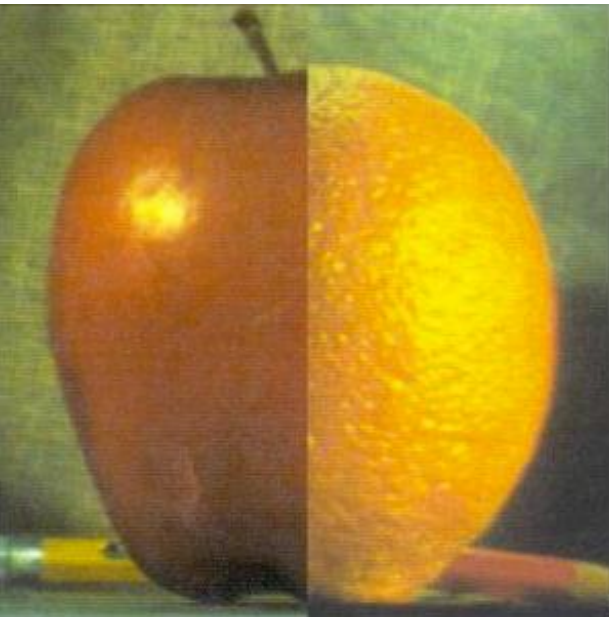
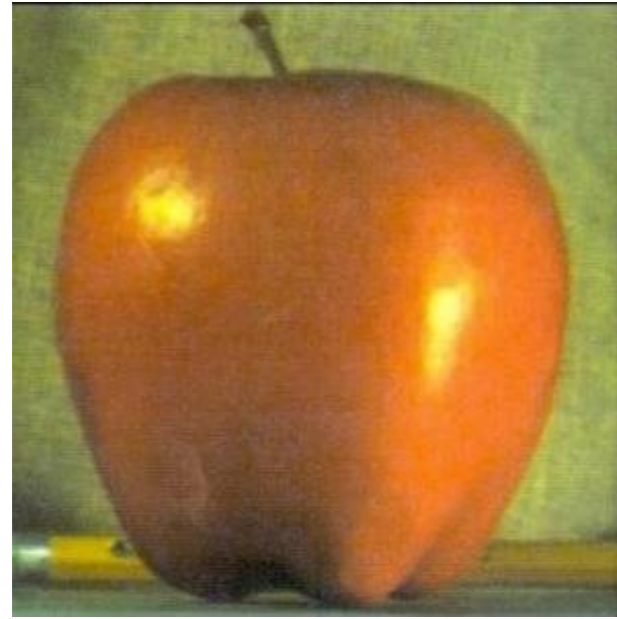
Image Merging with Laplacian Pyramids



