Schedule for Rest of Semester

Date	Lecture	Topic
11/20	24	Texture
11/27	25	Review of Statistics & Linear Algebra, Eigenvectors
11/29	26	Eigenvector expansions, Pattern Recognition
12/4	27	Cameras & calibration
12/4		Project due
12/6	28	Kalman filters & tracking

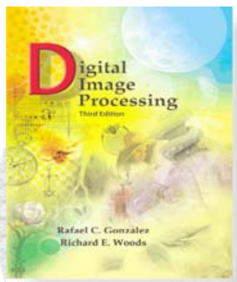
Project Image

Image is 5.6 MB compressed (157 MB uncompressed) 7608x7244 pixels

http://vorlon.case.edu/~flm/eecs490f06/Images/Images.html

Lecture #24

- What is texture?
- Texture issues: analysis,synthesis, segmentation, shape
- Filters and texture, texture recognition, spot and bar filters, filter banks
- Texture measures
- Scale: Gaussian and Laplacian pyramids
- Gabor filters
- Texture gradient
- Texture synthesis



Texture

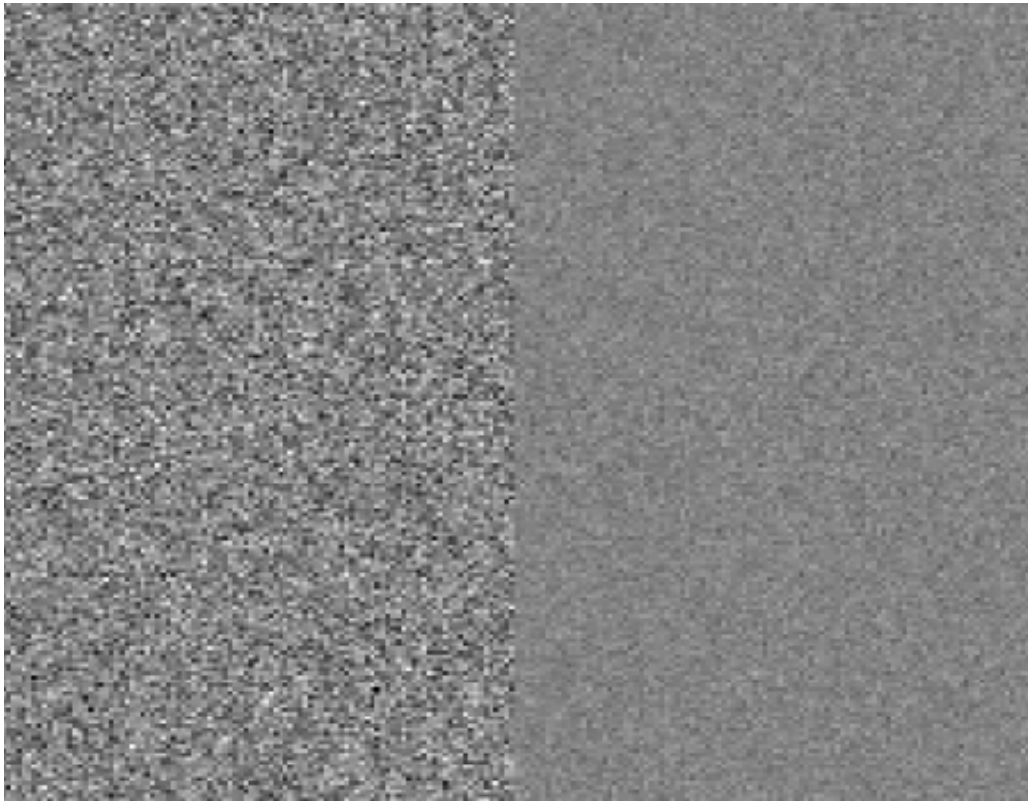
- Goal of computer vision: infer things about the world by looking at one or more images
- Geometry provides clues
- Image features provide clues
  - Edges
  - Corners
  - Filter Responses
- What next?
  - Texture
  - What is texture?

Texture

- Edge detectors find differences in overall intensity.
- Average intensity is only the simplest difference

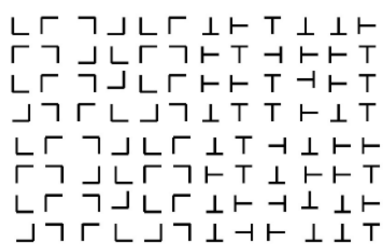
Texture

These two regions clearly have different textures.



What is Texture?

- Something that repeats with variation
- Must separate what repeats and what stays the same.
- One model for texture is as repeated trials of a random process
  - The probability distribution stays the same.
  - But each trial is different.

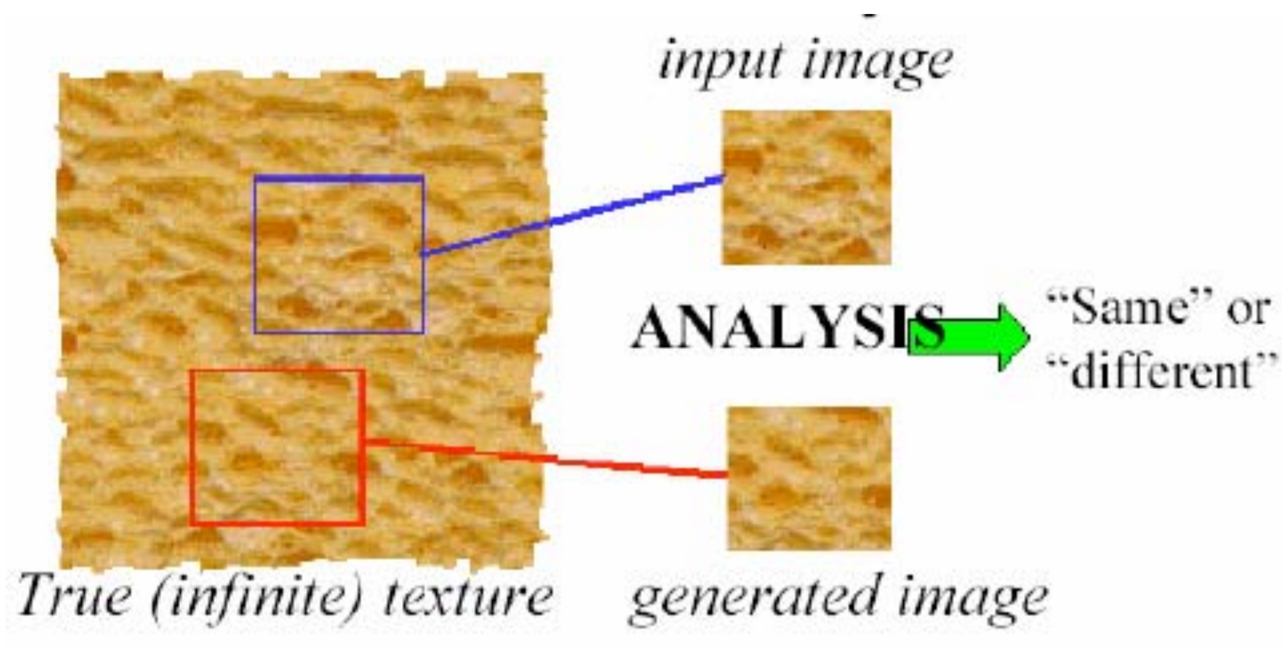


Texture Issues

1. Discrimination/Analysis
2. Synthesis
3. Texture segmentation and boundary detection
4. Shape from texture

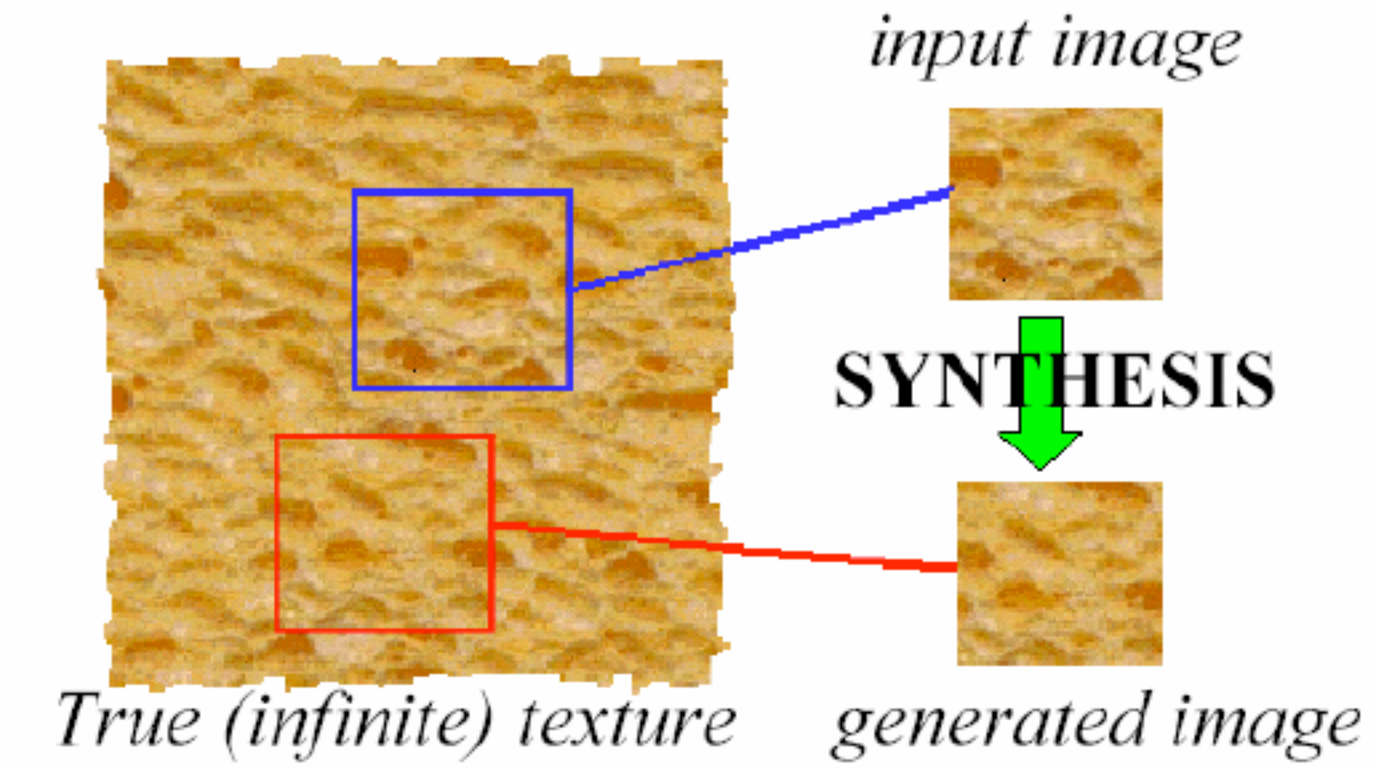
Texture Discrimination/Analysis

The goal of texture analysis is to compare textures and decide if they're made of the same "stuff"



