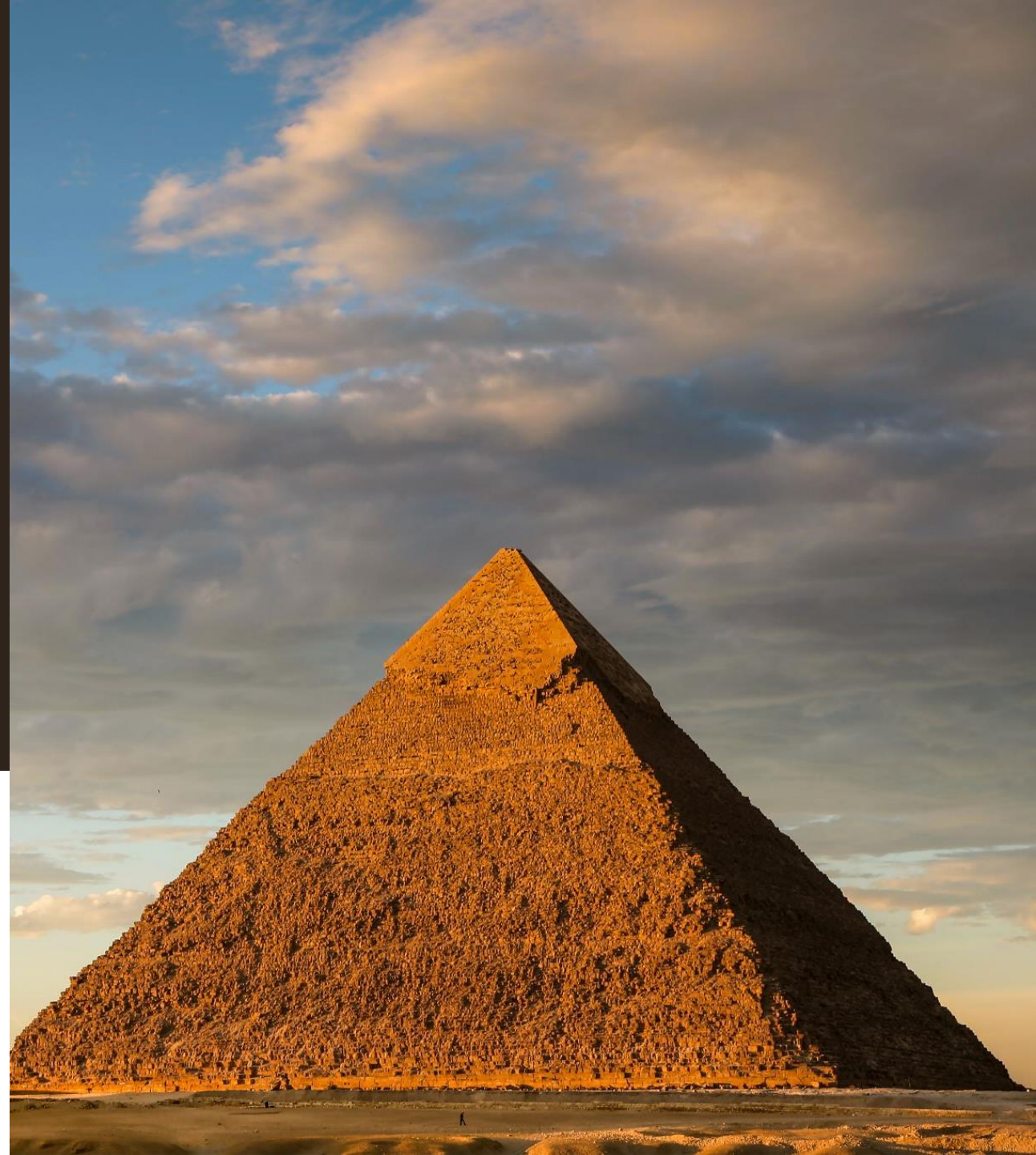


# Pyramid

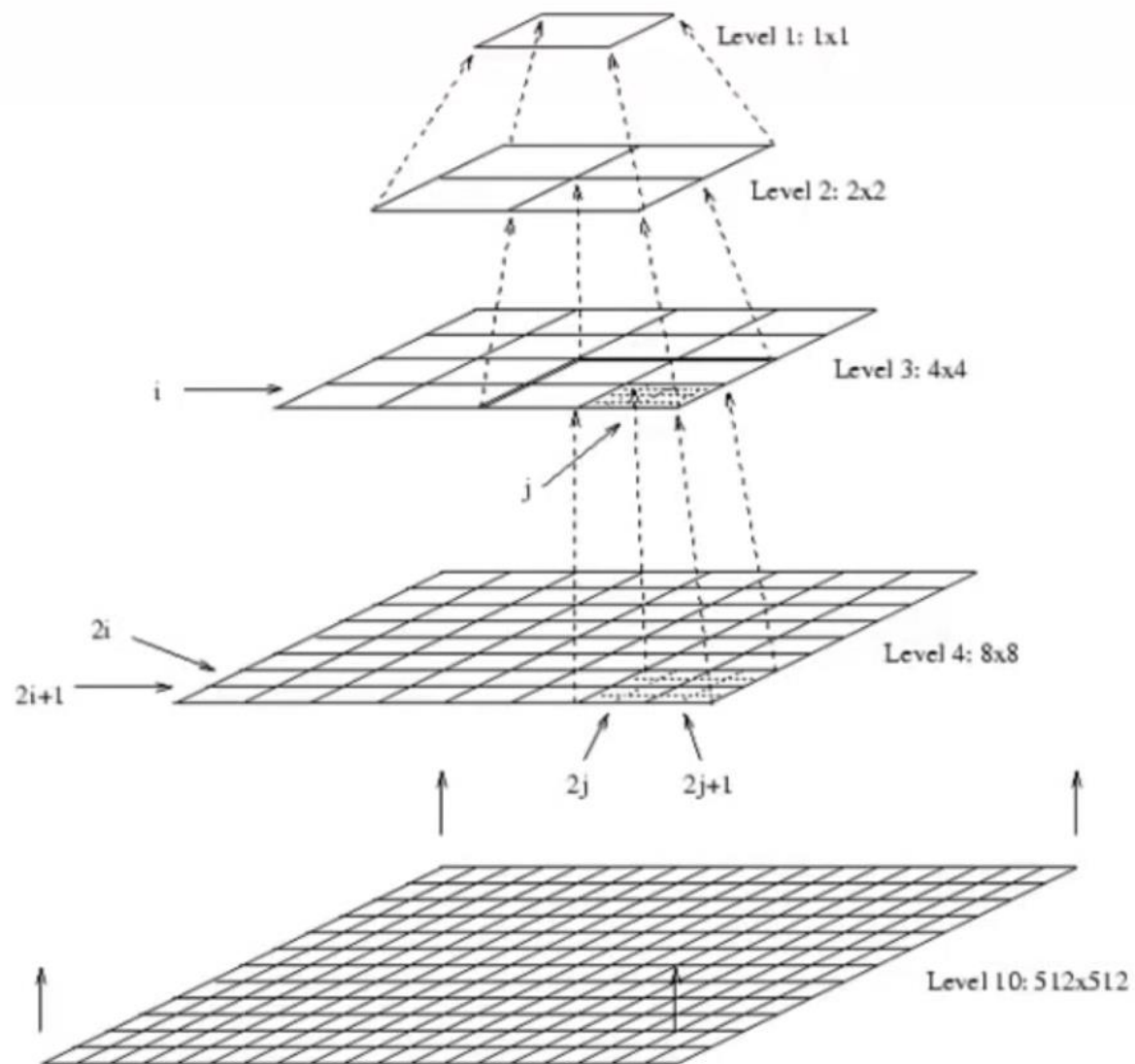
Gaussian



# Applications of Pyramid

1. Representing image at different Scales
2. Reduce or Expand the resolution
3. Image Compression
4. Image Compositing
5. Optical flow using Pyramids

