Gabor Filters

Ajay Charan (UG201211002)

Gabor Filter

- Gabor filter consists of two components,
- Complex sinusoid, (s(x,y)), known as carrier.
- Gaussian function, (g(x,y)), known as envelope
- Complex Gabor filter is product of these

Complex Sinusoid

$$c(x,y) = e^{-j(2\pi F(x\cos(\theta)+y\sin(\theta))+P)}$$

Gaussian Function

$$g_1(x,y) = \frac{1}{\sqrt{2\pi\sigma}} e^{-0.5\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)}$$

- In order to rotate the Gaussian function in the direction of the complex sinusoid, we have to modify the Gaussian function as

$$g(x,y) = \frac{1}{\sqrt{2\pi\sigma}} e^{-0.5(\frac{x_t}{\sigma_x^2} + \frac{y_t}{\sigma_y^2})}$$

$$x_t = (x * \cos(\theta) + y * \sin(\theta))^2$$

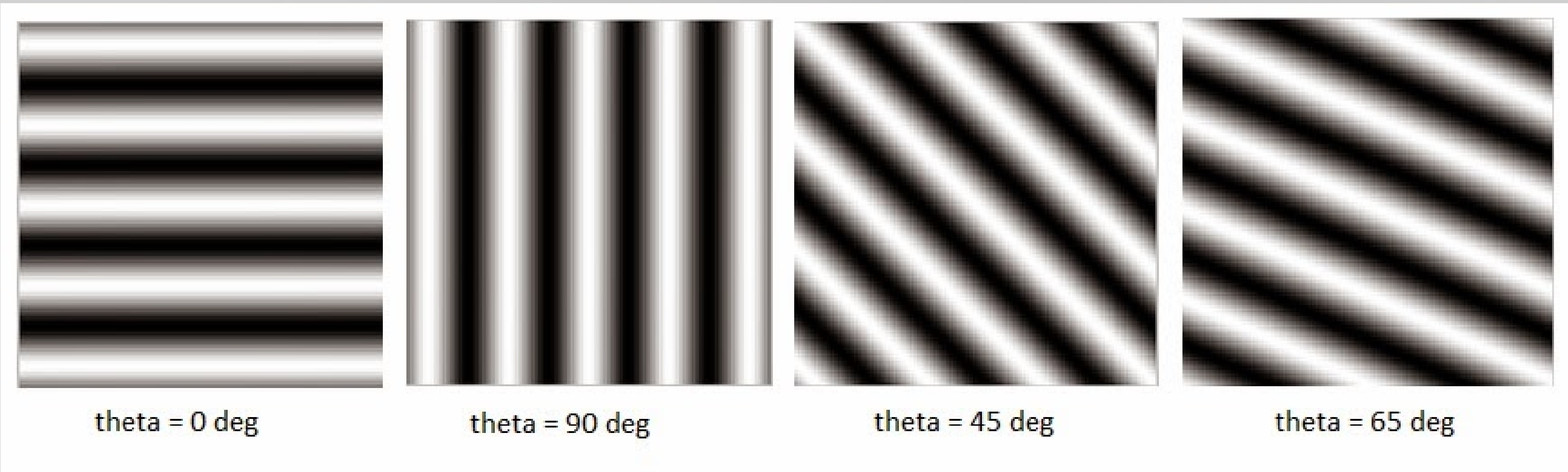
$$y_t = (-x * sin(\theta) + y * \cos(\theta))^2$$

2-D Gabor

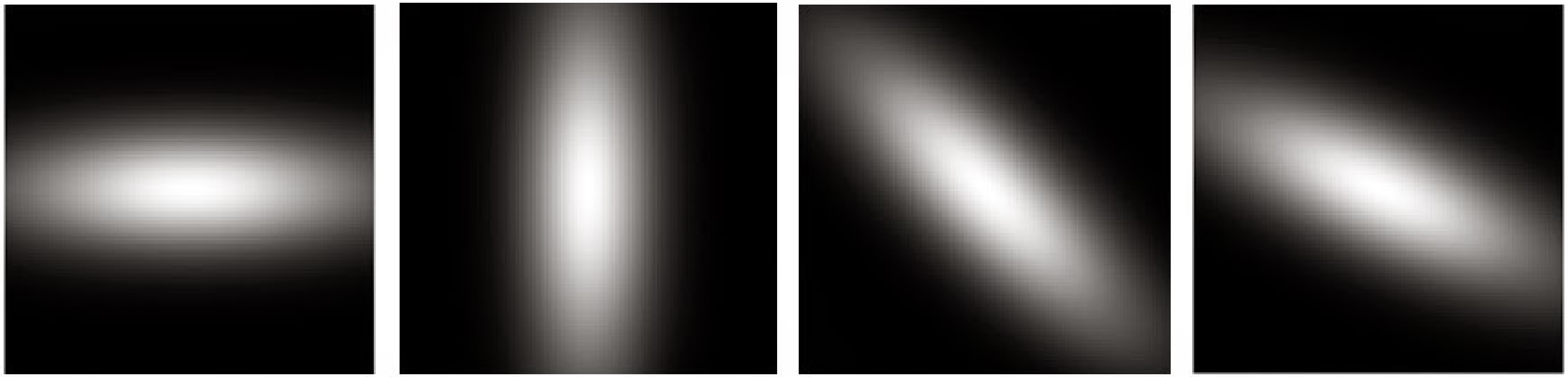
$$gabor(x,y) = c(x,y) * g(x,y)$$

$$gabor(x,y) = e^{-j(2\pi F(x\cos(\theta)+y\sin(\theta))+P)} * \frac{1}{\sqrt{2\pi}\sigma} e^{-0.5(\frac{x_t}{\sigma_x^2}+\frac{y_t}{\sigma_y^2})}$$

