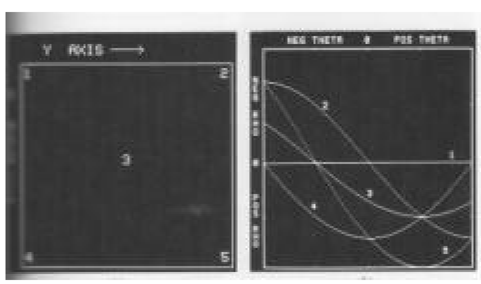
Lecture-12

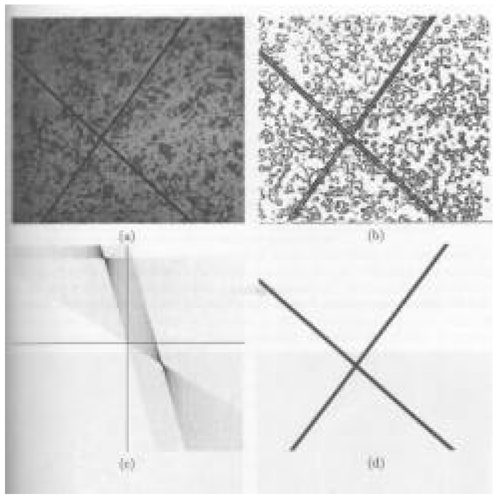
Hough Transform

Examples

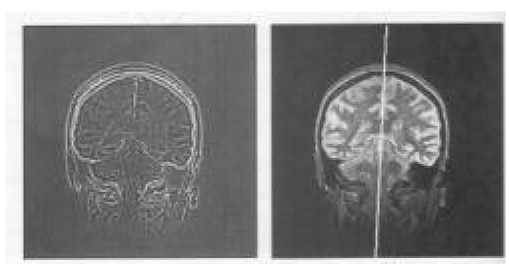
Hough Space



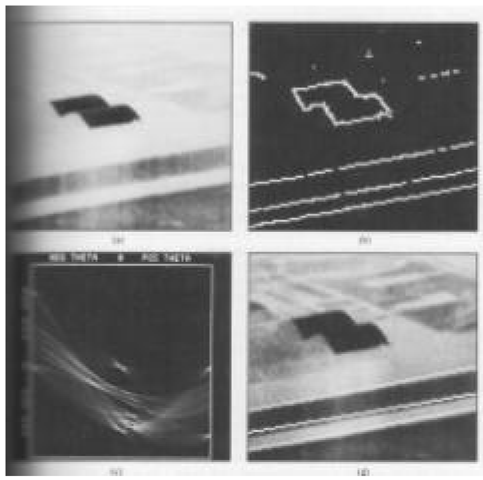
Fitting Lines In an Image



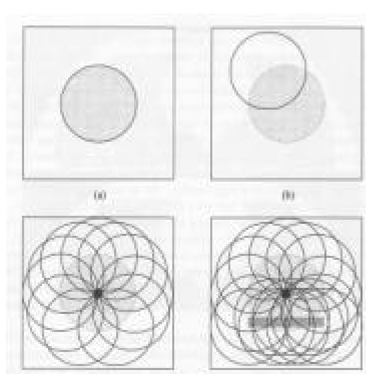
Fitting Lines In an Image



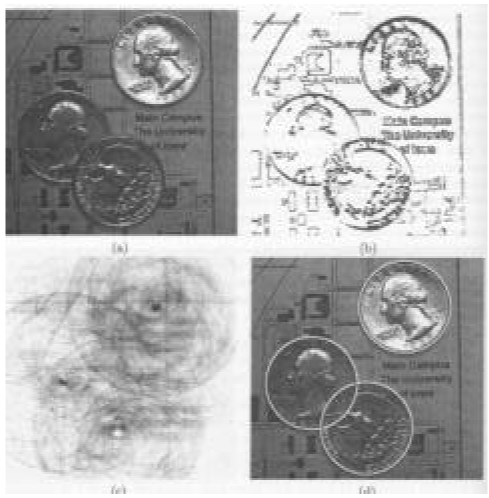
Fitting lines in an image



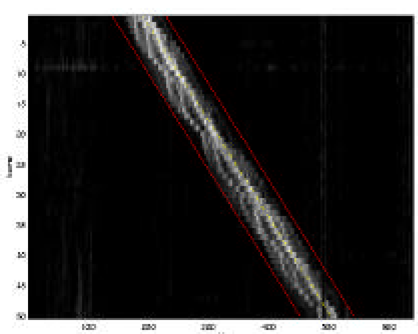
Fitting Circles

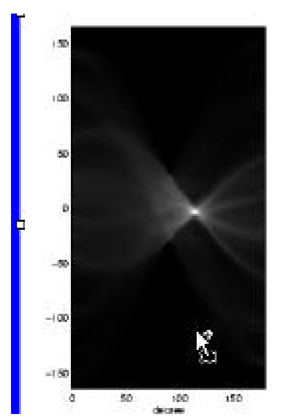


Fitting Circles



Detecting Lines in Gray Level Images



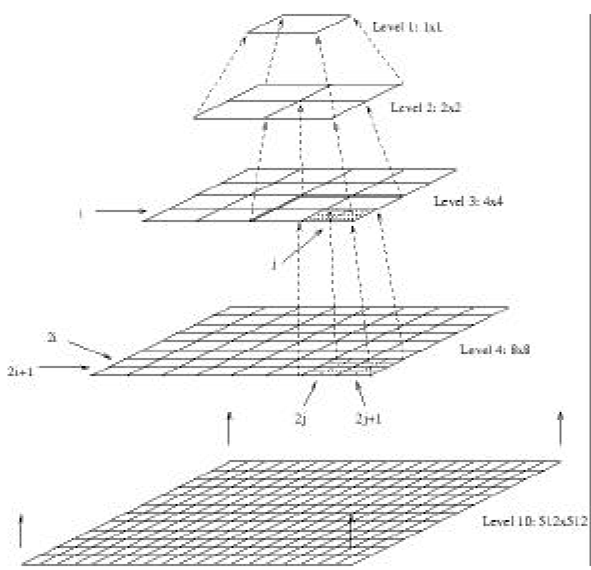


Detect yellow line in the middle
Use gray levels instead of edged
Increment the parameter space by gray level at a pixel instead of by 1.