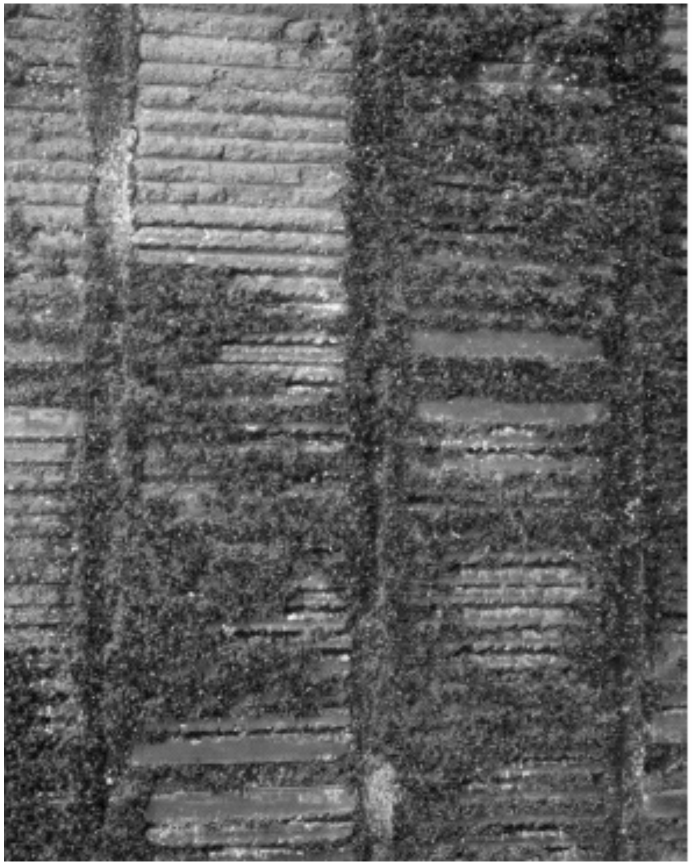
Texture Synthesis

The goal of texture synthesis is to create textures for regions

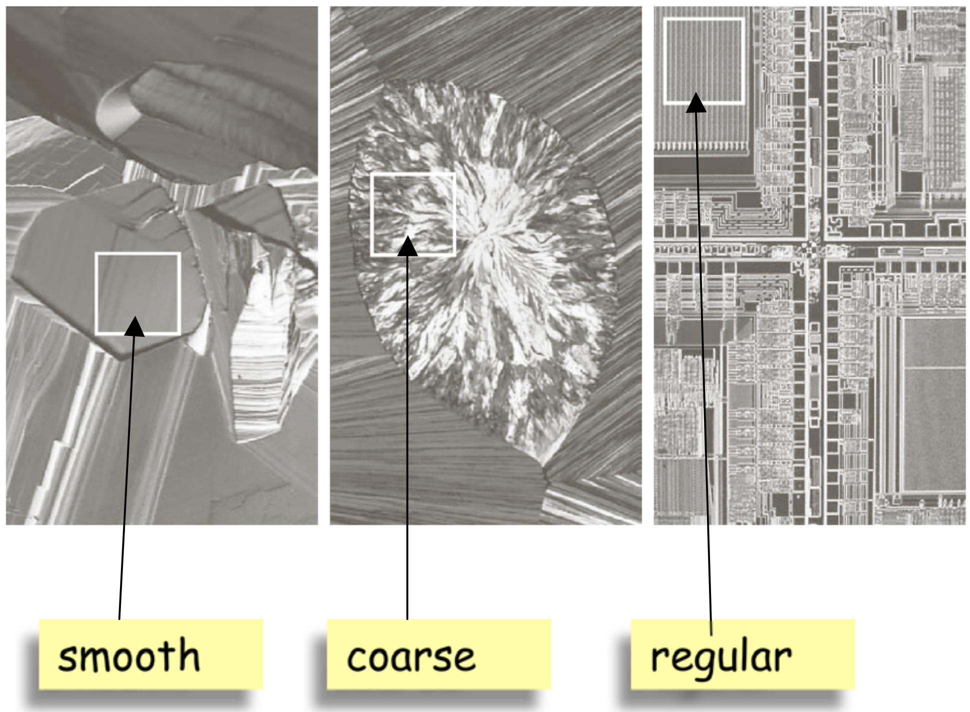


Texture

Textures are made of sub-elements arranged in patterns.



Texture



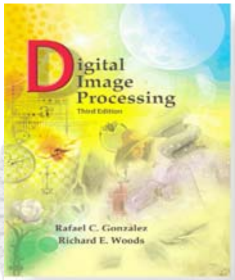
a b c

FIGURE 11.28 The white squares mark, from left to right, smooth, coarse, and regular textures. These are optical microscope images of a superconductor, human cholesterol, and a microprocessor. (Courtesy of Dr. Michael W. Davidson, Florida State University.)

These patterns are more readily apparent.

Texture Description

- Textons are the patterns of sub-elements that produce texture
- There is no canonical set of textons for texture representation and analysis such as the sinusoids for a Fourier series representation
- We can use filters to analyze textures



Filters for pattern recognition

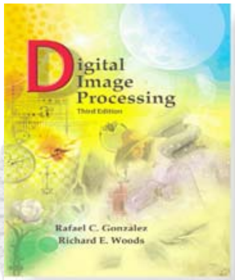
A filter responds most strongly to pattern elements that look like the filter. Consider the response of a filter h to image f_2

$$f_2(m,n) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f_1(i,j) h(m-i, n-j)$$

Now let m and n equal zero

$$f_2(0,0) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f_1(i,j) h(-i,-j)$$

The response is the dot product of the filter and the image. The difference between simple convolution (filtering) and correlation (a dot product) is the sign of the coordinates and the fact that the correlation is being translated over the image

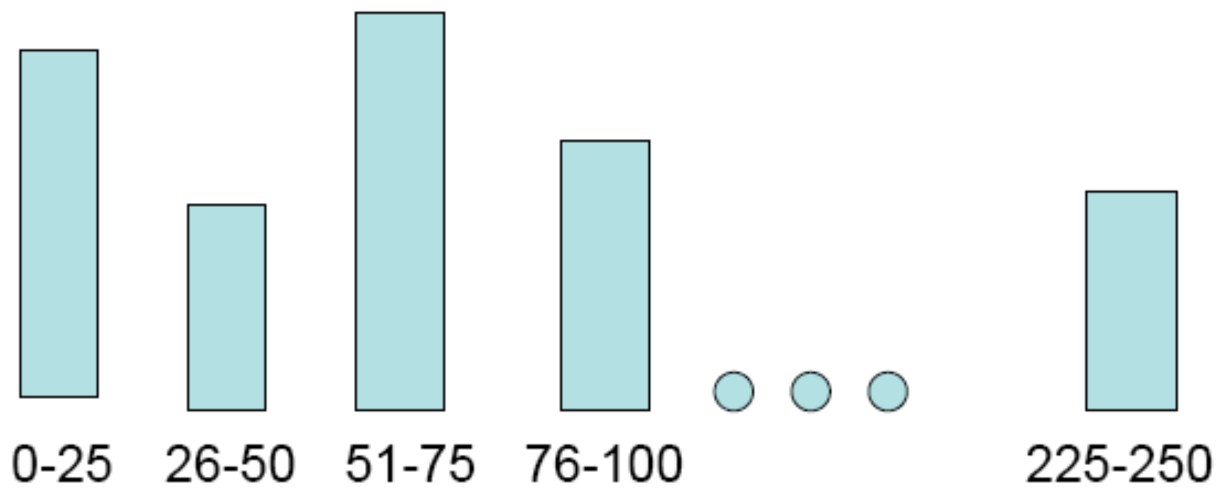


Texture Recognition

- Statistical
- Structural
- Histogram
- Fourier Transform
- Sets of filters

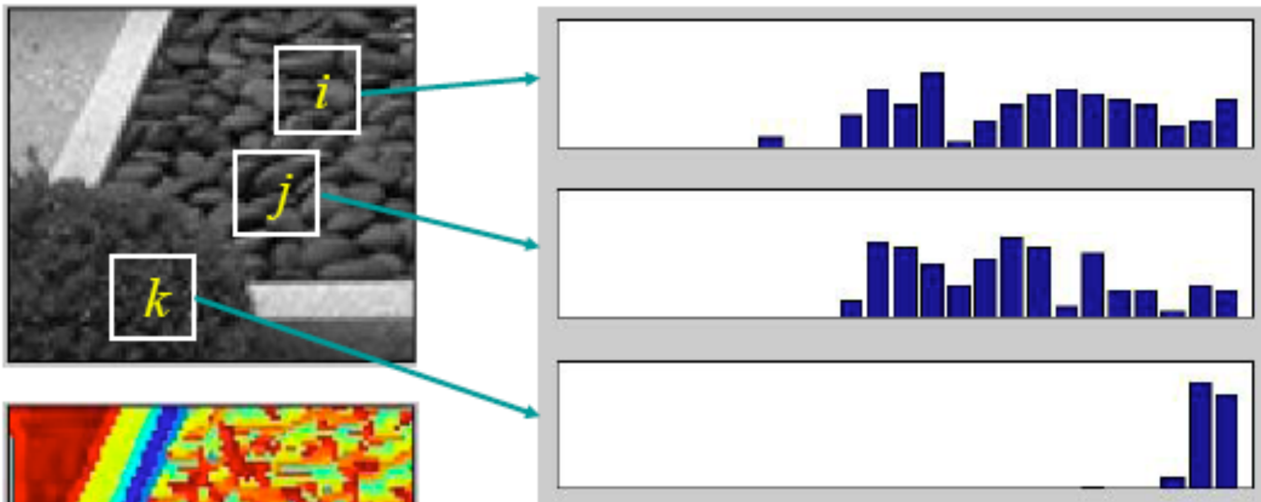
A Simple Texture Measure

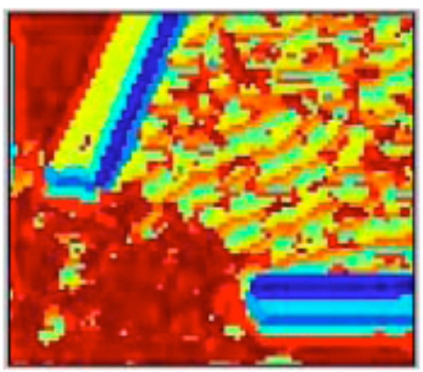
- Compare histograms.
  - Divide intensities into discrete ranges.
  - Count how many pixels in each range



Texture Measure

- Chi square distance between texton histograms





$$\chi^2(h_i,h_j)=\frac{1}{2}\sum_{m=1}^{K}\frac{[h_i(m)-h_j(m)]^2}{h_i(m)+h_j(m)}$$

More complex measures can use scale and orientation

(Malik)

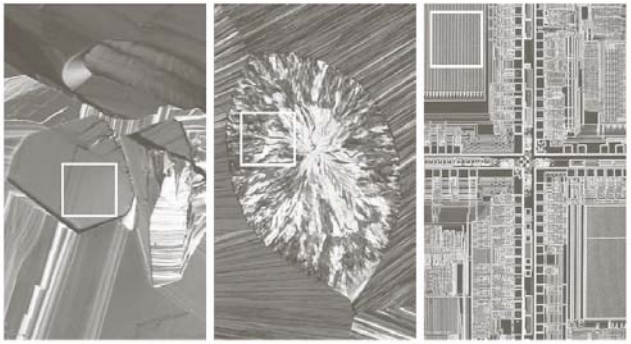
Local Statistics

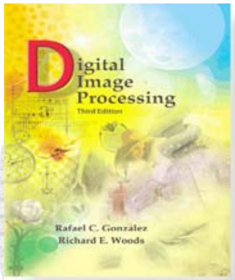
Statistics of each patch

Texture	Mean	Standard deviation	*R* (normalized)	Third moment	Uniformity	Entropy
Smooth	82.64	11.79	0.002	−0.105	0.026	5.434
Coarse	143.56	74.63	0.079	−0.151	0.005	7.783
Regular	99.72	33.73	0.017	0.750	0.013	6.674

TABLE 11.2 Texture measures for the subimages shown in Fig. 11.28.

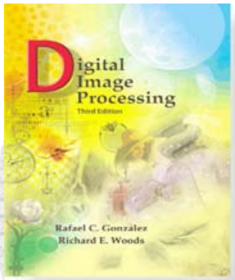
Mean - gives an idea of the average gray level and not texture
Standard deviation - some measure of smoothness
R (normalized) essentially the same as standard deviation
Third moment - are gray levels biased towards dark or light?
Uniformity -
Entropy - measure of variability





Texture

- Mean $$m = \sum_{i=0}^{L-1} z_i p(z_i)$$
- Variance $$\sigma^2(z) = \mu_2(z) = \sum_{i=0}^{L-1} (z_i - m)^2 p(z_i)$$
- R (normalized) $$R = 1 - \frac{1}{1 + \sigma^2(z)}$$
- Third moment $$\mu_3(z) = \sum_{i=0}^{L-1} (z_i - m)^3 p(z_i)$$
- Uniformity $$U = \sum_{i=0}^{L-1} p^2(z_i)$$
- Avg. Entropy $$e = -\sum_{i=0}^{L-1} p(z_i) \ln\left(p(z_i)\right)$$



Gray-level Co-Occurrence

Unlike a histogram a co-occurrence matrix has a sense of relative position.