Carrier for different thetas

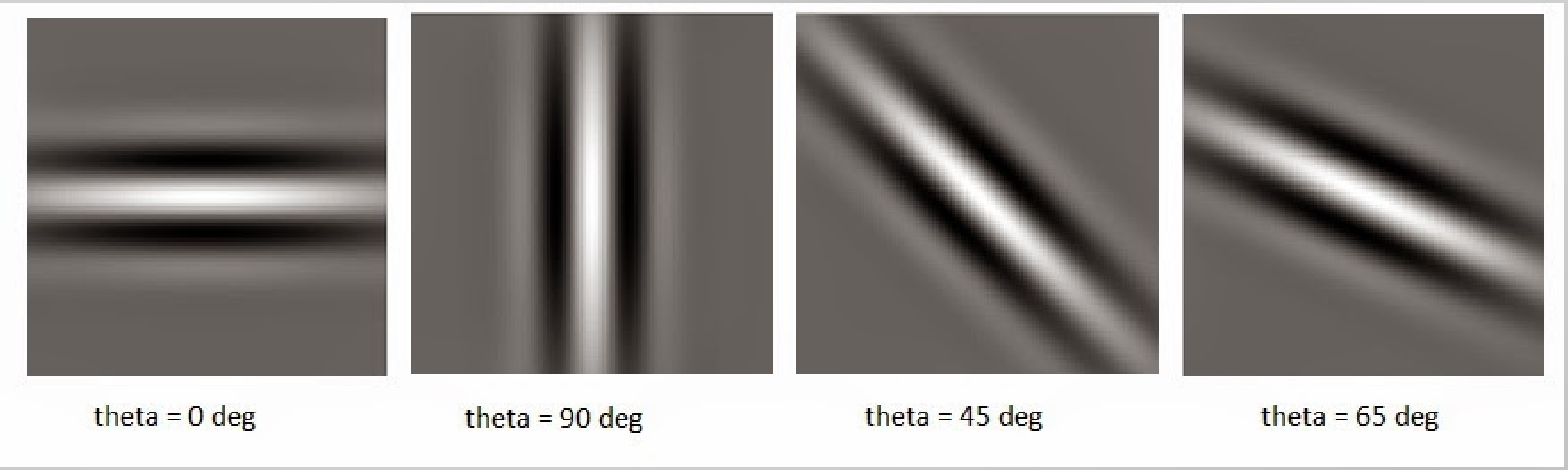


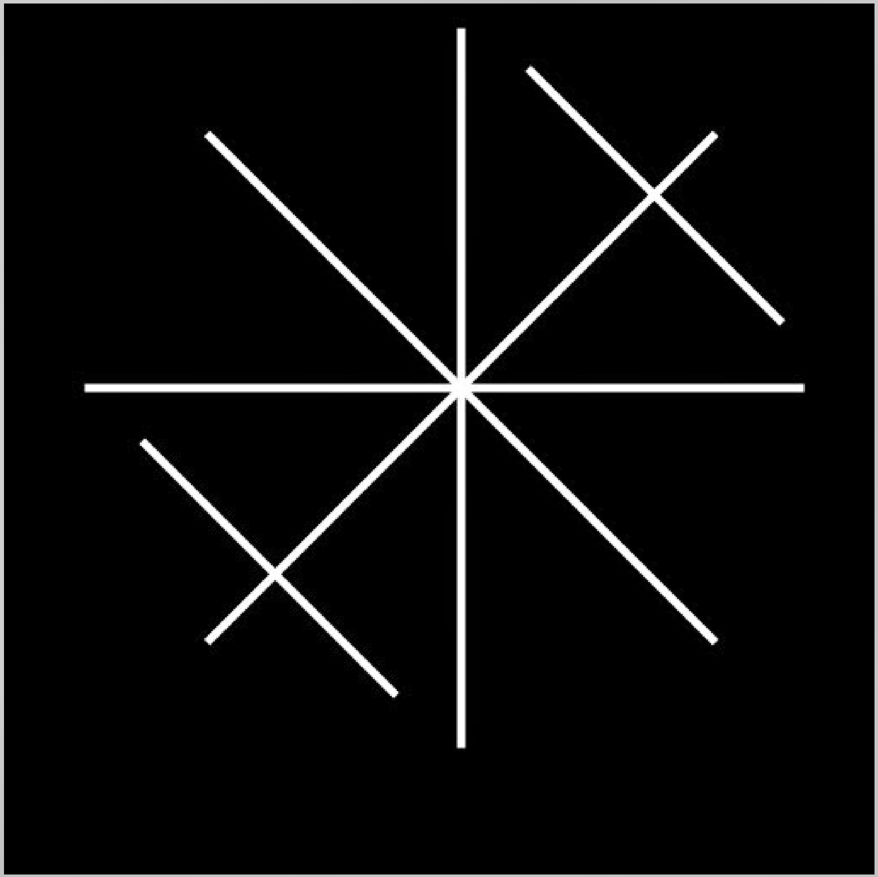
Gaussian envelope for different values of theta