Pyramids

- Very useful for representing images.
- Pyramid is built by using multiple copies of image.
- Each level in the pyramid is 1/4 of the size of previous level.
- The lowest level is of the highest resolution.
- The highest level is of the lowest resolution.

Pyramid

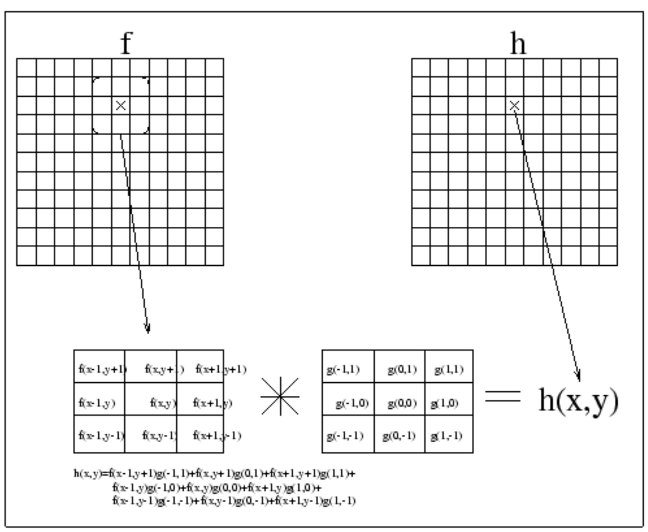


Gaussian Pyramids

$$g_l(i,j) = \sum_{m=-2}^{2}\sum_{n=-2}^{2} w(m,n)\, g_{l-1}(2i+m, 2j+n)$$

$$g_l = REDUCE[g_{l-1}]$$

Convolution



Gaussian Pyramids

$$g_{l,n}(i, j) = \sum_{p=-2}^{2}\sum_{q=-2}^{2} w(p, q) g_{l,n-1}(\frac{i-p}{2}, \frac{j-q}{2})$$

$$g_{l,n} = EXPAND[g_{l,n-1}]$$

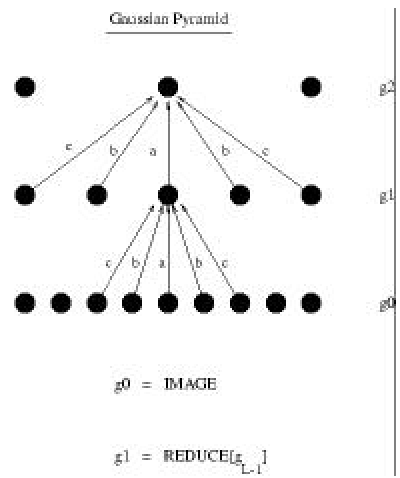
Reduce (1D)