1	1	7	5	3	2
5	1	6	1	2	5
8	8	6	8	1	2
4	3	4	5	5	1
8	7	8	7	6	2
7	8	6	2	6	2

Image f

	1	2	3	4	5	6	7	8
1	1	2	0	0	0	1	1	0
2	0	0	0	0	1	1	0	0
3	0	1	0	1	0	0	0	0
4	0	0	1	0	1	0	0	0
5	2	0	1	0	1	0	0	0
6	1	3	0	0	0	0	0	1
7	0	0	0	0	1	1	0	2
8	1	0	0	0	0	2	2	1

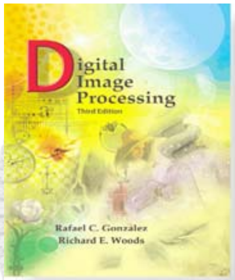
Co-occurrence matrix **G**

FIGURE 11.29 How to generate a co-occurrence matrix.

Reduce # of gray levels to keep size of Q reasonable.

Q is a position operator which defines the relationship between two pixels. In this case, <u>one pixel immediately to the right</u>. This is a gray level co-occurrence matrix.

The co-occurrence matrix G tabulates the number of times that pixel pairs with intensities z_i and z_j occur in f in the position specified by Q. For example g_{11} which is 1 to the right of 1 only occurs once in f. We can normalize the elements of G by the total number of pixel pairs, i.e., the sum of the elements in G.



Representation and Description

m_r, σ_r and m_c, σ_c are statistics for that row and column of G

Descriptor	Explanation	Formula
Maximum probability	Measures the strongest response of **G**. The range of values is [0, 1].	$\max_{i,j}(p_{ij})$
Correlation	A measure of how correlated a pixel is to its neighbor over the entire image. Range of values is 1 to −1, corresponding to perfect positive and perfect negative correlations. This measure is not defined if either standard deviation is zero.	$\sum_{i=1}^{K}\sum_{j=1}^{K}\frac{(i-m_r)(j-m_c)p_{ij}}{\sigma_r\sigma_c}$ $\sigma_r \neq 0;\ \sigma_c \neq 0$
Contrast	A measure of intensity contrast between a pixel and its neighbor over the entire image. The range of values is 0 (when **G** is constant) to $(K-1)^2$.	$\sum_{i=1}^{K}\sum_{j=1}^{K}(i-j)^2 p_{ij}$
Uniformity (also called Energy)	A measure of uniformity in the range [0, 1]. Uniformity is 1 for a constant image.	$\sum_{i=1}^{K}\sum_{j=1}^{K}p_{ij}^2$
Homogeneity	Measures the spatial closeness of the distribution of elements in **G** to the diagonal. The range of values is [0, 1], with the maximum being achieved when **G** is a diagonal matrix.	$\sum_{i=1}^{K}\sum_{i=1}^{K}\frac{p_{ij}}{1+\lvert i-j\rvert}$
Entropy	Measures the randomness of the elements of **G**. The entropy is 0 when all p_{ij}'s are 0 and is maximum when all p_{ij}'s are equal. The maximum value is $2\log_2 K$. (See Eq. (11.3-9) regarding entropy).	$-\sum_{i=1}^{K}\sum_{i=1}^{K}p_{ij}\log_2 p_{ij}$

TABLE 11.3 Descriptors used for characterizing co-occurrence matrices of size $K \times K$. The term p_{ij} is the ijth term of **G** divided by the sum of the elements of **G**.

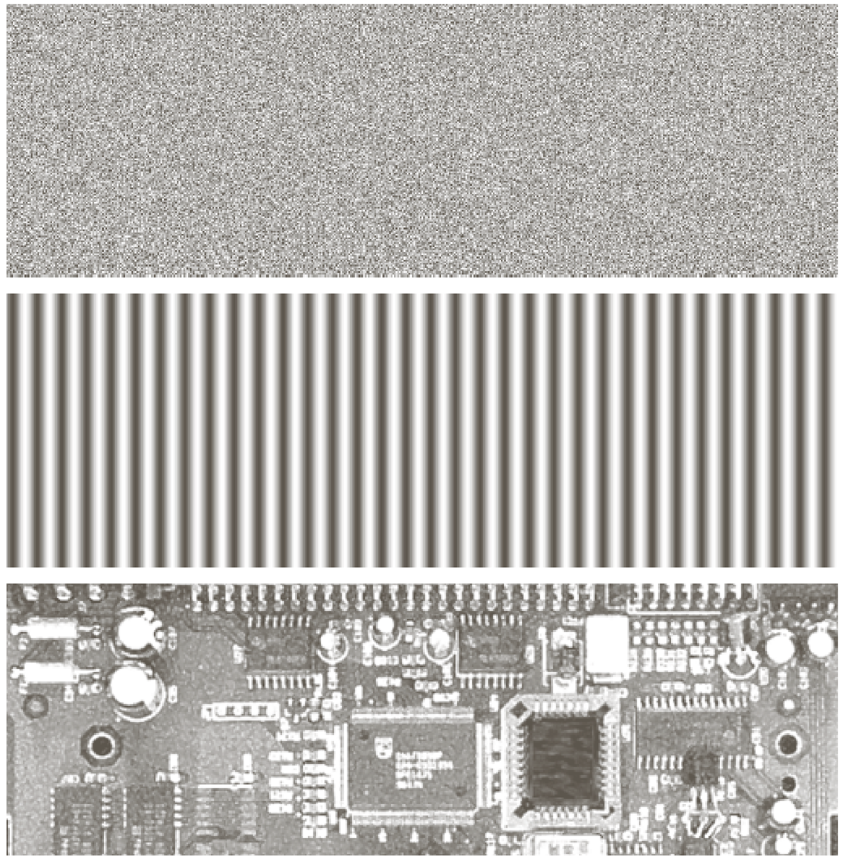
These are some of the measures used to characterize the normalized matrix G.

Representation and Description

Random texture

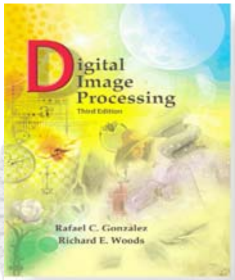
Periodic texture

Mixed texture

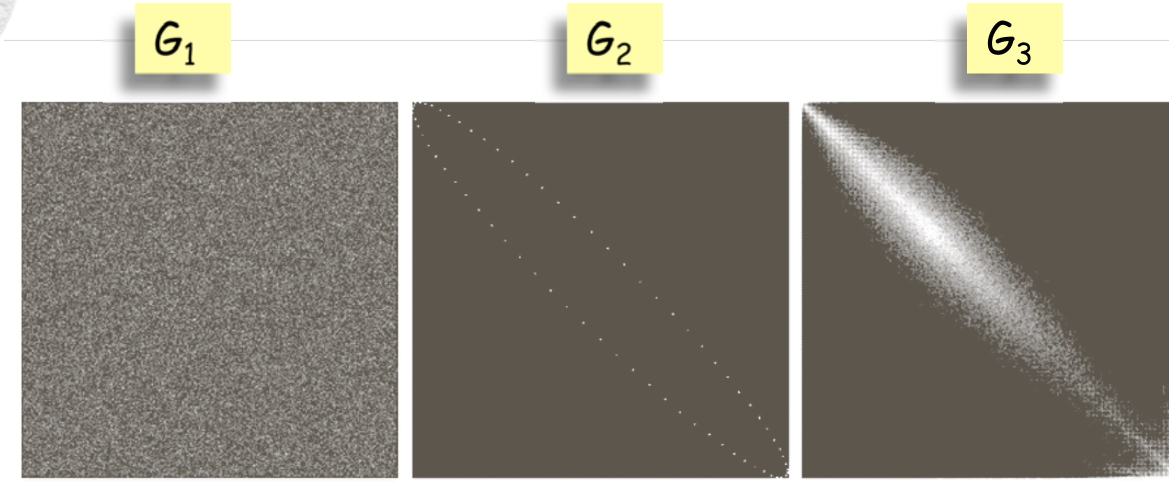


a
b
c

FIGURE 11.30 Images whose pixels have (a) random, (b) periodic, and (c) mixed texture patterns. Each image is of size 263×800 pixels.



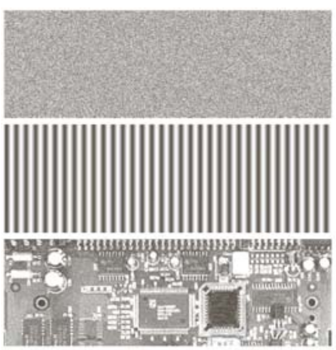
Representation and Description



a b c

FIGURE 11.31 256 × 256 co-occurrence matrices, $\mathbf{G}_1$, $\mathbf{G}_2$, and $\mathbf{G}_3$, corresponding from left to right to the images in Fig. 11.30.

Co-occurrence matrices G of the three pictures.



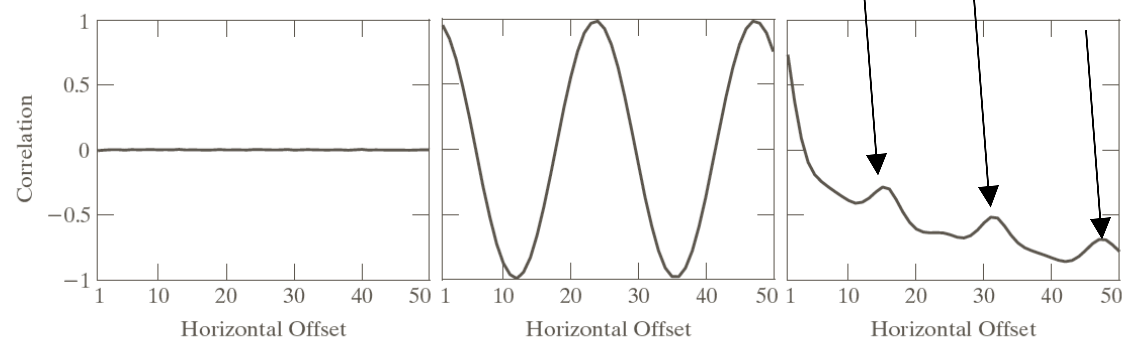
Normalized Co-occurrence Matrix	Max Probability	Correlation	Contrast	Uniformity	Homogeneity	Entropy
	Descriptor					
$\mathbf{G}_1/n_1$	0.00006	−0.0005	10838	0.00002	0.0366	15.75
$\mathbf{G}_2/n_2$	0.01500	0.9650	570	0.01230	0.0824	6.43
$\mathbf{G}_3/n_3$	0.06860	0.8798	1356	0.00480	0.2048	13.58

TABLE 11.4 Descriptors evaluated using the co-occurrence matrices displayed in Fig. 11.31.

Correlation

Peaks primarily due to basic 16 pixel spacing of circuit board connections.

Change the position operator in constructing the G matrix from 1 pixel to the right to 2, 3, … to the right. Compute the correlation of G for each offset and plot these.



a b c

FIGURE 11.32 Values of the correlation descriptor as a function of offset (distance between "adjacent" pixels) corresponding to the (a) noisy, (b) sinusoidal, and (c) circuit board images in Fig. 11.30.

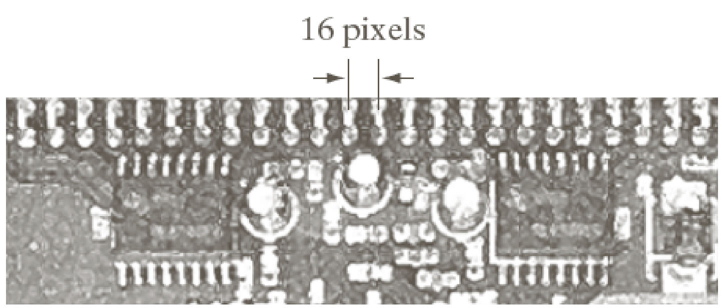
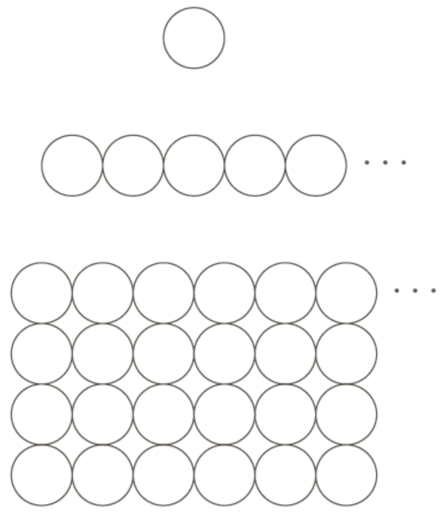


FIGURE 11.33 A zoomed section of the circuit board image showing periodicity of components.