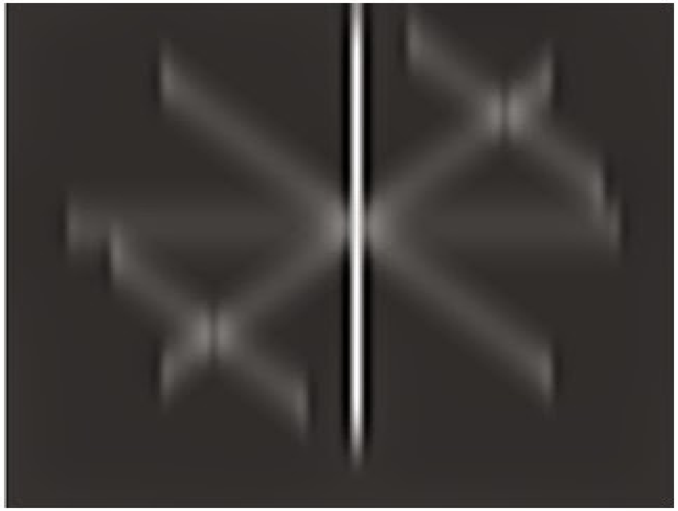


theta = 0 deg theta = 90 deg theta = 45 deg theta = 65 deg

Gabor Kernel for different thetas





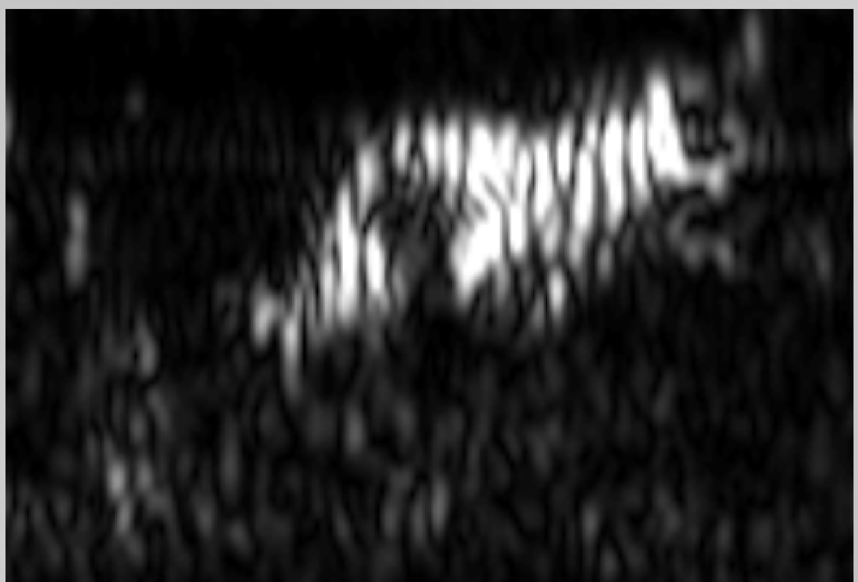


theta = 90 deg

theta = 135 deg

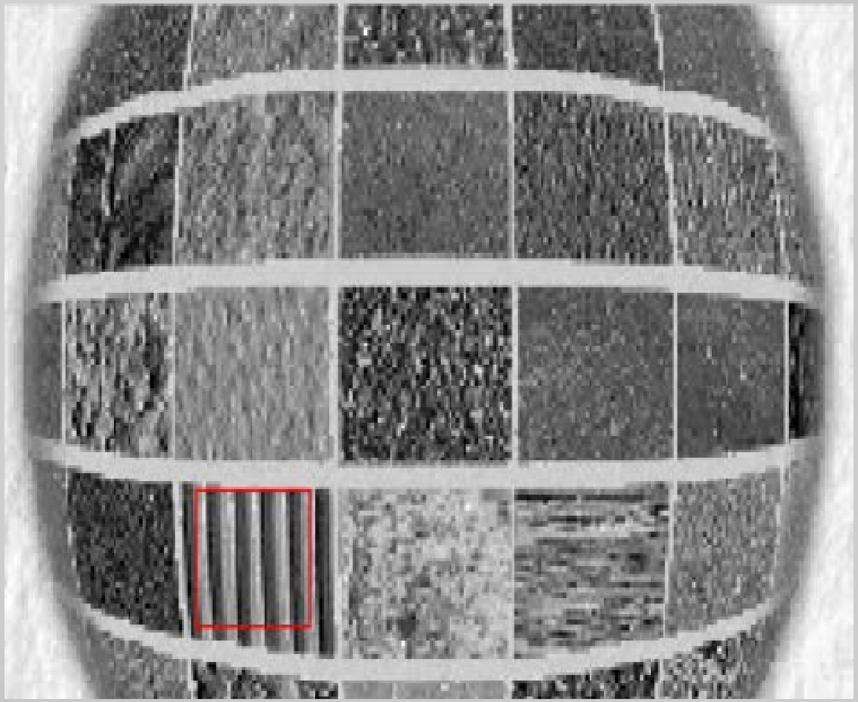
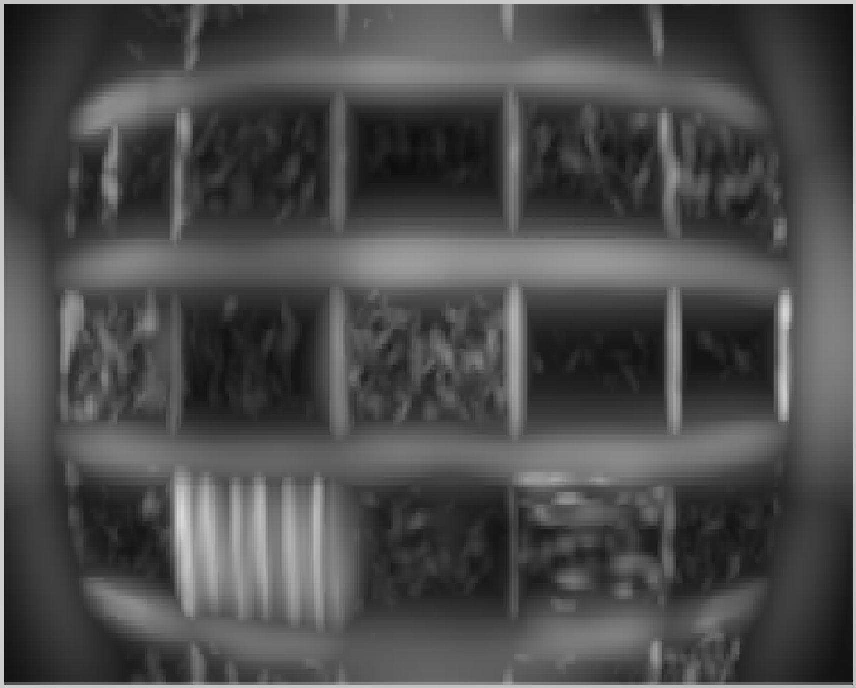
theta = 0 deg

theta = 45 deg



Gabor Space

- Typically, a bank of multiple Gabor filters is generated by varying the frequency and orientation of a mother wavelet.
- These filters are then convolved with the signal image, yielding a Gabor space.
- To put it simply, values in Gabor space represent the level of activation (i.e., response amplitude) for each filter at each spatial location in the signal image.

Applications

- It is believed that the first stage of visual processing in the brain, called area V1, implements a "filter bank" of "Gabor filters“
- Texture analysis and retrieval
- Facial expression recognition
- Text-only region recognition