$$g_l(i) = \sum_{m=-2}^{2} \hat{w}(m) g_{l-1}(2i+m)$$

$$g_l(2) = \hat{w}(-2) g_{l-1}(4-2) + \hat{w}(-1) g_{l-1} \hat{w}(4-1) +$$
$$\hat{w}(0) g_{l-1}(4) + \hat{w}(1) g_{l-1}(4+1) + \hat{w}(2) g_{l-1}(4+2)$$

$$g_l(2) = \hat{w}(-2) g_{l-1}(2) + \hat{w}(-1) g_{l-1} \hat{w}(3) +$$
$$\hat{w}(0) g_{l-1}(4) + \hat{w}(1) g_{l-1}(5) + \hat{w}(2) g_{l-1}(6)$$

Reduce



Expand (1D)

$$g_{l,n}(i) = \sum_{p=-2}^{2} \hat{w}(p) g_{l,n-1}\left(\frac{i-p}{2}\right)$$

$$g_{l,n}(4) = \hat{w}(-2) g_{l,n-1}\left(\frac{4-2}{2}\right) + \hat{w}(-1) g_{l,n-1}\left(\frac{4-1}{2}\right) +$$

$$\hat{w}(0) g_{l,n-1}\left(\frac{4}{2}\right) + \hat{w}(1) g_{l,n-1}\left(\frac{4+1}{1}\right) + \hat{w}(2) g_{l,n-1}\left(\frac{4+2}{2}\right)$$

$$g_{l,n}(4) = \hat{w}(-2) g_{l,n-1}(1) + \hat{w}(0) g_{l,n-1}(2) + \hat{w}(2) g_{l,n-1}(3)$$

Expand (1D)

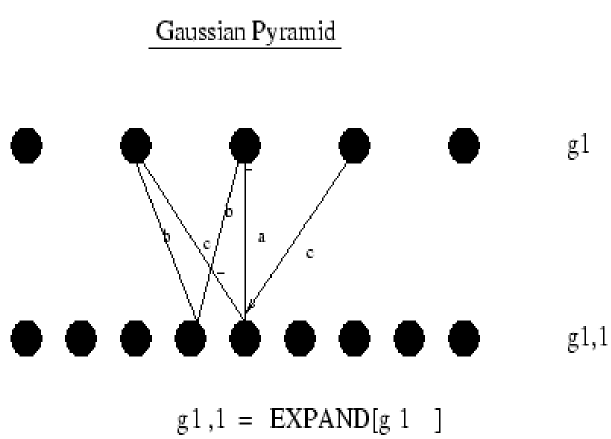
$$g_{l,n}(i) = \sum_{p=-2}^{2} \hat{w}(p) g_{l,n-1}\left(\frac{i-p}{2}\right)$$

$$g_{l,n}(3) = \hat{w}(-2) g_{l,n-1}\left(\frac{3-2}{2}\right) + \hat{w}(-1) g_{l,n-1}\left(\frac{3-1}{2}\right) +$$

$$\hat{w}(0) g_{l,n-1}\left(\frac{3}{2}\right) + \hat{w}(1) g_{l,n-1}\left(\frac{3+1}{1}\right) + \hat{w}(2) g_{l,n-1}\left(\frac{3+2}{2}\right)$$

$$g_{l,n}(3) = \hat{w}(-1) g_{l,n-1}(1) + \hat{w}(1) g_{l,n-1}(2)$$

Expand



Convolution Mask

$$[w(-2), w(-1), w(0), w(1), w(2)]$$

Convolution Mask

- Separable

$$w(m,n) = \hat{w}(m)\hat{w}(n)$$

- Symmetric

$$\hat{w}(i) = \hat{w}(-i)$$

$$[c, b, a, b, c]$$

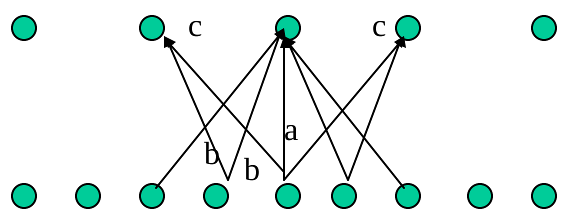
Convolution Mask

- The sum of mask should be 1.