Shape Grammars



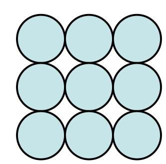
a
b
c

FIGURE 11.34 (a) Texture primitive. (b) Pattern generated by the rule $S \rightarrow aS$. (c) 2-D texture pattern generated by this and other rules.

Grammars are rules for constructing textures. In this case, this is a regular texture.

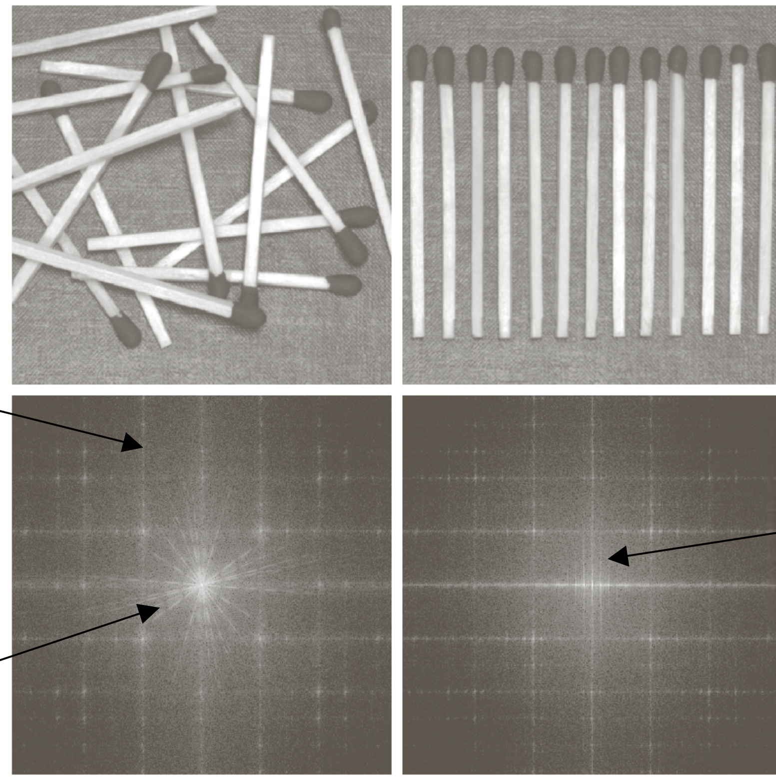
S->aS
S->bA (circle down)
A->cA (circle left)
A->c
A->bS
A->b
S->a

Starting with S=a can give aaabccbaa using the rules to the left.



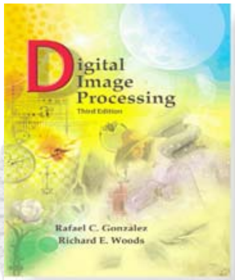
Terminated with the S->a rule

Fourier Description



a b
c d

FIGURE 11.35 (a) and (b) Images of random and ordered objects. (c) and (d) Corresponding Fourier spectra. All images are of size 600×600 pixels.



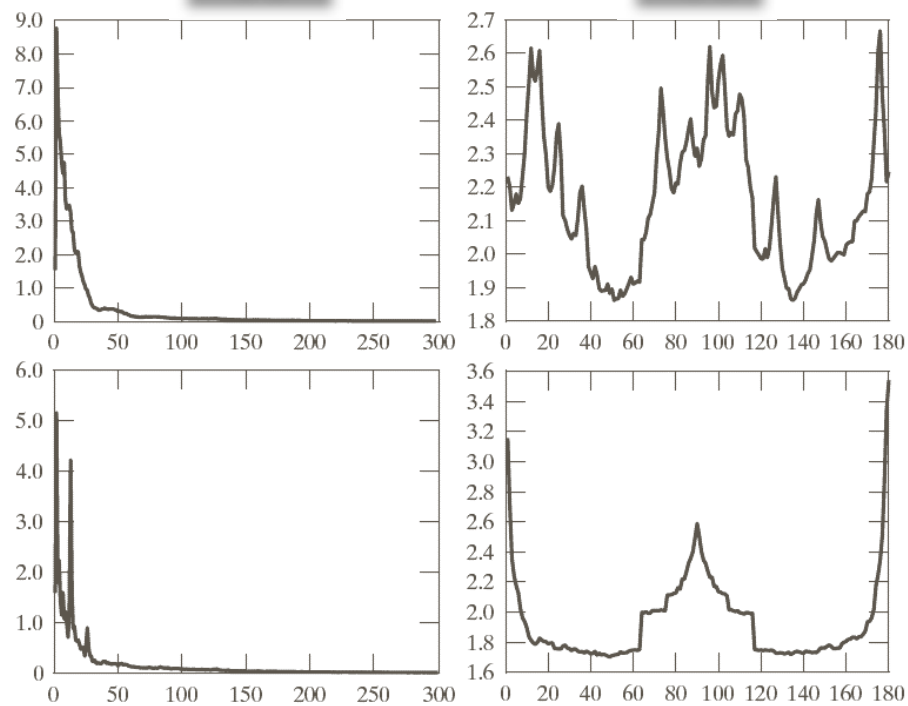
Fourier Description

$$S(r) = \sum_{\theta=0}^{\pi} S_\theta(r)$$

$S_\theta(r)$ is the Fourier transform along a particular direction θ

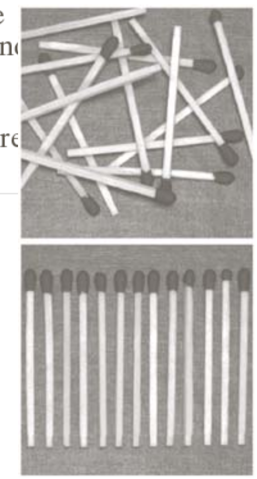
$$S(\theta) = \sum_{r=1}^{R_0} S_r(\theta)$$

$S_r(\theta)$ is the Fourier transform around a circle of radius r



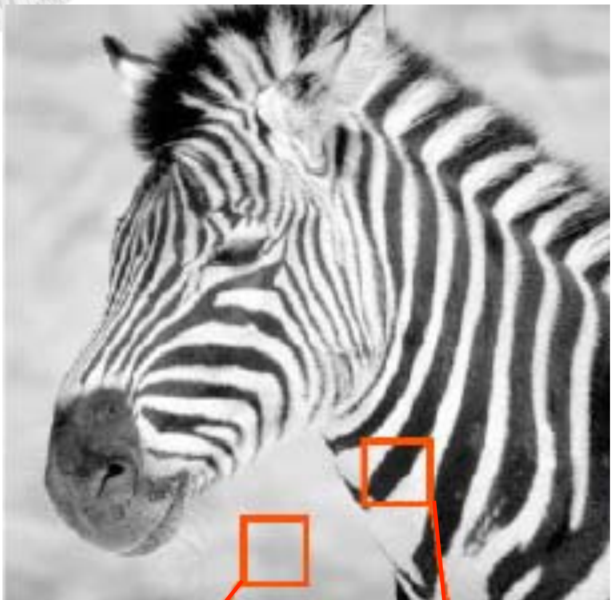
a b
c d

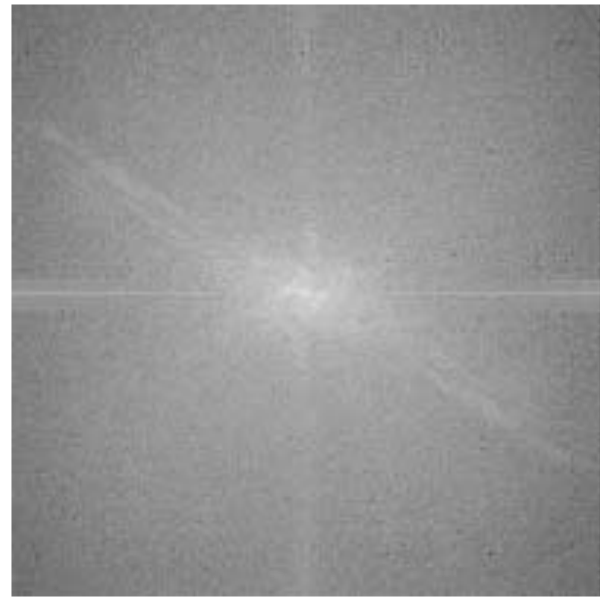
FIGURE 11.36 Plots of (a) $S(r)$ and (b) $S(\theta)$ for Fig. 11.35(a). (c) and (d) are plots of $S(r)$ and $S(\theta)$ for Fig. 11.35(b). All vertical axes are $\times 10^5$.



Write the Fourier spectrum in polar form to analyze the principal direction of the texture patterns. Figure (d) clearly shows the strong orientations near 0°, 90°, and 180°

Fourier Measures





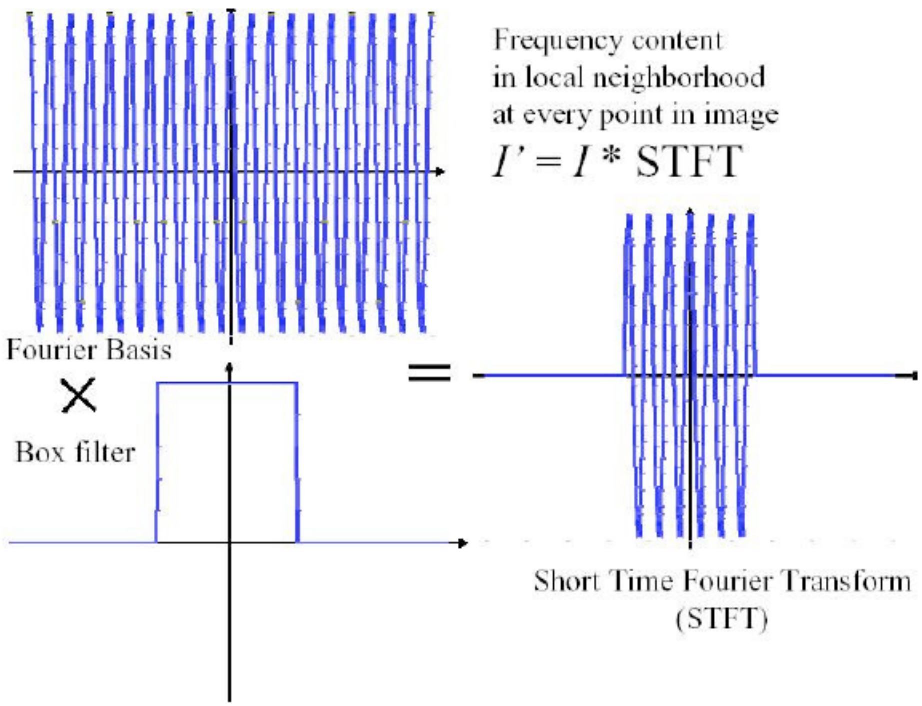
Low frequency content in all directions

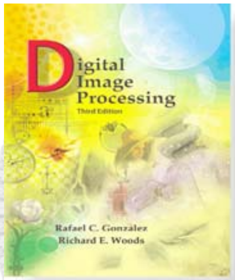
High frequency at 45° direction

Fourier transform gives information about entire image for all frequencies (scales) and orientations. This may not be useful for local textures.