Pyramid Blending Example 1



Pyramid Blending Example 1



Pyramid Blending Arbitrary Shape

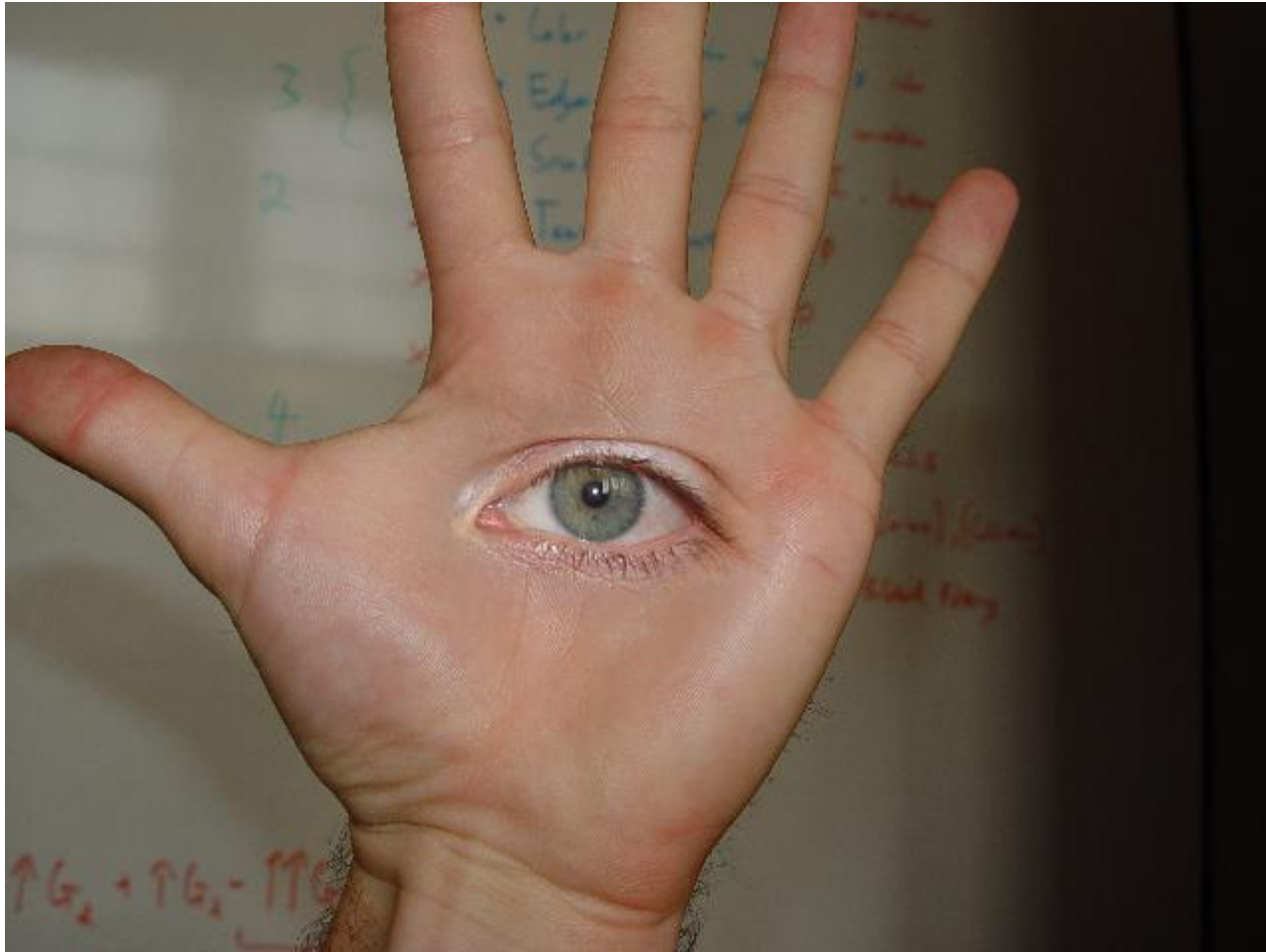
- Given two images A and B , and a binary mask M
- Construct Laplacian Pyramids L_a and L_b
- Construct a Gaussian Pyramid G_m
- Create a third Laplacian Pyramid L_c where for each level k

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

- Sum all levels L_c in to get the blended image

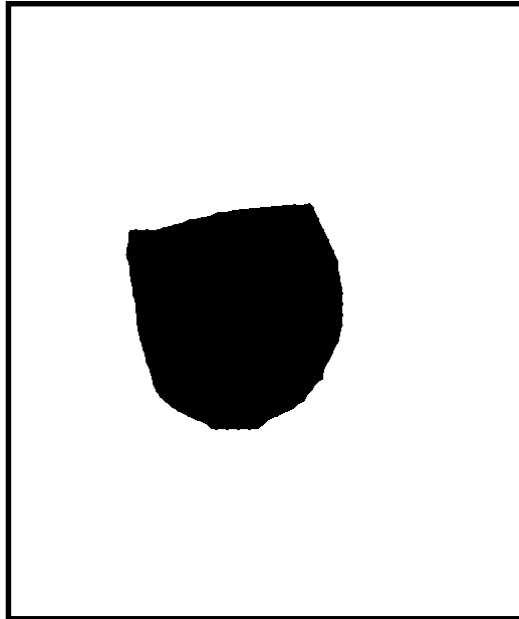
Pyramid Blending Example 2





© prof. dmartin

Pyramid Blending Example



Pyramid Blending Example



Pyramids (not Laplacian) & sea (tal)