$$a + 2b + 2c = 1$$

•All nodes at a given level must contribute the same total weight to the nodes at the next higher level.

$$a + 2c = 2b$$



Convolution Mask

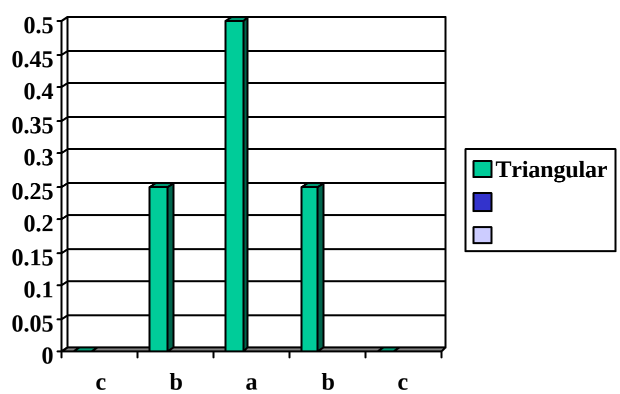
$$\hat{w}(0) = a$$

$$\hat{w}(-1) = \hat{w}(1) = \frac{1}{4}$$

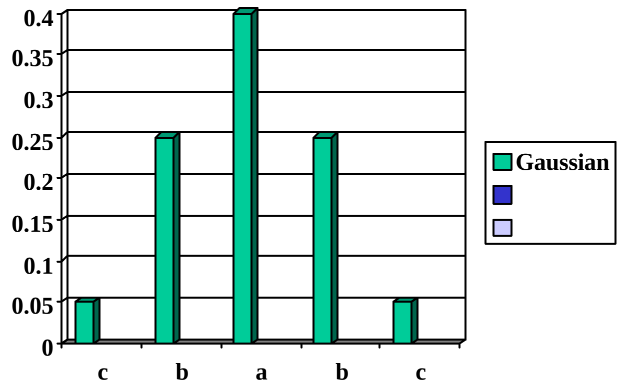
$$\hat{w}(-2) = \hat{w}(2) = \frac{1}{4} - \frac{a}{2}$$

a=.4 GAUSSIAN, a=.5 TRINGULAR

Triangular



Approximate Gaussian



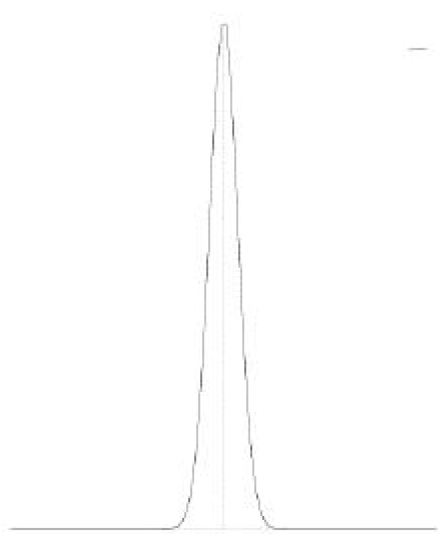
Gaussian

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

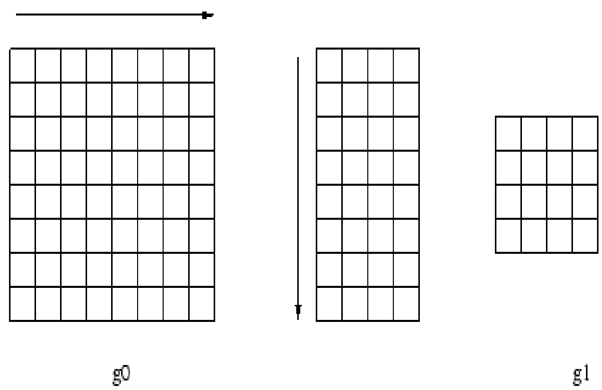
Gaussian

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

x	-3	-2	-1	0	1	2	3
g(x)	.011	.13	.6	1	.6	.13	.011



Separability



Algorithm

- Apply 1-D mask to alternate pixels along each row of image.
- Apply 1-D mask to each pixel along alternate columns of resultant image from previous step.