Fourier Measures

- Solution is to use Short Term Fourier transforms



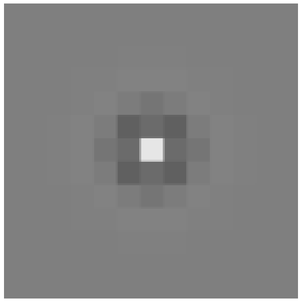


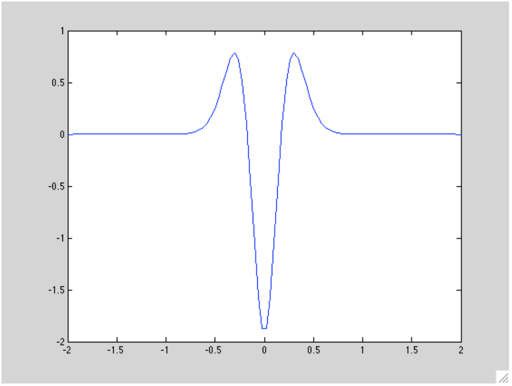
Collections of Filters

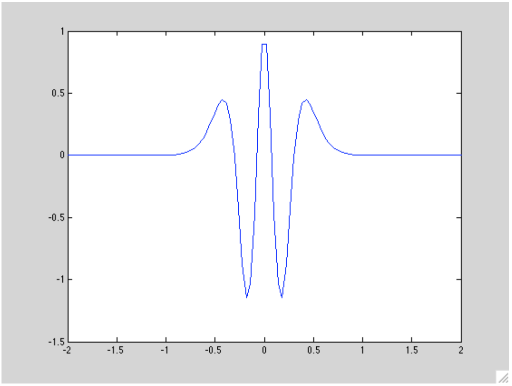
- Use a collection of filters to collect texture info
- The collection should be based upon scale (size) and direction
- A basic texture detection filter collection is spots and bars
  - Spot filters are non-directional and can recognize differences from the surrounding pixels
  - Bar filters respond to oriented structure
  - Spot and bar filters can be constructed as weighted sums of Gaussians
- Gabor filters describe frequency and orientation

Spot filters

- Two concentric, symmetric Gaussians, with weights 1 and -1, and corresponding sigmas 0.71 and 1.14
- Three concentric, symmetric Gaussians, with weights 1, -2 and 1, and corresponding sigmas 0.62, 1 and 1.6







Adapted from Forsyth & Ponce.

Oriented Bar Filters

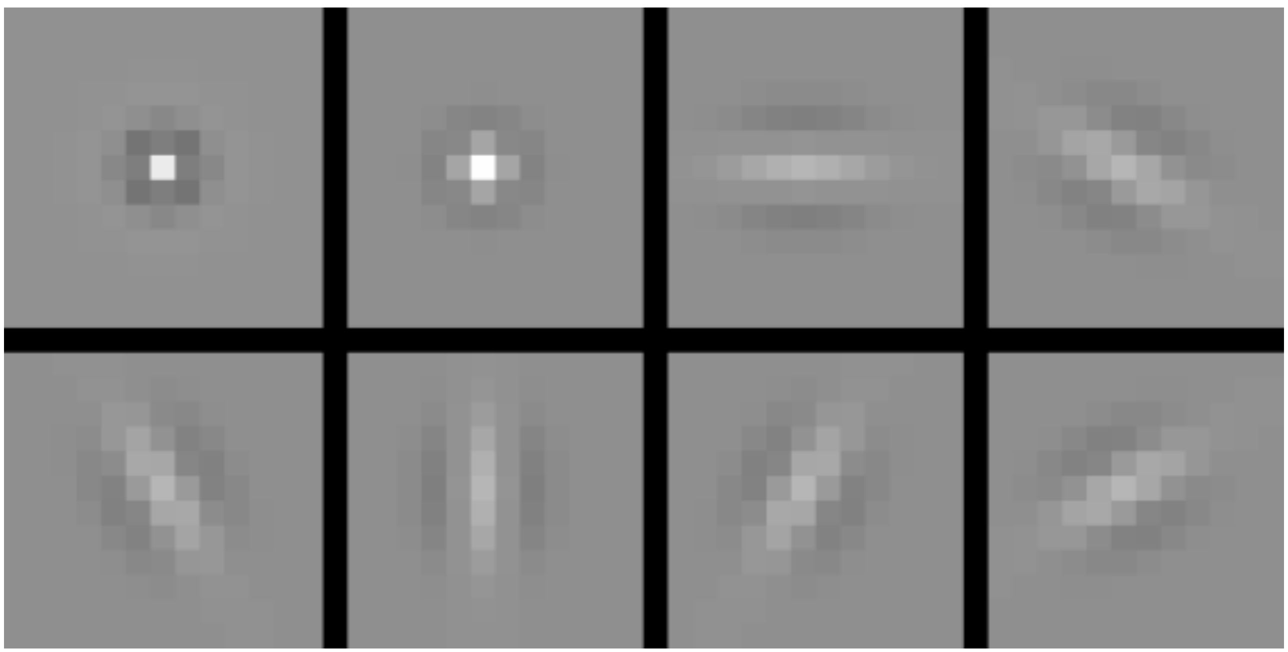
Oriented bar filters: basic horizontal bar filter with different <u>orientation</u> and, more generally, scale and phase

- Horizontal bar is three Gaussians with weights -1, 2 and -1; σ_x=2, σ_y=1; corresponding centers are offset to (0,1), (0,0) and (0,-1)
- Rotated by 45°

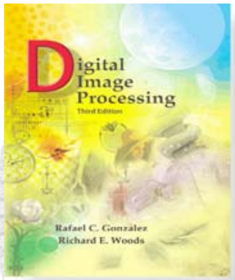
Adapted from Forsyth & Ponce.

Spot and Bar Filters

Collection of Spot and Oriented Bar Filters (from Malik and Perona)

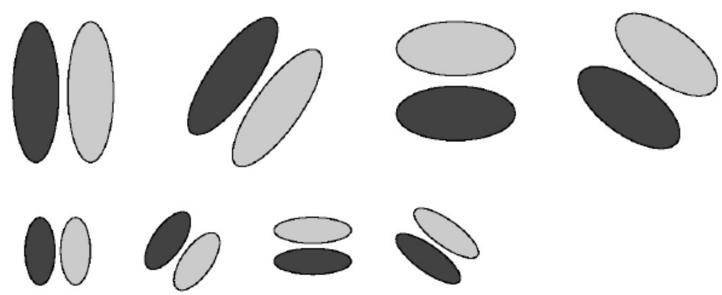


Adapted from Forsyth & Ponce.



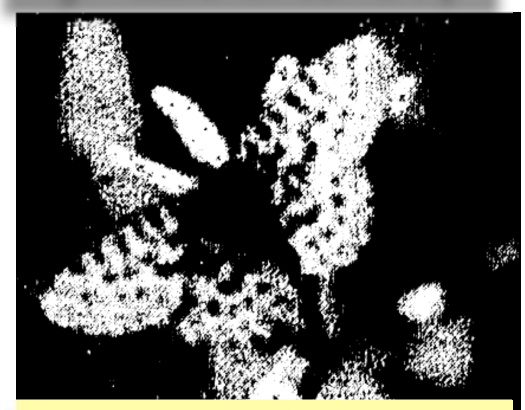
How many filters?

- Number of scales typically ranges from two to eighteen
- Number of orientations does not seem to matter in practice as long as there are about six orientations
- Basic spot filter is made from Gaussians and bar filters are made by differentiating oriented Gaussians

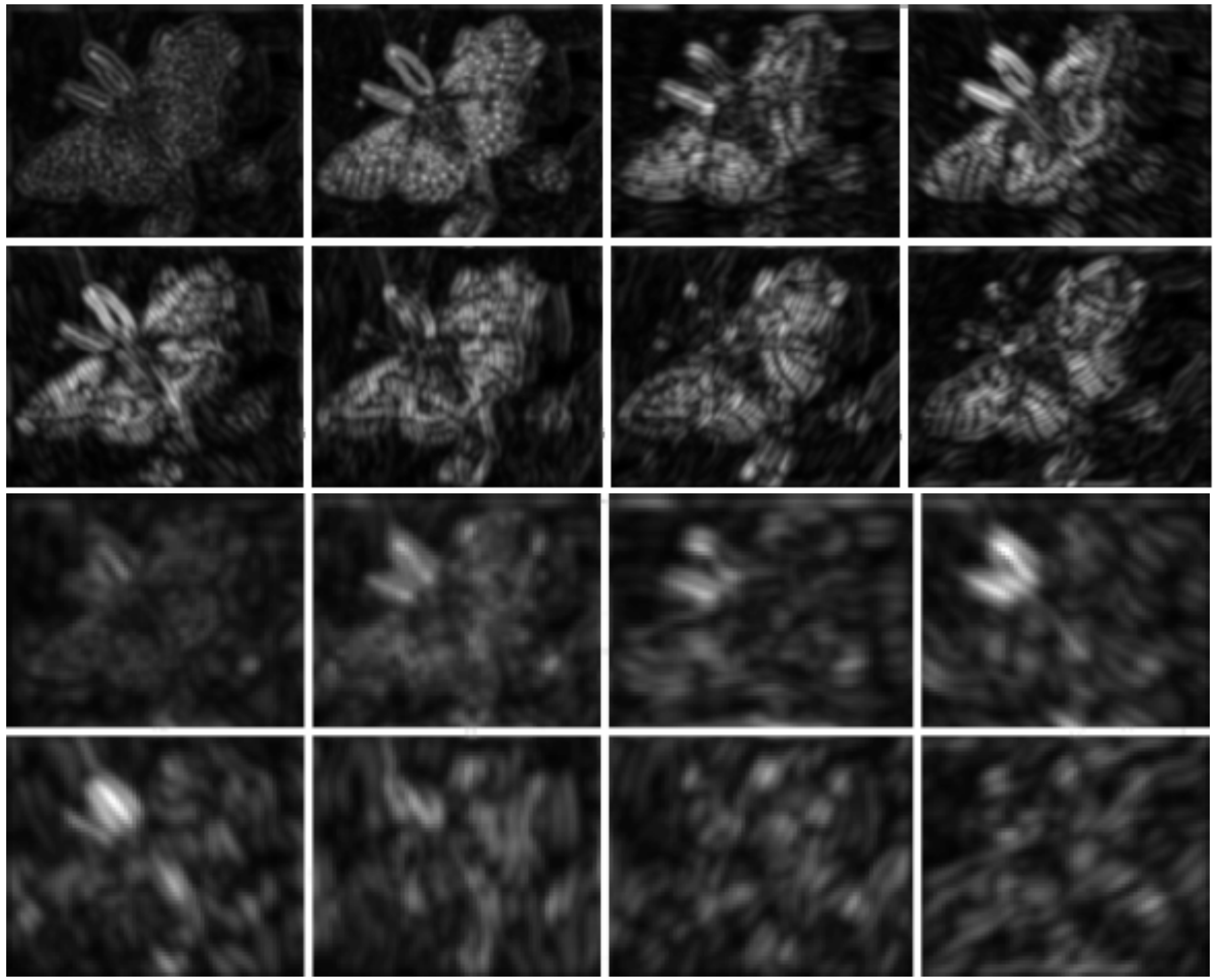


Response to Spot and Bar Filters

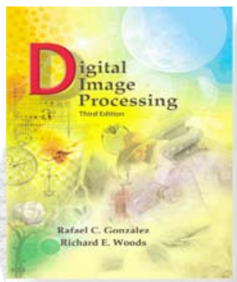
High resolution butterfly



Lower resolution (coarser scale) butterfly



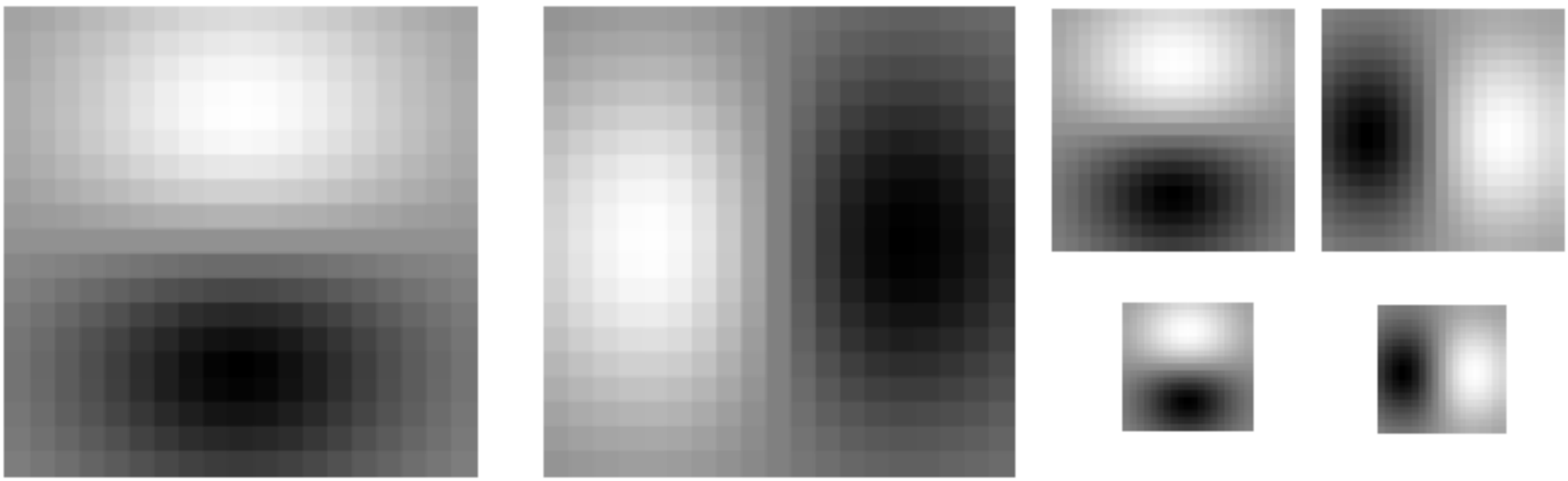
Absolute value of spot and bar filter collection response



Texture Measures

- A simple texture measure is to compute the sum of the squared filter outputs over some moving window. This is equivalent to a smoothed mean.
- If the filters have horizontal and vertical orientations then the results can be thresholded and made into a binary texture vector $(H_{texture}, V_{texture})$
- Another approach is to compute the mean and standard deviation of each filter output for a moving window and put them into a more complex texture vector.