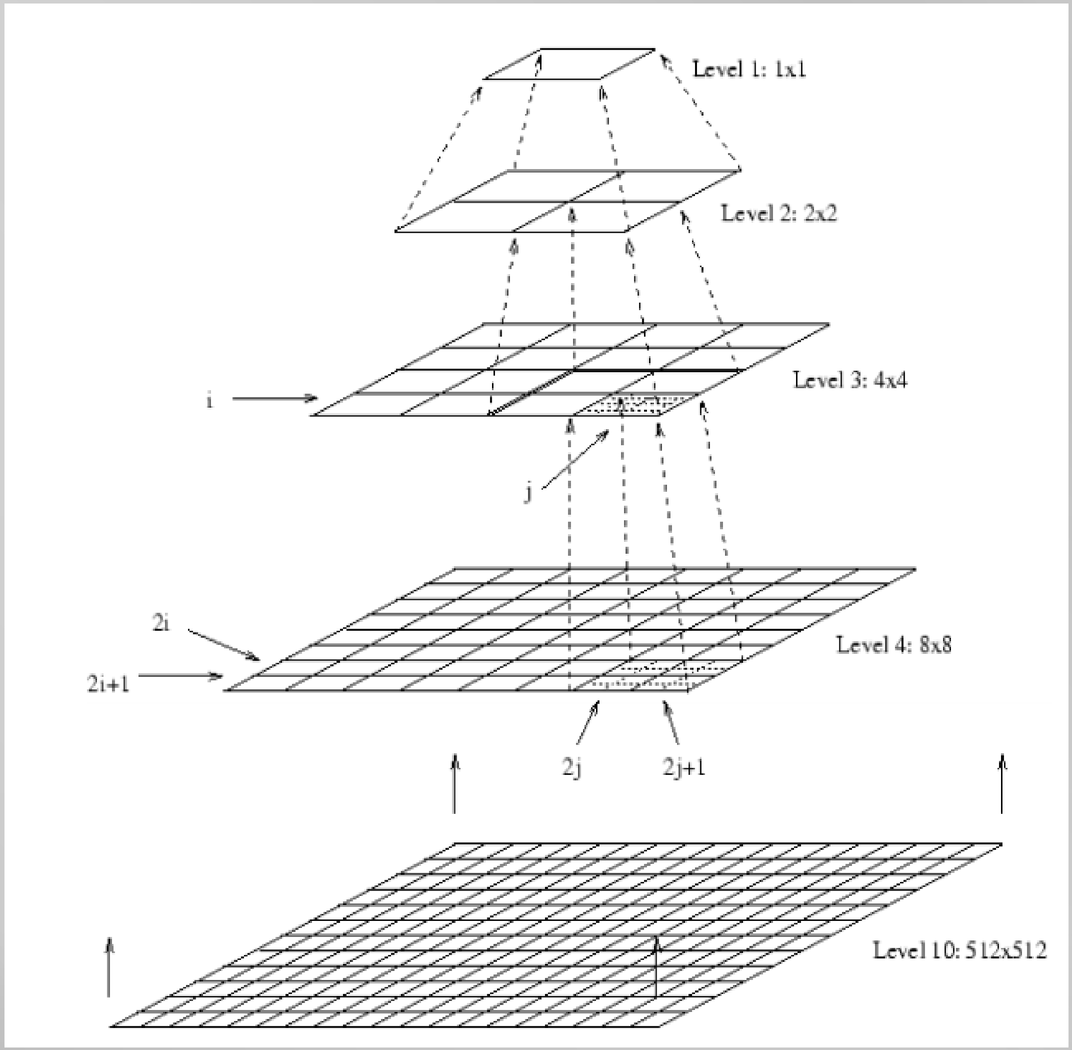
Resources and References

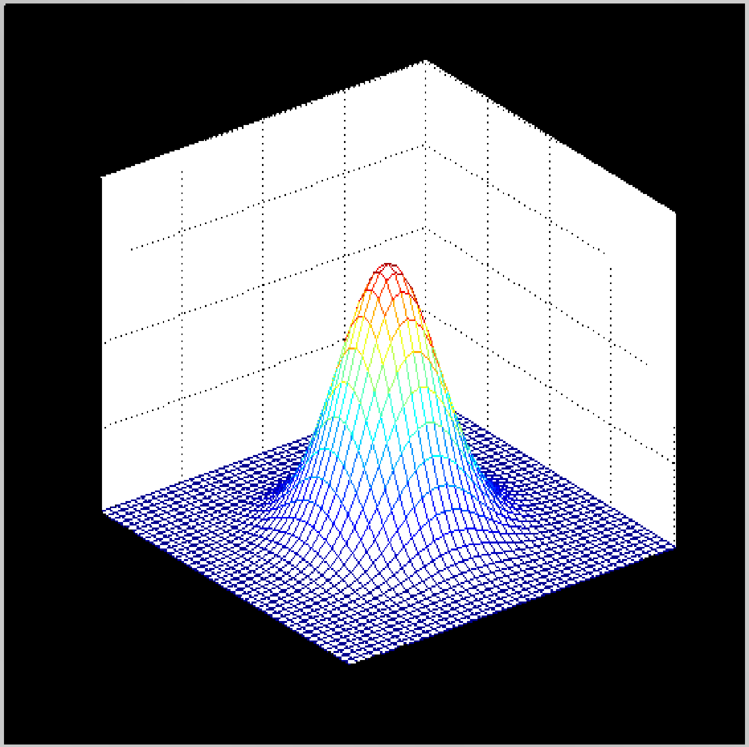
- An interactive tool for visualizing Gabor filters." [Online]. Available: http://www.cs.rug.nl/imaging/simplecell.html
- http://www.mathworks.com/matlabcentral/fileexchange/44630-gabor-feature-extraction
- Fogel, I.; Sagi, D. (1989). "Gabor filters as texture discriminator". Biological Cybernetics
- Wenfei Gu, Cheng Xiang, Y. V. Venkatesh, Dong Huang, Hai Lin, "Facial expression recognition using radial encoding of local Gabor features and classifier synthesis," Pattern Recognition, 45(1): 80-91, 2012
- S Sabari Raju, Peeta Basa Pati and A G Ramakrishnan, "Gabor Filter Based Block Energy Analysis for Text extraction from Digital Document Images," Proc. First International Workshop on Document Image Analysis for Libraries (DIAL-04), Palo Alto, USA, Jan. 2004, pp. 233-243.
- S Sabari Raju, P B Pati and A G Ramakrishnan, "Text Localization and Extraction from Complex Color Images," Proc. First International Conference on Advances in Visual Computing (ISVC05), Nevada, USA, LNCS 3804, Springer Verlag, Dec. 5-7, 2005, pp. 486-493.
- D. J. Gabor, "Theory of communication," IEEE,vol. 93, no. 26, pp. 429–457, 1946.