# Gaussian Pyramid



# Gaussian Pyramid (Reduce)

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

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

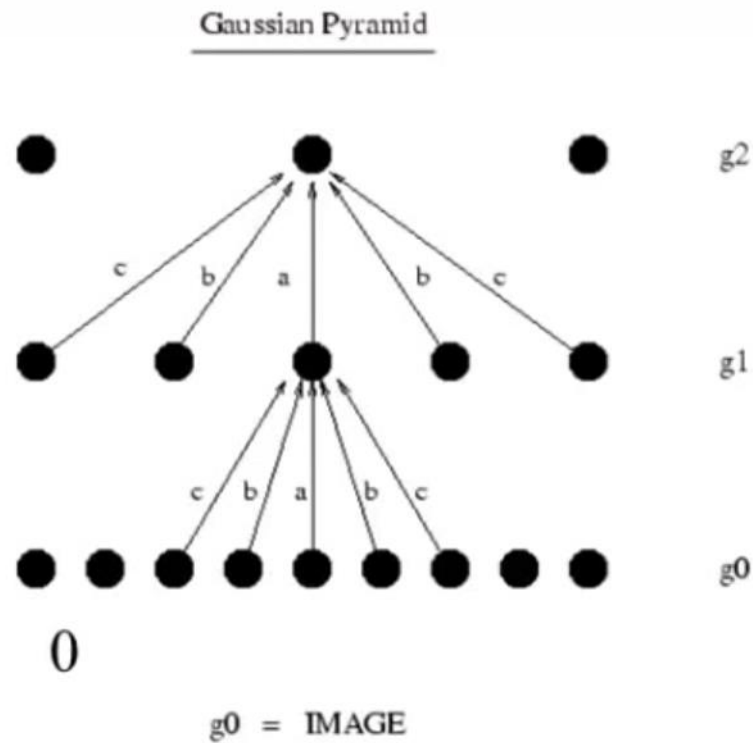
# Gaussian Pyramid (Reduce)

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

$$i=1$$

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

# Gaussian Pyramid (Reduce)



# Gaussian Pyramid (Expand)

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

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

# Gaussian Pyramid (Expand)

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

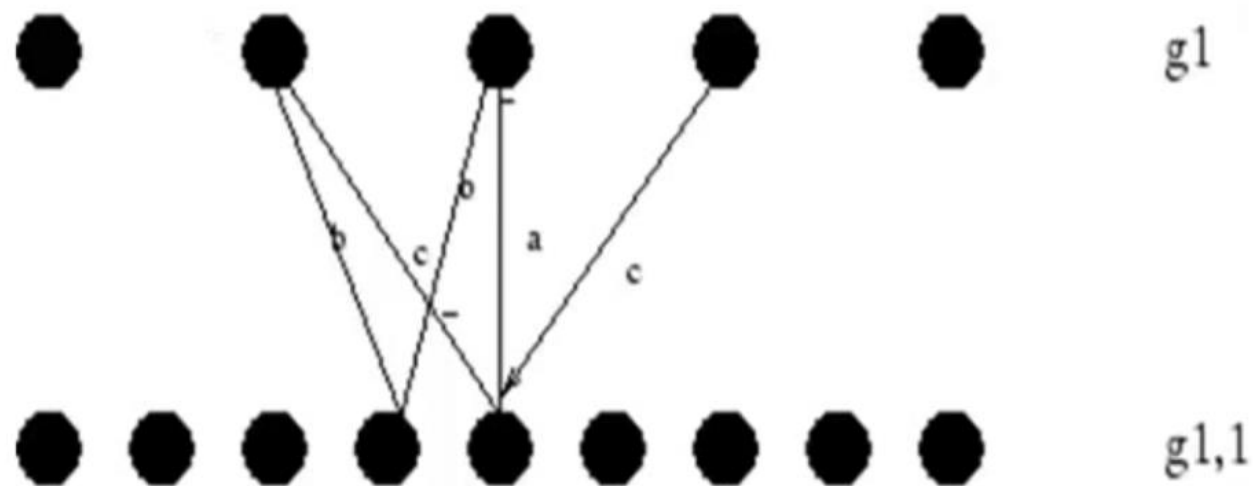
$$g_{l,n}(3) = \hat{w}(-2) g_{l,n-1}(5/2) + \hat{w}(-1) g_{l,n-1}(2) + \hat{w}(0) g_{l,n-1}(3/2) + \hat{w}(1) g_{l,n-1}(1) + \hat{w}(2) g_{l,n-1}(1/2)$$