Difference of Gaussian Filters

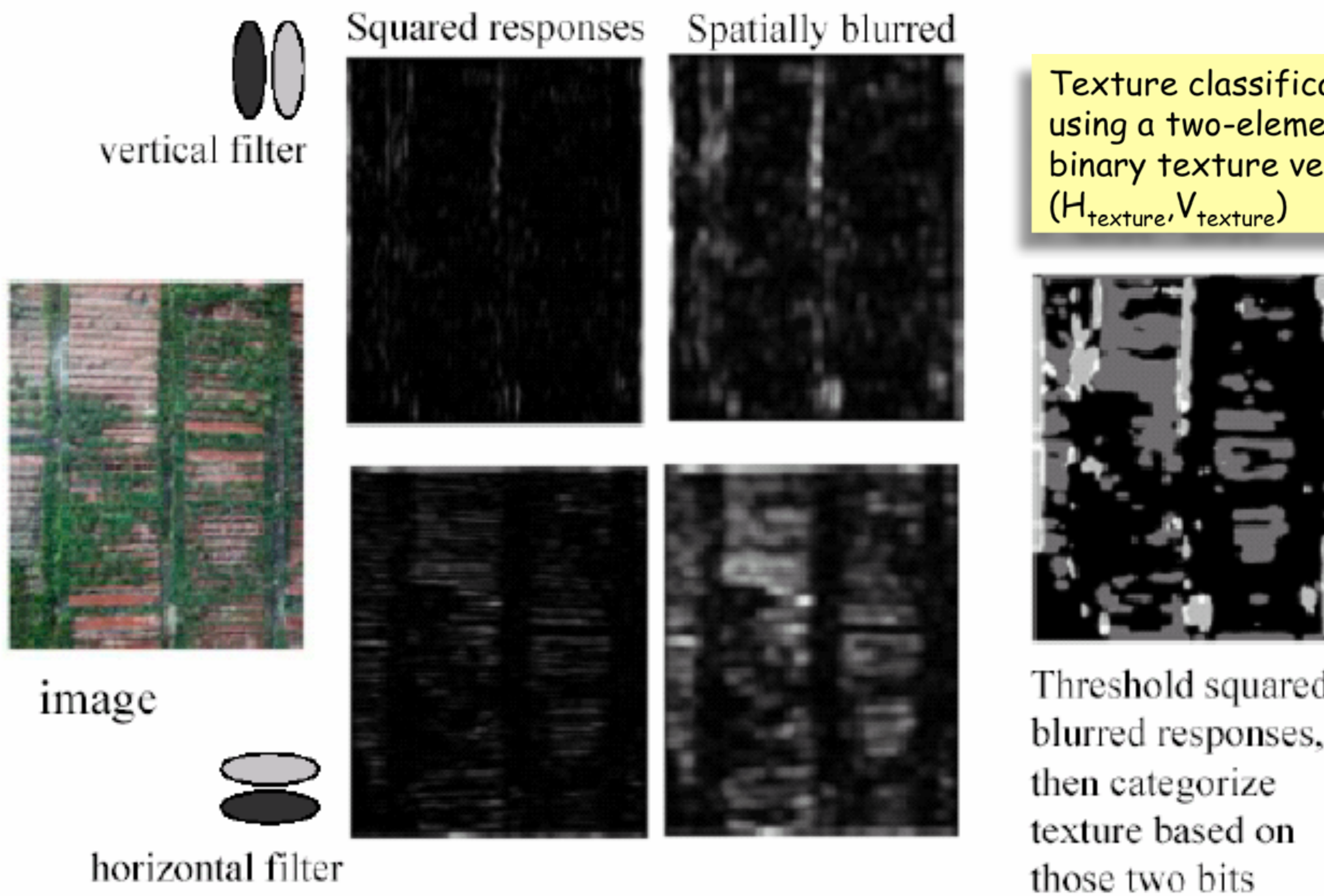


Difference of Gaussian filters are often used in texture analysis to construct spot and rod filters

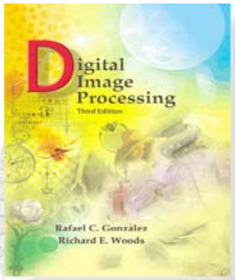
Texture Classification

(using sum of squared filter outputs over a window)

Texture classification using a two-element binary texture vector $(H_{texture}, V_{texture})$

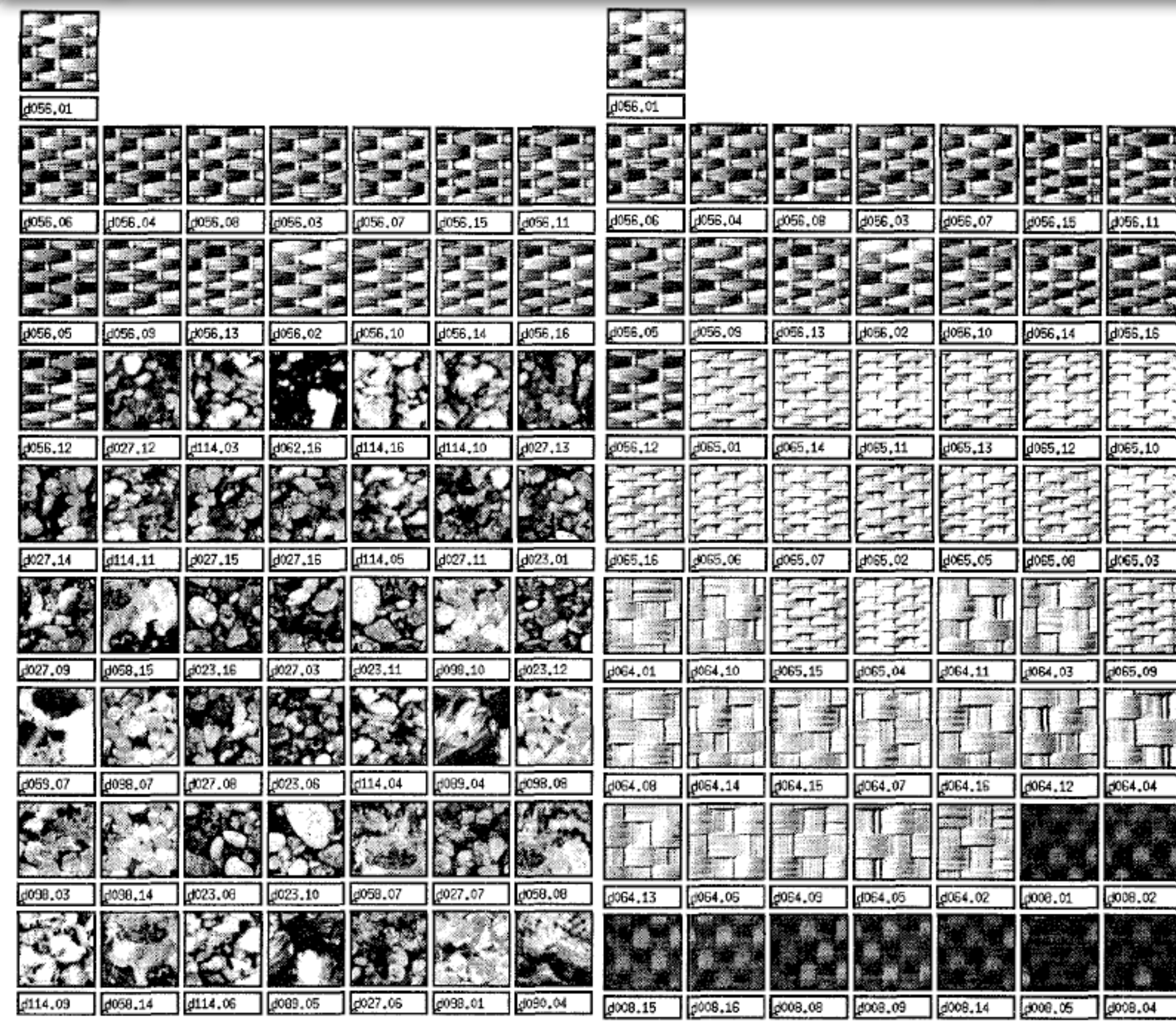


Threshold squared, blurred responses, then categorize texture based on those two bits



Texture Classification

(using mean and standard deviation of each filter output over a window)



Similar textures using Euclidian distance measure

Similar textures using modified Euclidian distance measure

Similarity decreases left->right, top->bottom

Picking Scale

- One can start with a small scale (size of the window) and increase the scale until there is no significant change in the texture classification
- Another method of determining scale is polarity
  - Compute the average direction (the dominant orientation) of the gradient for a window. Compute the dot product between the gradient at each point and the average gradient orientation.
  - Compute the averages for the positive and then the negative dot products.
  - Start at a small scale and increase the scale until the polarity does not change.

Gaussian Pyramid

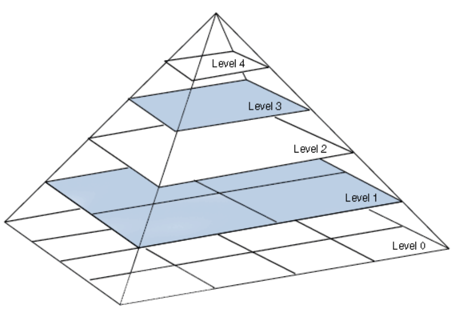
- $S^{\downarrow}$ downsamples an image I, i.e., the j,k-th element of $S^{\downarrow}(I)$ is the 2j,2k-th element of I.
- The n-th level of a pyramid P(I) is denoted $P(I)_n$

- Define a Gaussian pyramid as

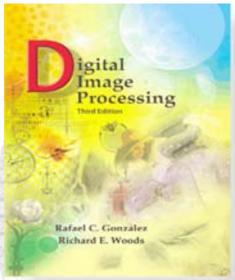
$$P_{Gaussian}(I)_{n+1}=S^{\downarrow}(G_\sigma \star P_{Gaussian}(I)_n = S^{\downarrow} G_\sigma (P_{Gaussian}(I)_n)$$

- The finest scale is the starting laye

$$P_{Gaussian}(I)_1=I$$

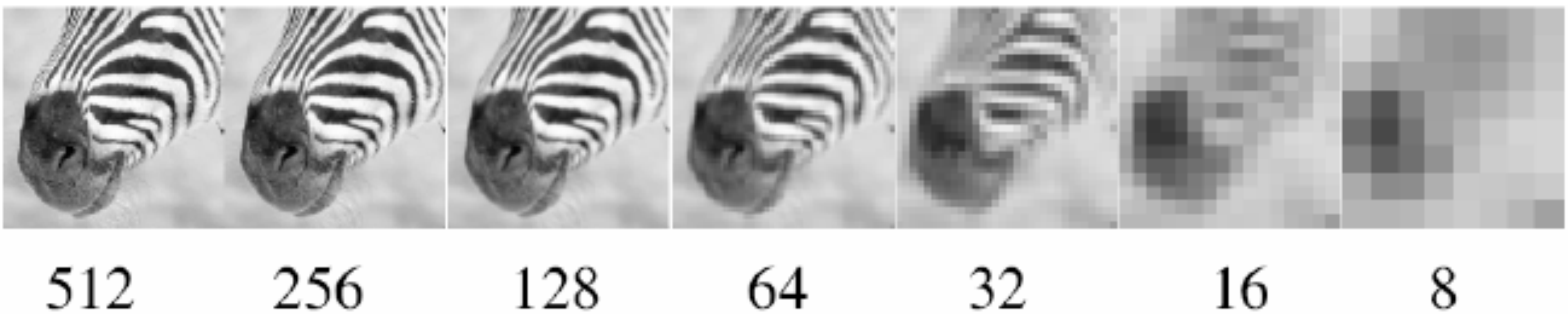


Smooth each layer with a Gaussian and downsample to get the next layer.