Pyramid

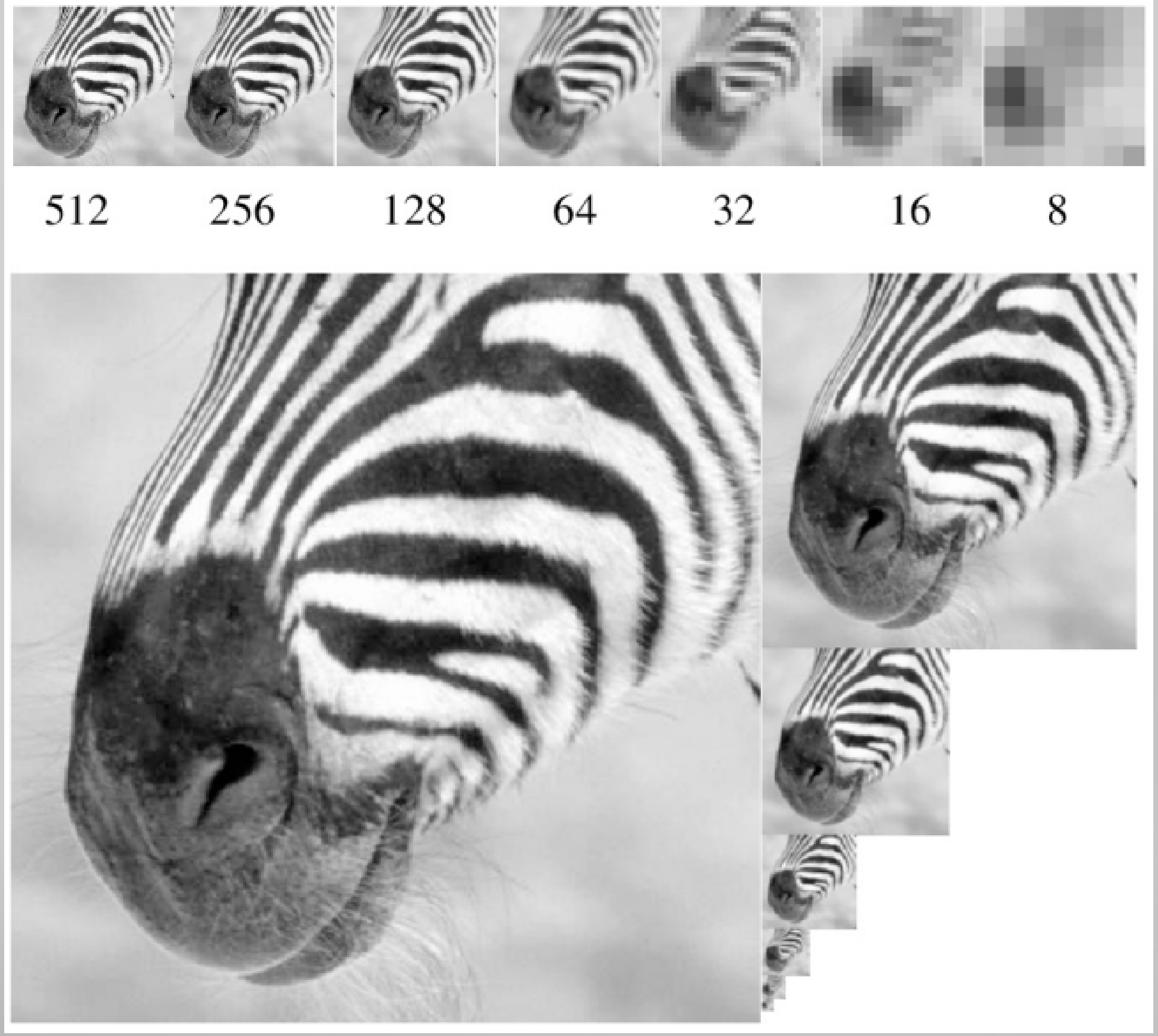
Gaussian Pyramid

- 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



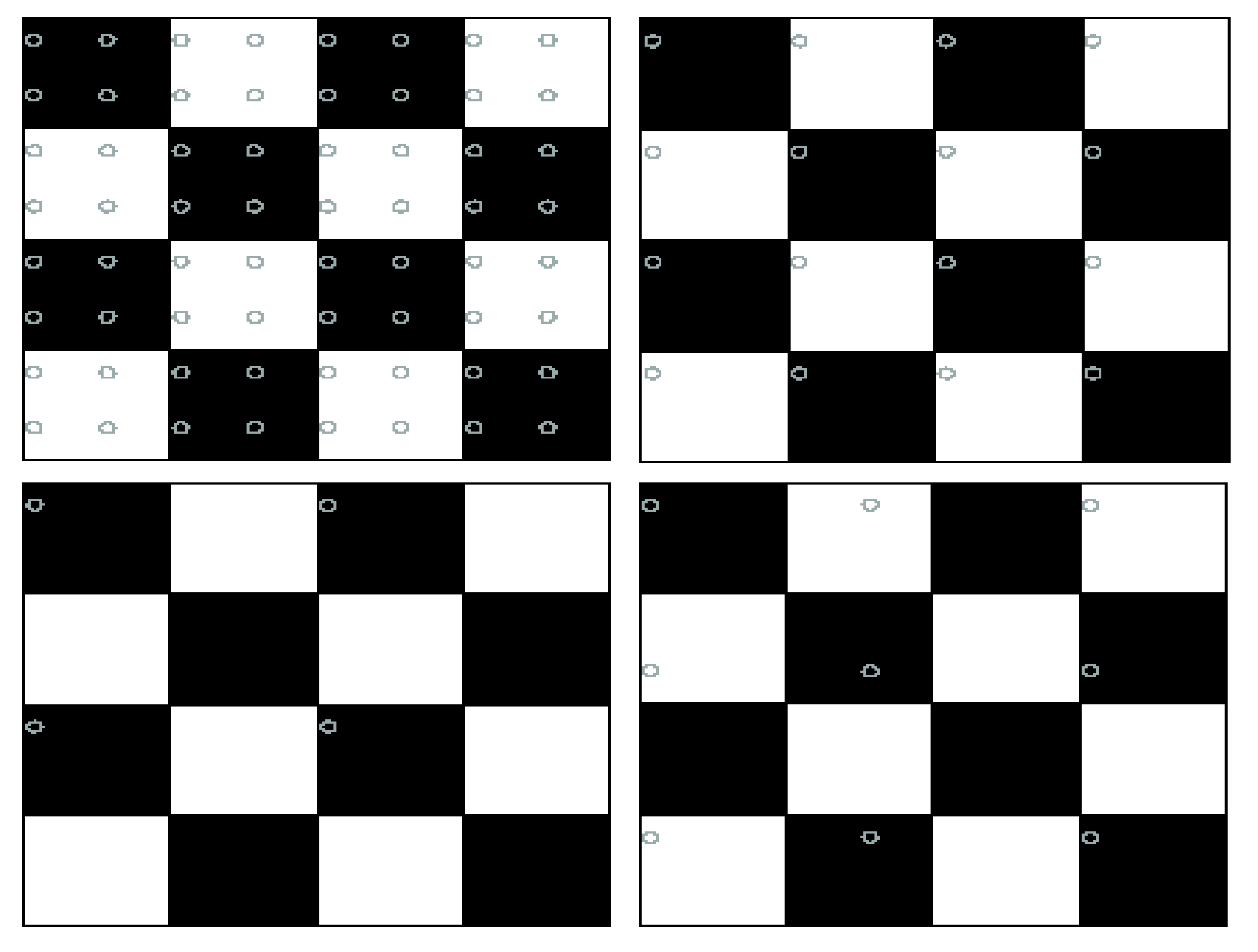


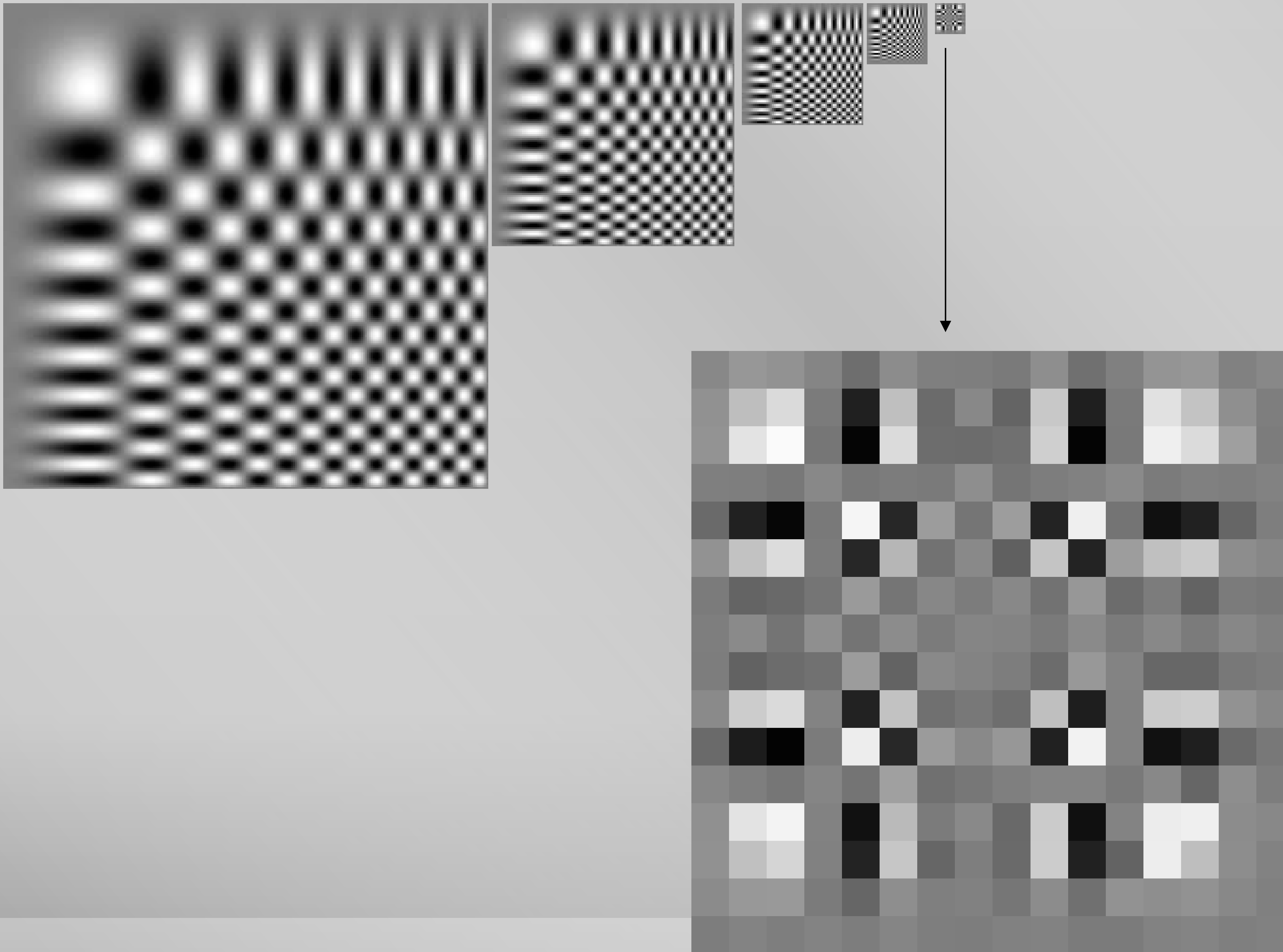
$$H(i,j)=\frac{1}{2\pi\sigma^2}\exp\left(-\frac{\left((i-k-1)^2+(j-k-1)^2\right)}{2\sigma^2}\right)$$


512
256
128
64
32
16
8

Aliasing

- Can’t shrink an image by taking every second pixel
- If we do, characteristic errors appear
- Typically, small phenomena look bigger; fast phenomena can look slower
- Wagon wheels rolling the wrong way in movies
- Checkerboards misrepresented in ray tracing
- Striped shirts look funny on colour television





Synthesis

- smooth and sample
- gaussians are low pass filters
- a gaussian*gaussian=another gaussian