$$g_{l,n}(4) = \hat{w}(-2) g_{l,n-1}(3) + \hat{w}(-1) g_{l,n-1}(5/2) + \hat{w}(0) g_{l,n-1}(2) + \hat{w}(1) g_{l,n-1}(3/2) + \hat{w}(2) g_{l,n-1}(1)$$



# Gaussian Pyramid (Expand)

Gaussian Pyramid



# Convolution mask properties

$$\begin{array}{cccccc} [w(-2), w(-1), w(0), w(1), w(2)] \\ [c \quad b \quad a \quad b \quad c] \end{array}$$

Separable

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

Symmetric

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

# Convolution mask properties

The sum of mask should be 1.

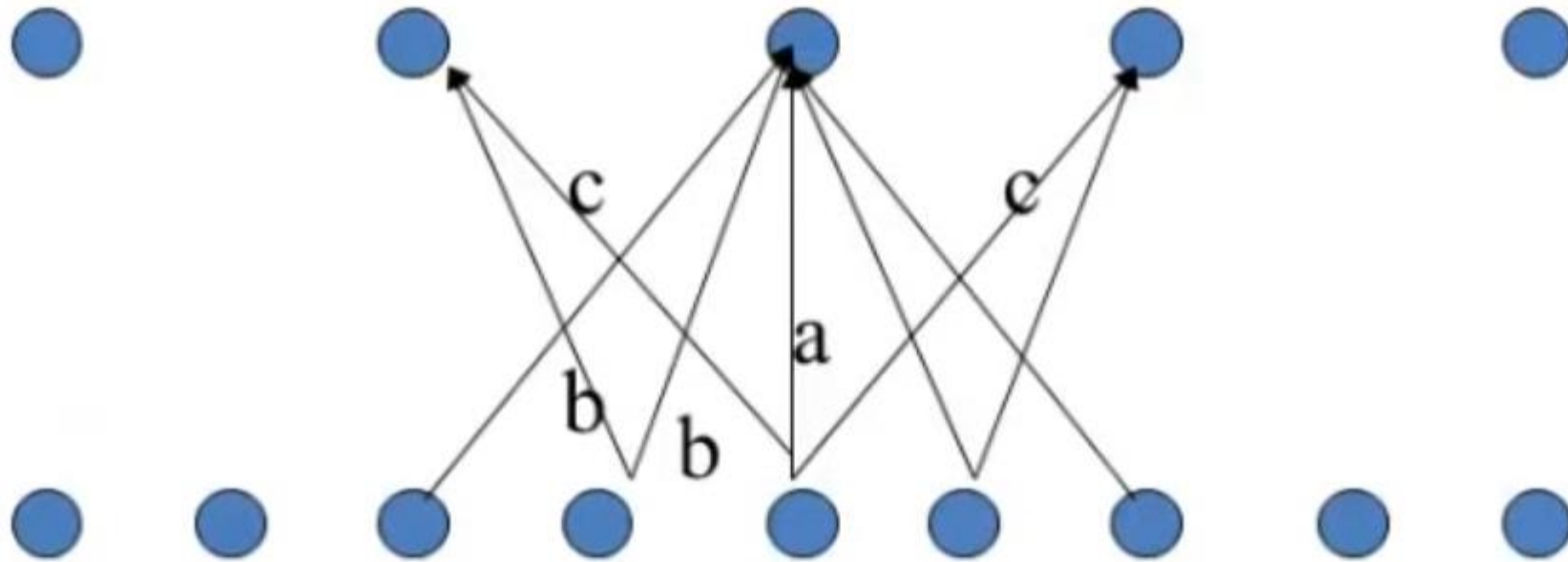
$$a + 2b + 2c = 1$$

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

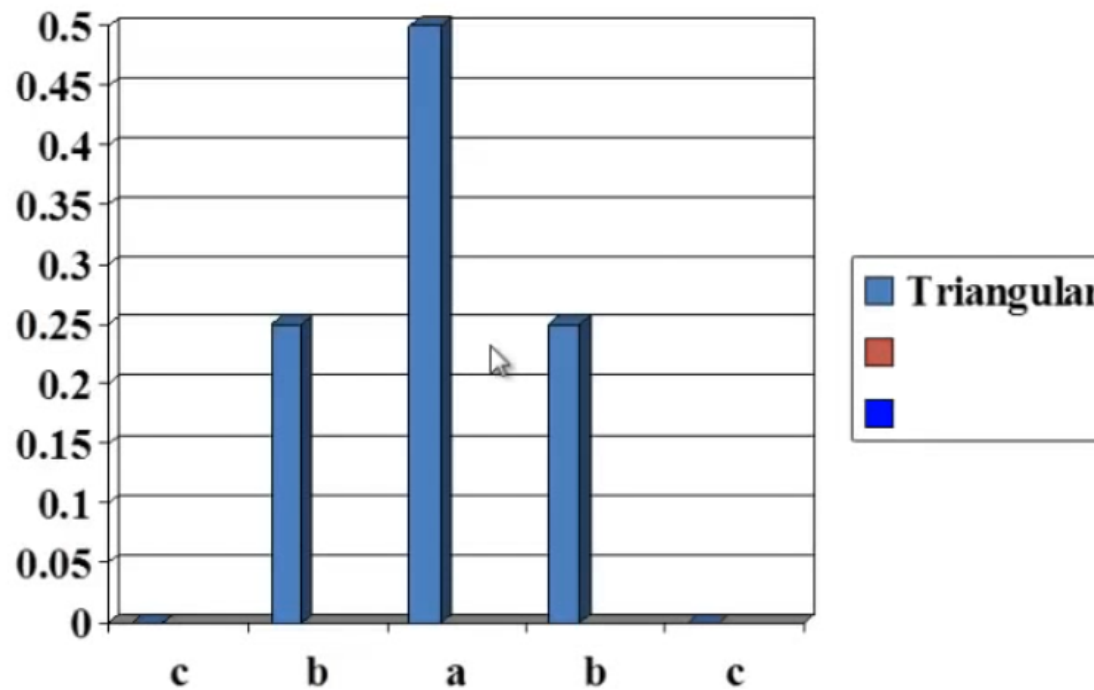
$$a + 2c = 2b$$