Gaussian Pyramid

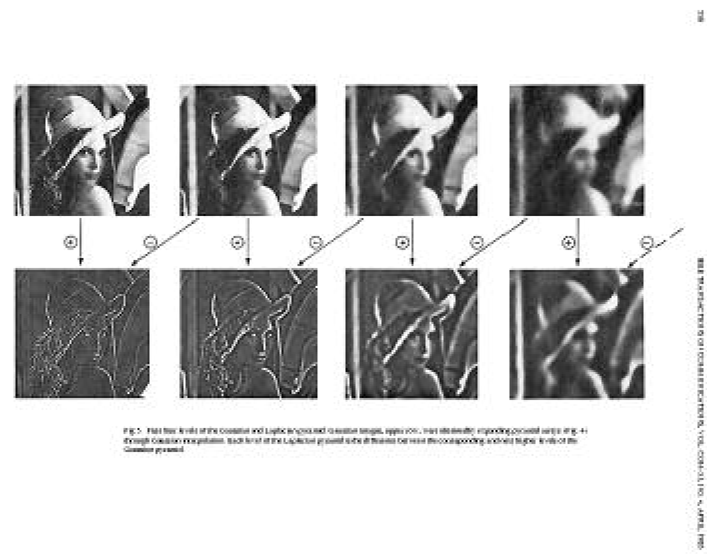
Laplacian Pyramids

- Similar to edge detected images.
- Most pixels are zero.
- Can be used for image compression.

$$L_1 = g_1 - EXPAND[g_2]$$

$$L_2 = g_2 - EXPAND[g_3]$$

$$L_3 = g_3 - EXPAND[g_4]$$



Coding using Laplacian Pyramid

- Compute Gaussian pyramid

$$g_1, g_2, g_3, g_4$$

- Compute Laplacian pyramid

$$L_1 = g_1 - EXPAND[g_2]$$
$$L_2 = g_2 - EXPAND[g_3]$$
$$L_3 = g_3 - EXPAND[g_4]$$
$$L_4 = g_4$$

- Code Laplacian pyramid

Decoding using Laplacian pyramid

- Decode Laplacian pyramid.
- Compute Gaussian pyramid from Laplacian pyramid.

$$g_4 = L_4$$

$$g_3 = EXPAND[g_4] + L_3$$

$$g_2 = EXPAND[g_3] + L_2$$

$$g_1 = EXPAND[g_2] + L_1$$

- is reconstructed image.

Laplacian Pyramid

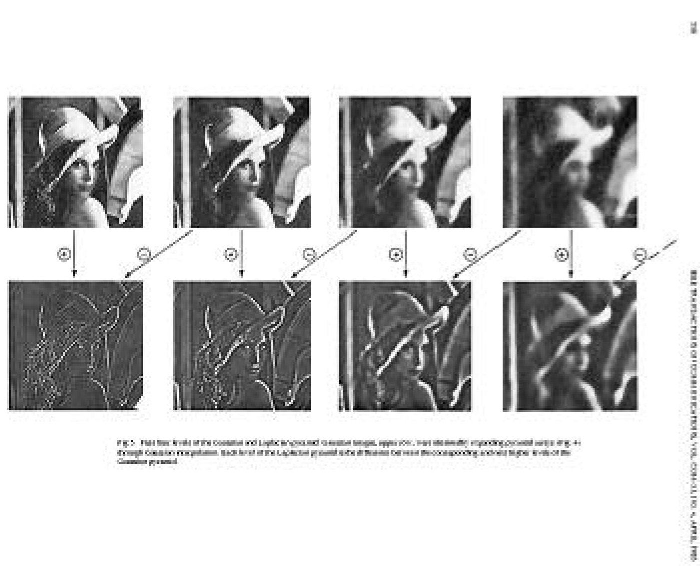
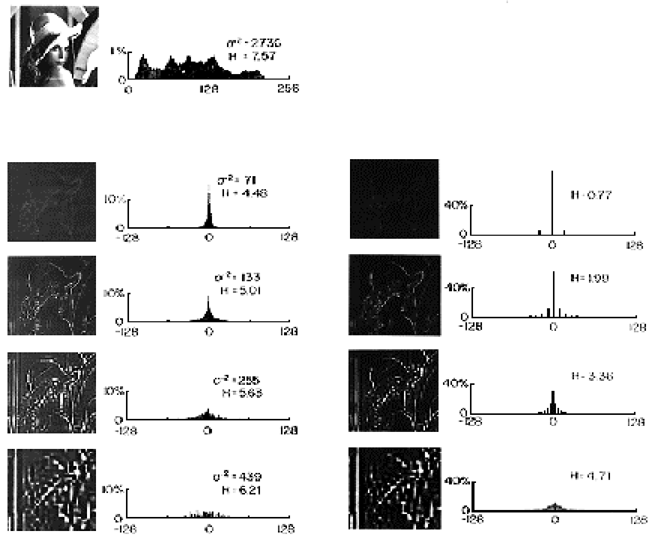
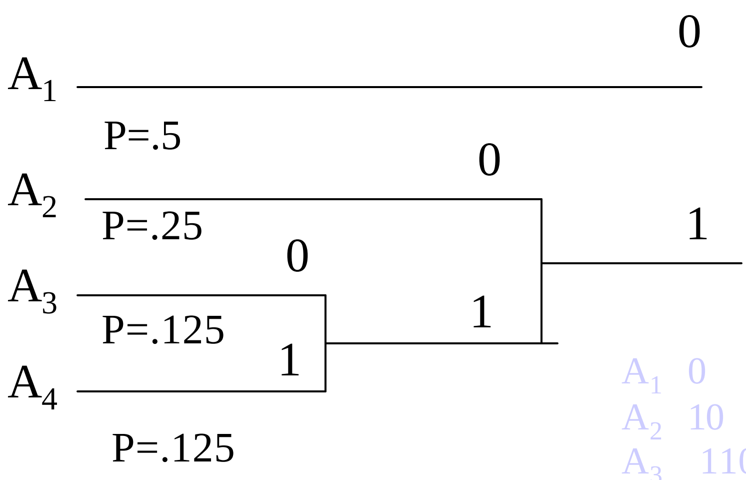