Operations

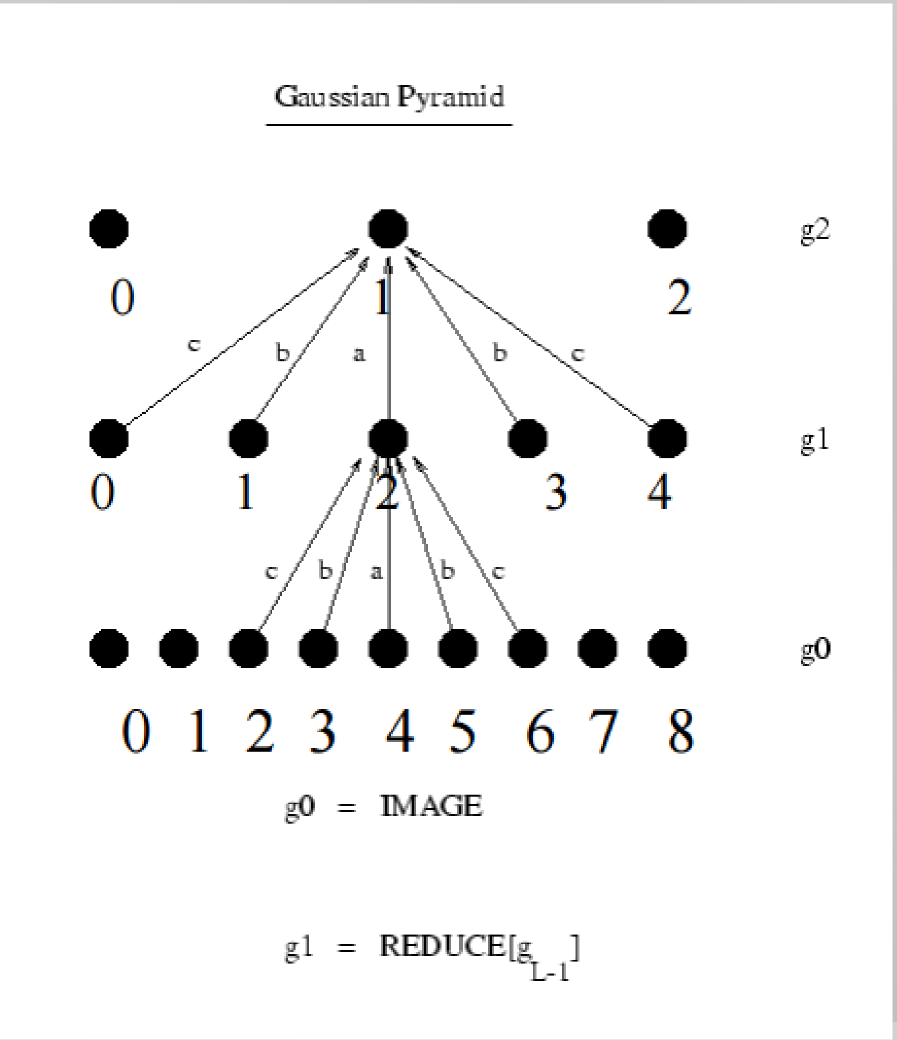
- Reduce

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

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

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

1 D case



Gaussian Pyramid
g2
0
1
2
c
b
a
b
c
g1
0
1
2
3
4
c
b
a
b
c
g0
0 1 2 3 4 5 6 7 8
g0 = IMAGE
g1 = REDUCE[g L-1]

What about 2D?

- Separability of Gaussian

$$\hat{I}(x,y) = I(x,y) * G(x,y)$$

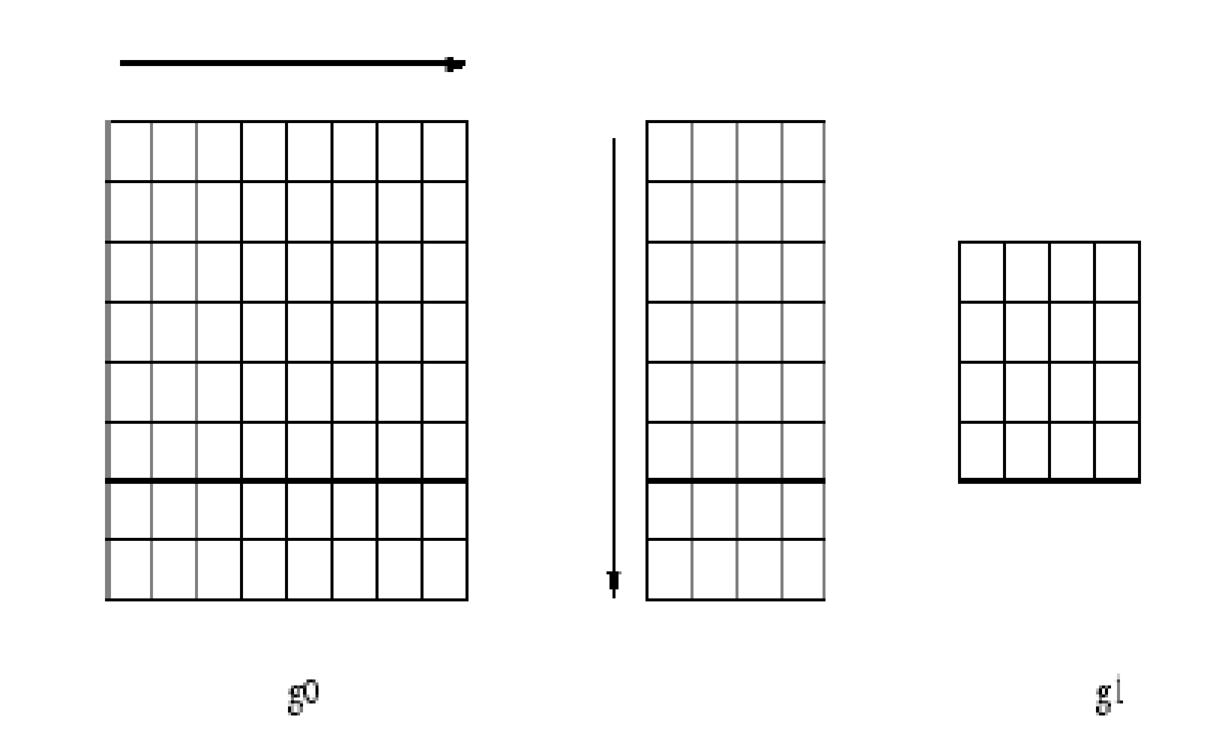
- Requires n2 k2 multiplications for n by n image and k by k kernel