$$\Rightarrow b = \frac{1}{4} \quad ; \quad c = \frac{1}{4} - \frac{a}{2}$$

# Convolution mask properties

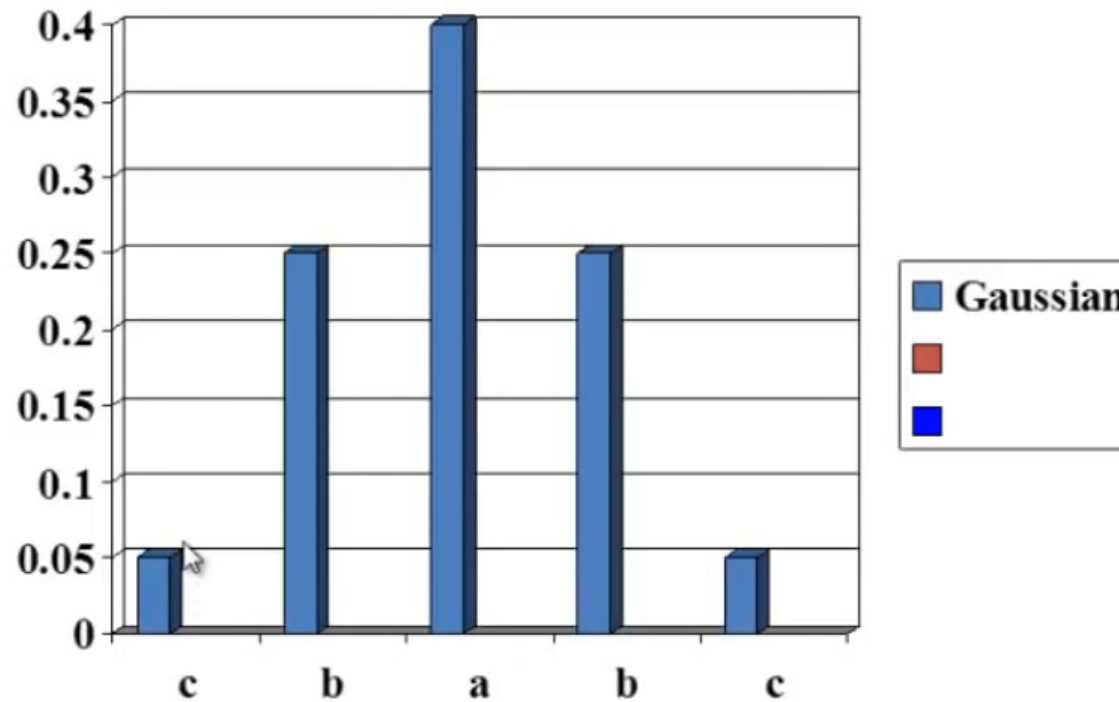


# Convolution mask (Triangular)



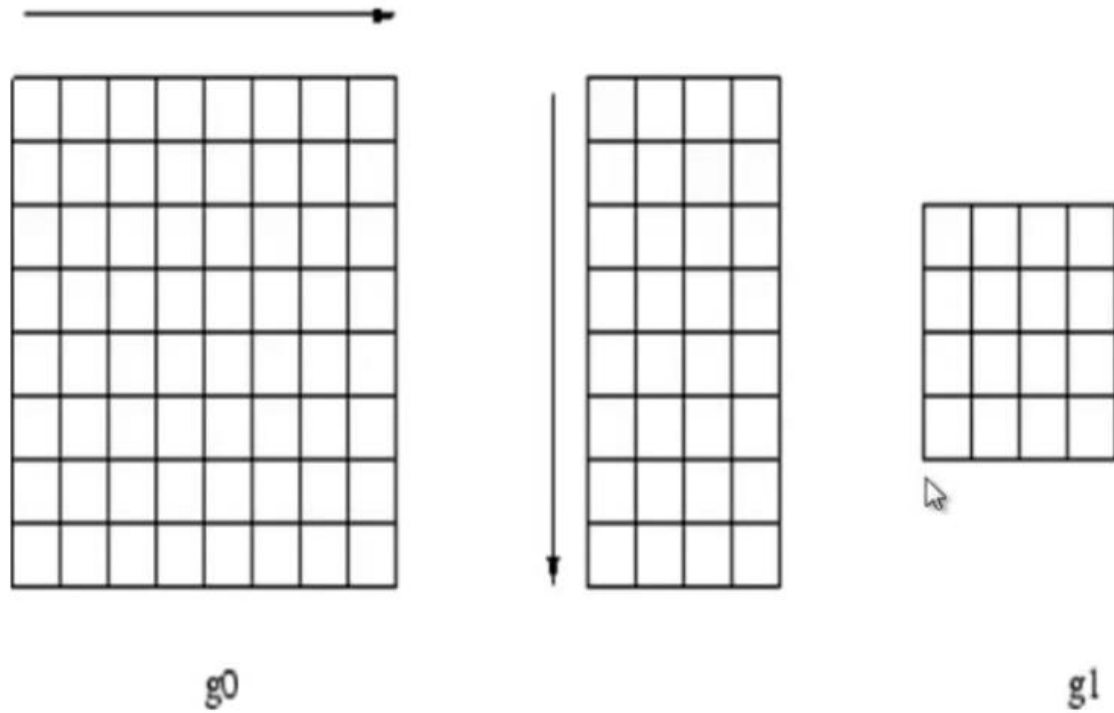
$$\underline{a = 0.5}$$

# Convolution mask (Gaussian)



$$\underline{a = 0.4}$$

# Separability



# Algorithm

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