Image Compression (Entropy)



Huffman Coding (Example-1)



A$_1$ 0
A$_2$ 10
A$_3$ 110
A$_4$ 111

Huffman Coding

Entropy $$H = -\sum_{i=0}^{255} p(i)\log_2 p(i)$$

$$H = -.5\log.5 - .25\log.25 - .125\log.125 - .125\log.125 = 1.75$$

Image Compression

1.58

1

(a)

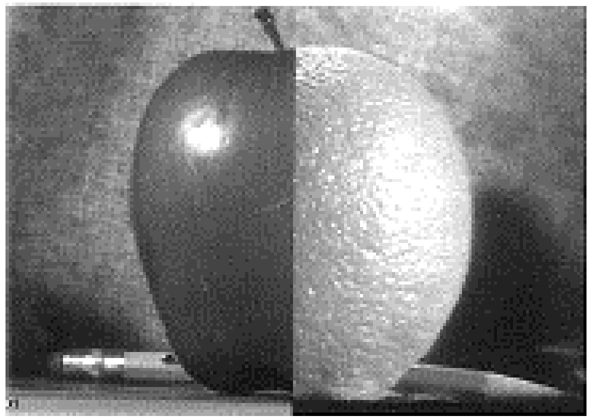
(b)

.73

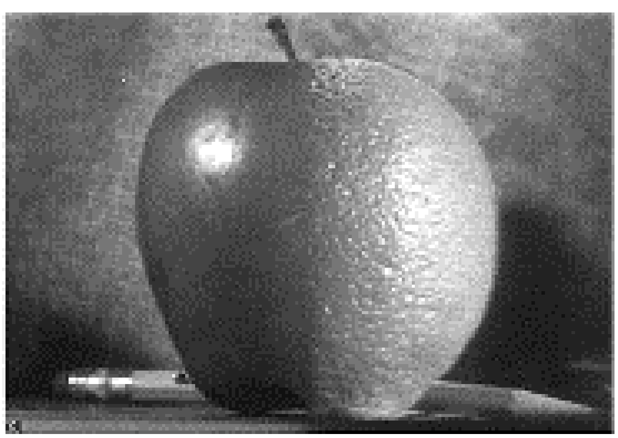
(c)

(d)

Combining Apple & Orange



Combining Apple & Orange



Algorithm

- Generate Laplacian pyramid Lo of orange image.
- Generate Laplacian pyramid La of apple image.
- Generate Laplacian pyramid Lc by copying left half of nodes at each level from apple and right half of nodes from orange pyramids.
- Reconstruct combined image from Lc.