$$\hat{I}(x,y) = I(x,y) * G(x) * G(y)$$

- Requires $2kn^2$ multiplications for n by n image and k by k kernel

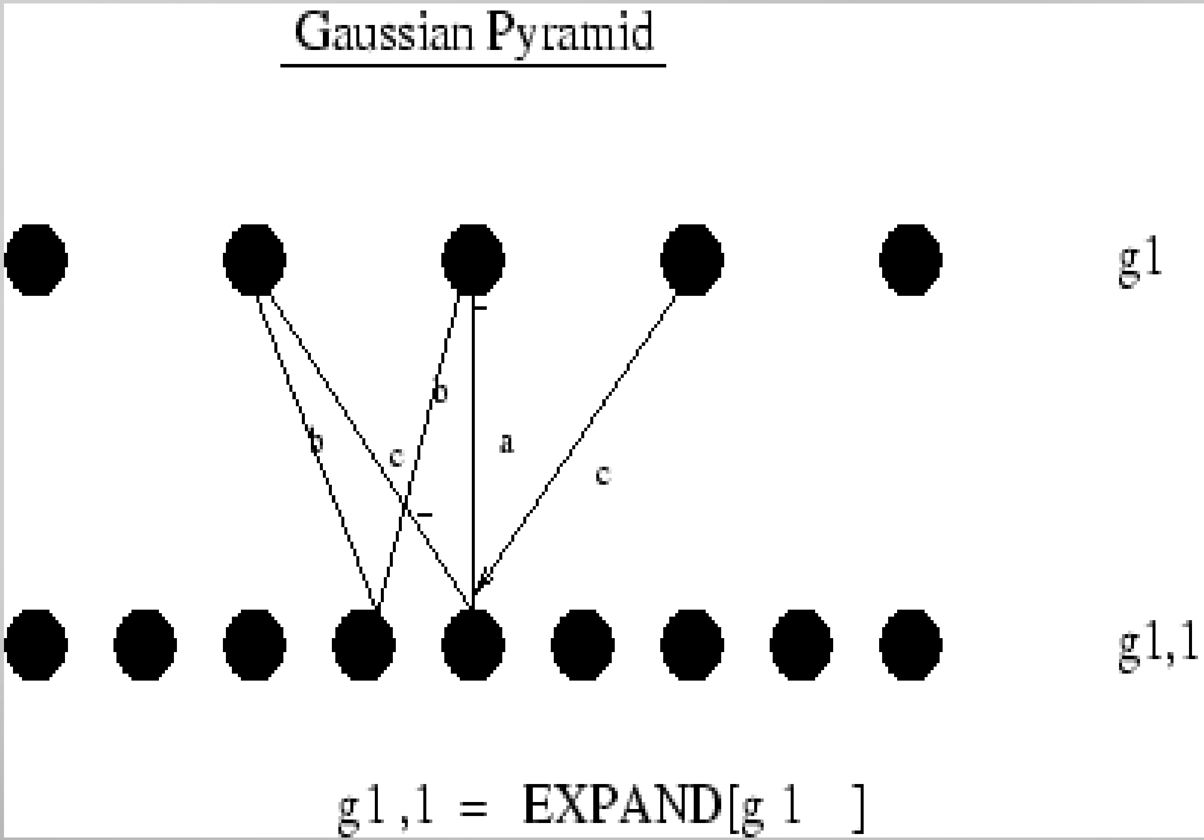
Algorithm

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



Operations

- Expand



Expand (1D)

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

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

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

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

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

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

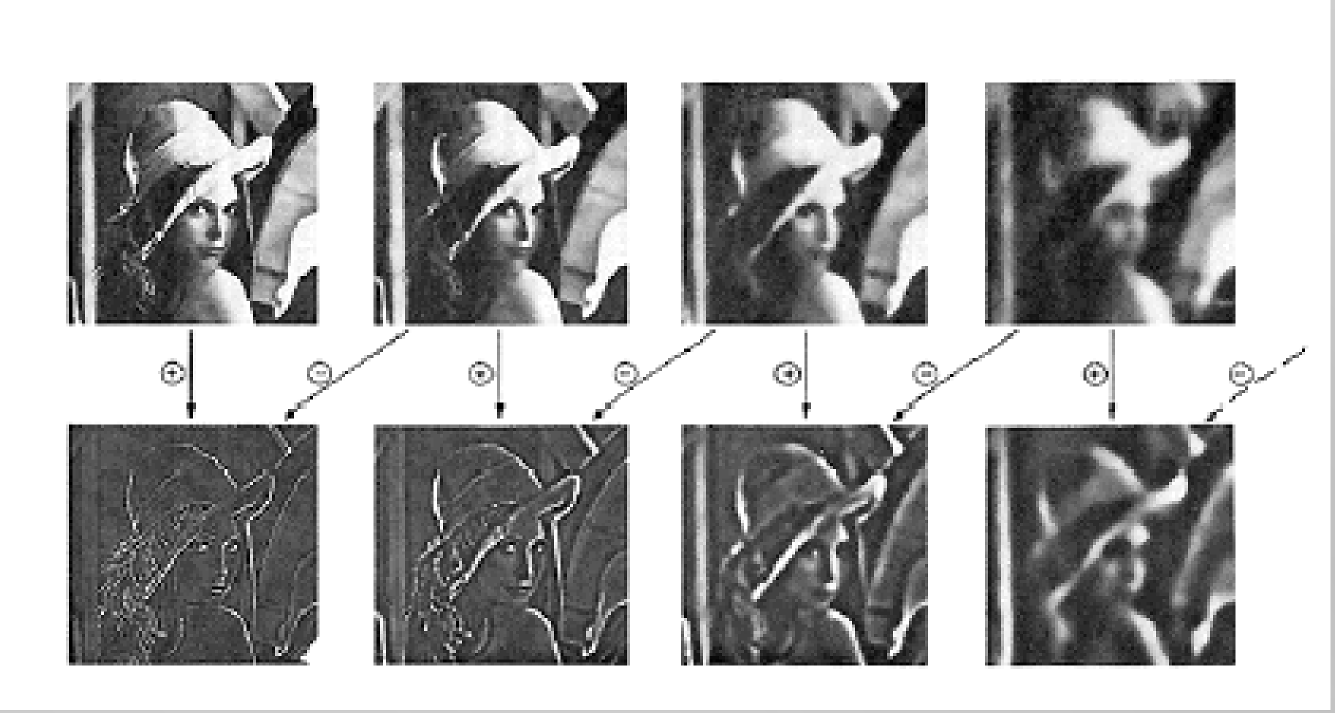
$$\hat{w}(0) g_{l,n-1}(\frac{3}{2}) + \hat{w}(1) g_{l,n-1}(\frac{3+1}{2}) + \hat{w}(2) g_{l,n-1}(\frac{3+2}{2})$$

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

Applications of Gaussian Pyramids

- Search for correspondence (Look at coarse scales, then refine with finer scales)
- Edge tracking (A “good” edge at a fine scale has parents at a coarser scale)
- Control of detail and computational cost in matching e.g. finding stripes
- Texture representation
- up- or down- sampling images.
- Multi-resolution image analysis
- Look for an object over various spatial scales
- Coarse-to-fine image processing: form blur estimate or the motion analysis on very low-resolution image, upsample and repeat. Often a successful strategy for avoiding local minima in complicated estimation tasks.

Laplacian Pyramid



Resources

- http://www.cns.nyu.edu/~eero/software.html
- http://in.mathworks.com/help/images/ref/impyramid.html
- Use the OpenCV functions pyrUp and pyrDown to downsample or upsample a given image

References

- Burt, Peter and Adelson, Ted, "The Laplacian Pyramid as a Compact Image Code", IEEE Trans. Communications, 9:4, 532–540, 1983.
- Crowley, J. L. and Sanderson, A. C. "Multiple resolution representation and probabilistic matching of 2-D gray-scale shape", IEEE Transactions on Pattern Analysis and Machine Intelligence, 9(1), pp 113-121, 1987.
- Lindeberg, Tony, "Scale-space for discrete signals," PAMI(12), No. 3, March 1990, pp. 234-254.
- Lindeberg, Tony. Scale-Space Theory in Computer Vision, Kluwer Academic Publishers, 1994, ISBN 0-7923-9418-6 (see specifically Chapter 2 for an overview of Gaussian and Laplacian image pyramids and Chapter 3 for theory about generalized binomial kernels and discrete Gaussian kernels)
- Manduchi, Roberto; Perona, Pietro; Shy, Doug (1997). "Efficient Deformable Filter Banks" (PDF). California Institute of Technology/University of Padua.
- Crowley, J, Riff O. Fast computation of scale normalised Gaussian receptive fields, Proc. Scale-Space'03, Isle of Skye, Scotland, Springer Lecture Notes in Computer Science, volume 2695, 2003.
- Lowe, D. G. (2004). "Distinctive image features from scale-invariant keypoints". International Journal of Computer Vision 60 (2): 91–110. doi:10.1023/B:VISI.0000029664.99615.94.