Gaussian Pyramid

Smooth each layer with a Gaussian and downsample to get the next layer.



Each image shown at the same size with pixels of differing sizes

Gaussian Pyramid

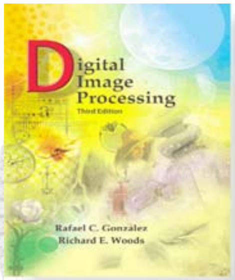
Pixels shown actual size

Laplacian Pyramid

- A Gaussian pyramid duplicates spatial frequency information between layers
- Define $S^{\uparrow}(I)$ which upsamples an image from level n+1 to level n, i.e., produces elements at (2j-1,2k-1),(2j,2k-1),(2j-1,2k),and (2j,2k) all with the same value from the (j,k) element of I
- A Laplacian pyramid is a difference of Gaussians

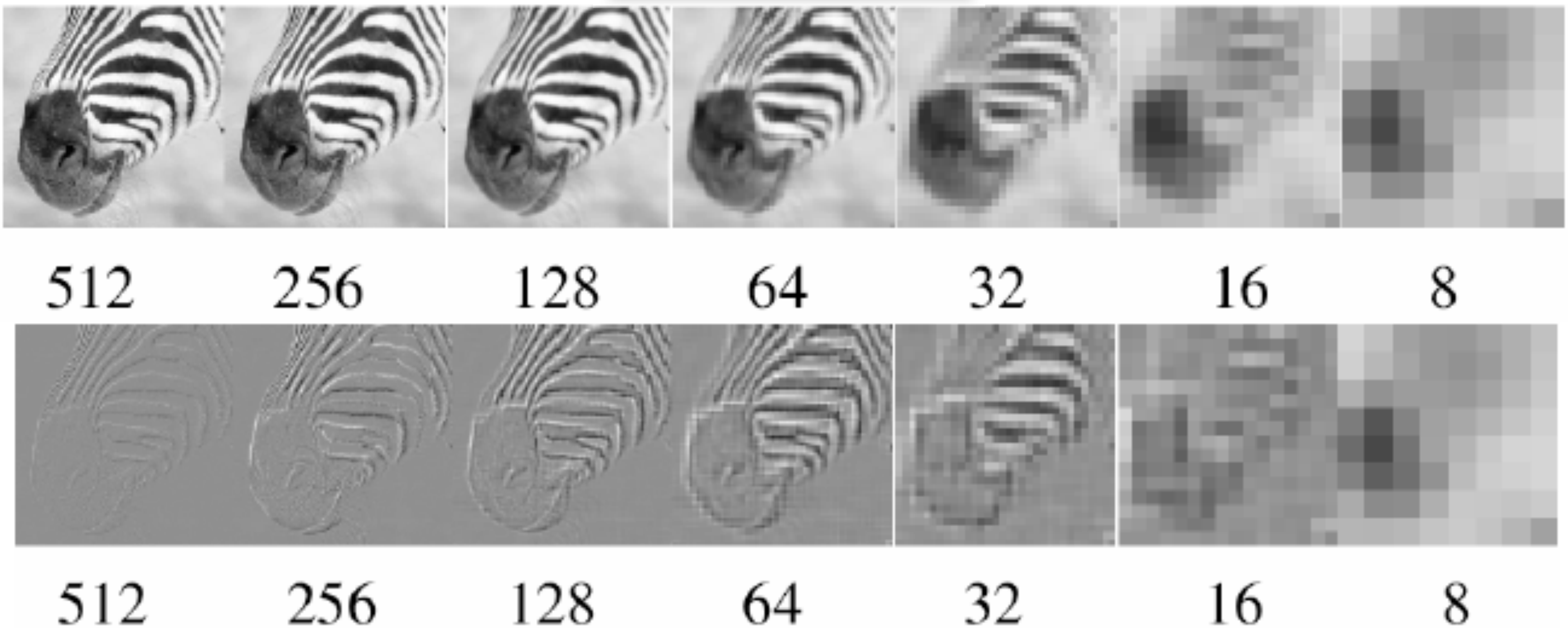
  $P_{Laplacian}(I)_k = P_{Gaussian}(I)_k - S^{\uparrow}\,(P_{Gaussian}(I)_{k+1})$

  Since both are filtered each level of the Laplacian pyramid represents a range of spatial frequencies



Laplacian Pyramid

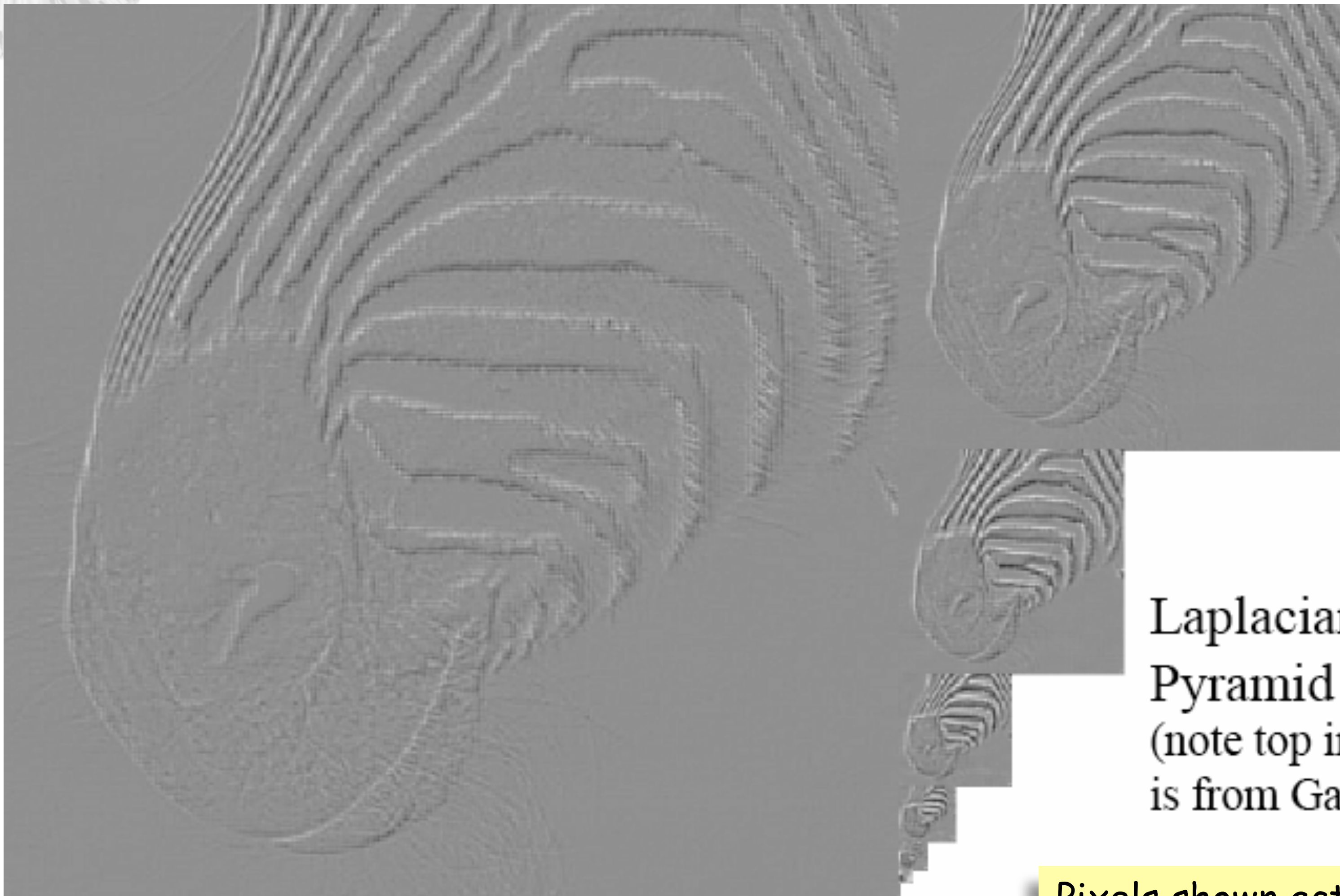
Gaussian pyramid



Laplacian pyramid

Strong responses at particular scales occur because each layer corresponds to a band-pass filter. When the scale of the image matches the frequency response of the Laplacian level there is a strong response

Laplacian Pyramid

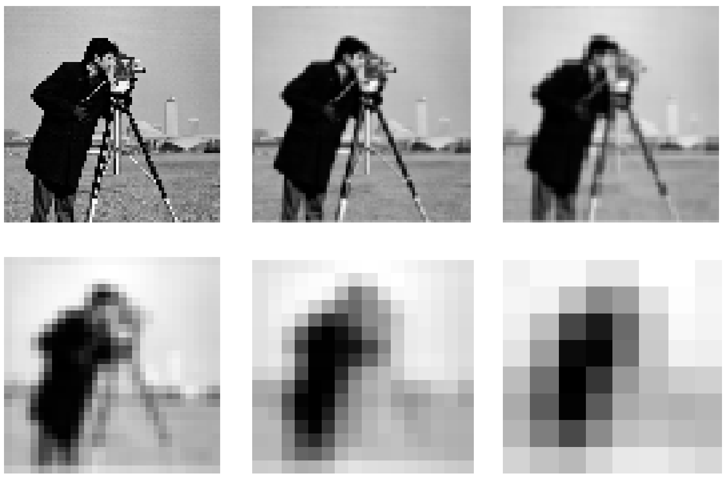


Laplacian Pyramid
(note top image is from Gaussian)

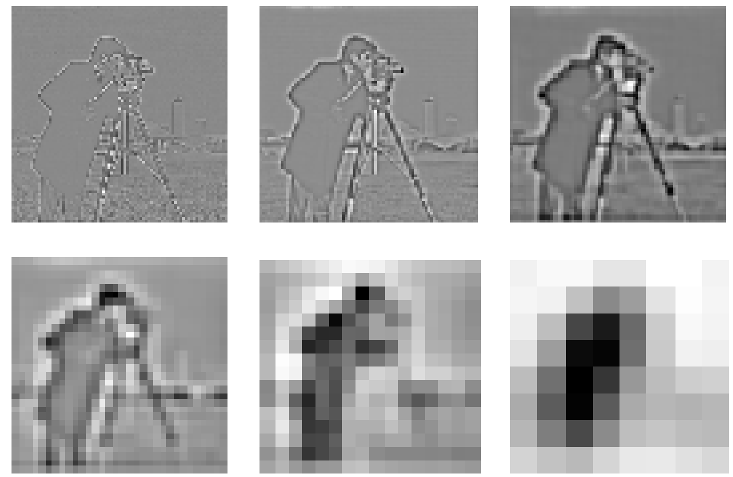
Pixels shown actual size

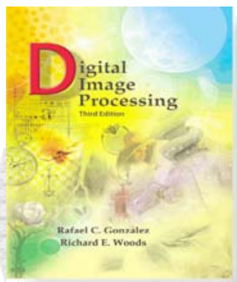
Another Example

Gaussian Pyramid



Laplacian Pyramid





Laplacian Pyramid

An image can be recovered from its Laplacian pyramid by the following algorithm:

Working layer is the coarsest layer

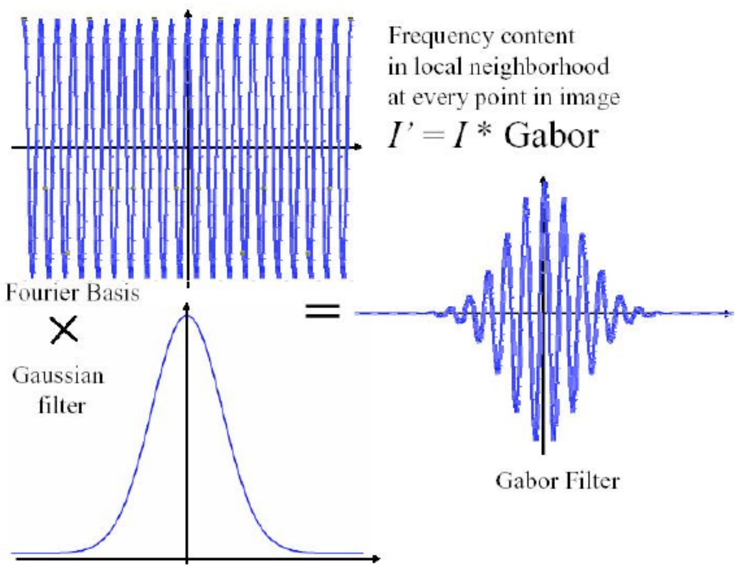
For each layer going from next coarsest to fine

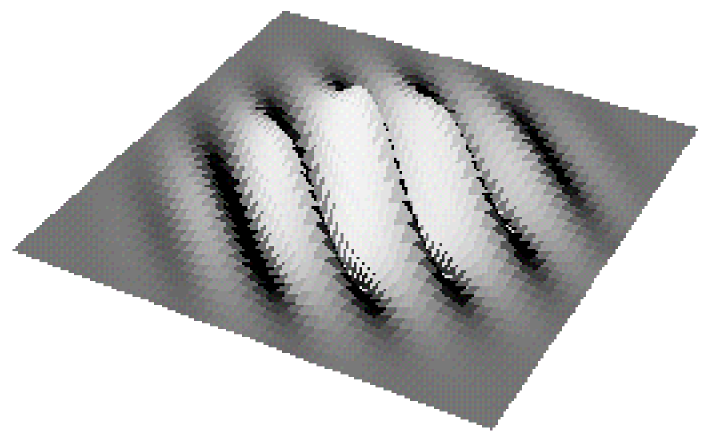
- Upsample the working layer and add the current layer to the result.
- Set the working level to be the result of this operation

end

Gabor Filters

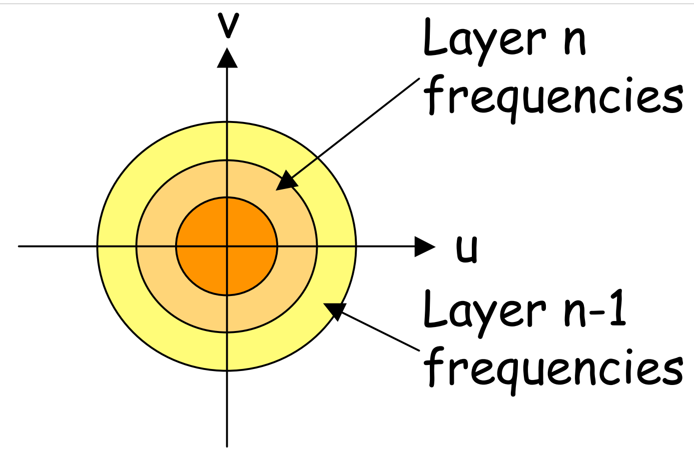
Gabor filters are logical extensions of the short term Fourier transform





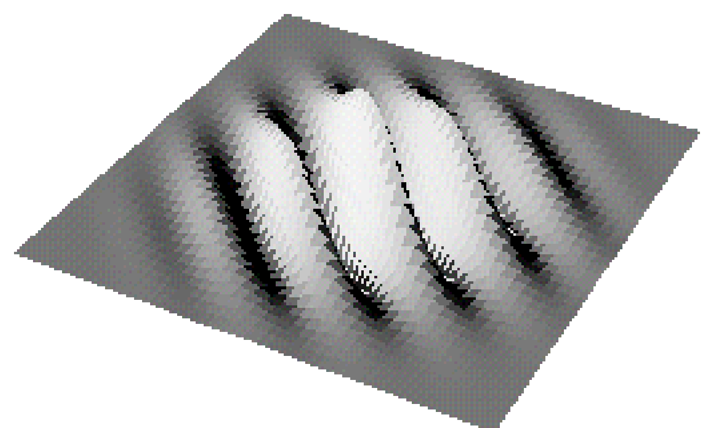
Gabor Filters

Each layer of the Laplacian pyramid can be thought of as a range of spatial frequencies, i.e., an annulus in the Fourier u-v space

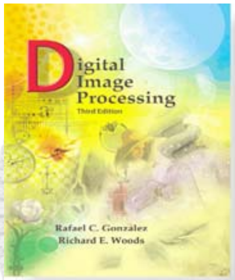


A Fourier transform is a transform of the entire image. A Gabor filter is a product of a Gaussian and a Fourier basis function which can perform oriented, local frequency analysis

A Gabor filter is also very similar to the response of receptive fields in the visual cortex.



Journal of Neurophysiology, Vol 58, Issue 6 1233-1258
An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex
J. P. Jones and L. A. Palmer
Department of Anatomy, University of Pennsylvania School of Medicine, Philadelphia 19104-6058.



Gabor Filters

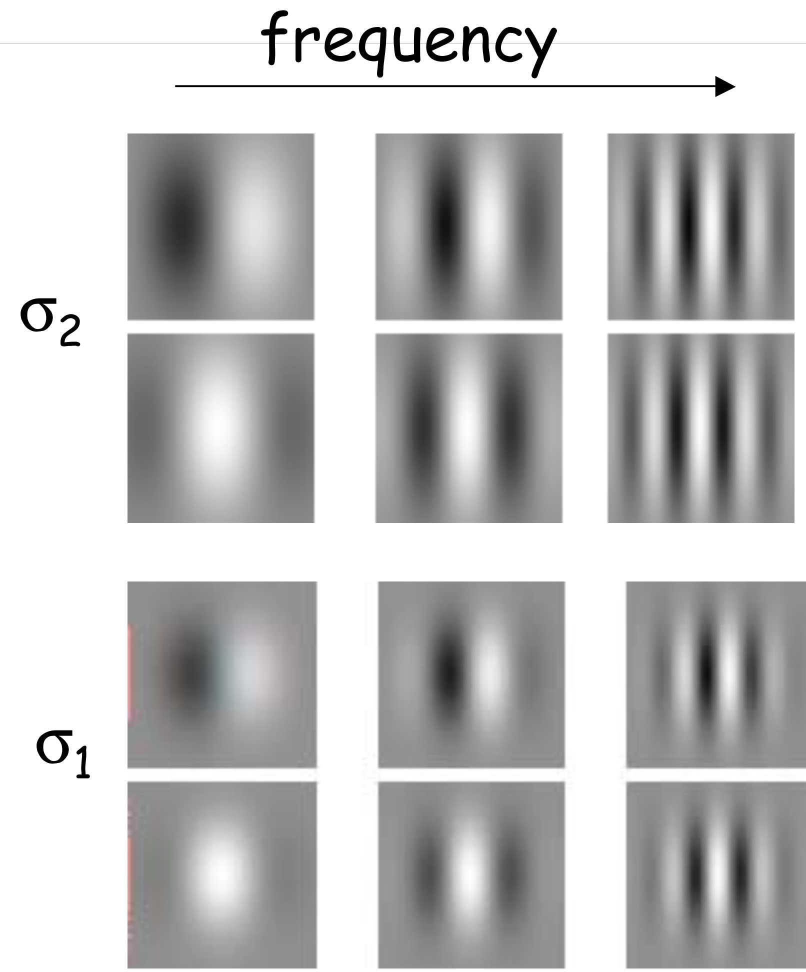
Gabor filters at two different scales and three spatial frequencies

Top row shows anti-symmetric (or odd) filters,

$$G_{antisymmetric}(x,y) = \sin\left(k_0 x + k_1 y\right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Bottom row shows symmetric (or even) filters.

$$G_{symmetric}(x,y) = \cos\left(k_x x + k_y y\right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$



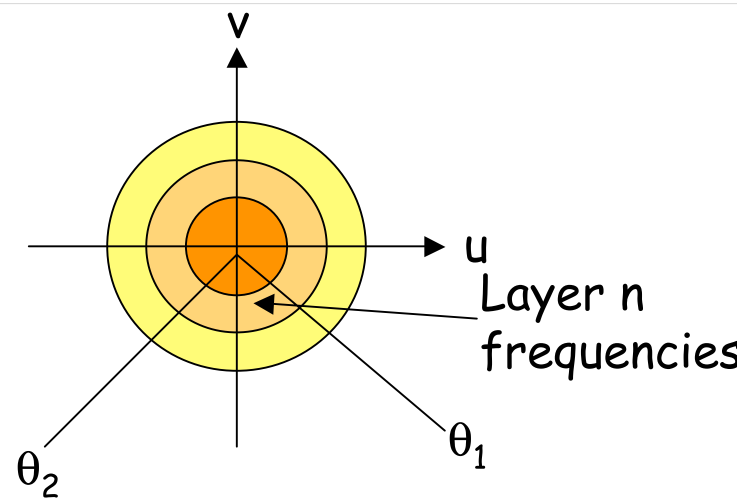
Gabor Filters

A Gabor filter gives what is known as an oriented pyramid where each filter corresponds to a angular segment of an annulus in frequency space

$$G_{antisymmetric}(x,y) = \sin(k_0 x + k_1 y)\, e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$G_{symmetric}(x,y) = \cos(k_x x + k_y y)\, e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Gabor filters are anti-symmetric since in the limit as σ->∞ the Gabor functions become the Fourier basis functions



A Gabor filter might pass frequencies in layer n and between θ_1 and θ_2

Gabor filters are examples of wavelets

Pyramids

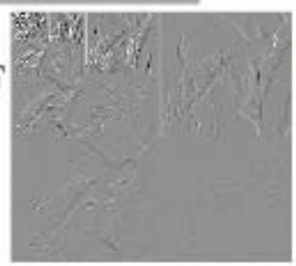
- Gaussian

Progressively blurred and subsampled versions of the image. Adds scale invariance to fixed-size algorithms.

- Laplacian

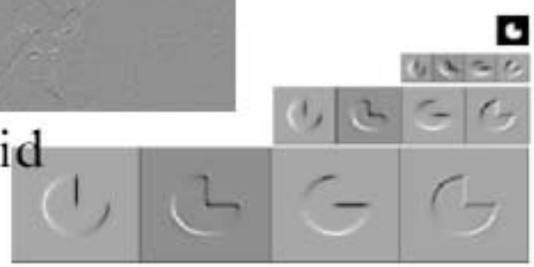
Shows the information added in Gaussian pyramid at each spatial scale. Useful for noise reduction & coding.

- Wavelet/QMF



Bandpassed representation, complete, but with aliasing and some non-oriented subbands.

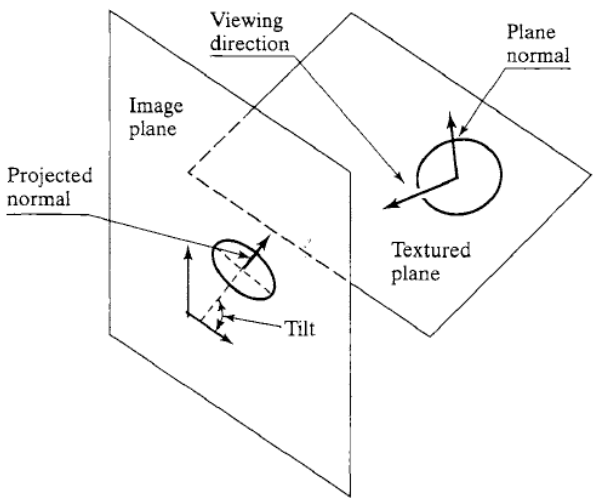
- Steerable pyramid



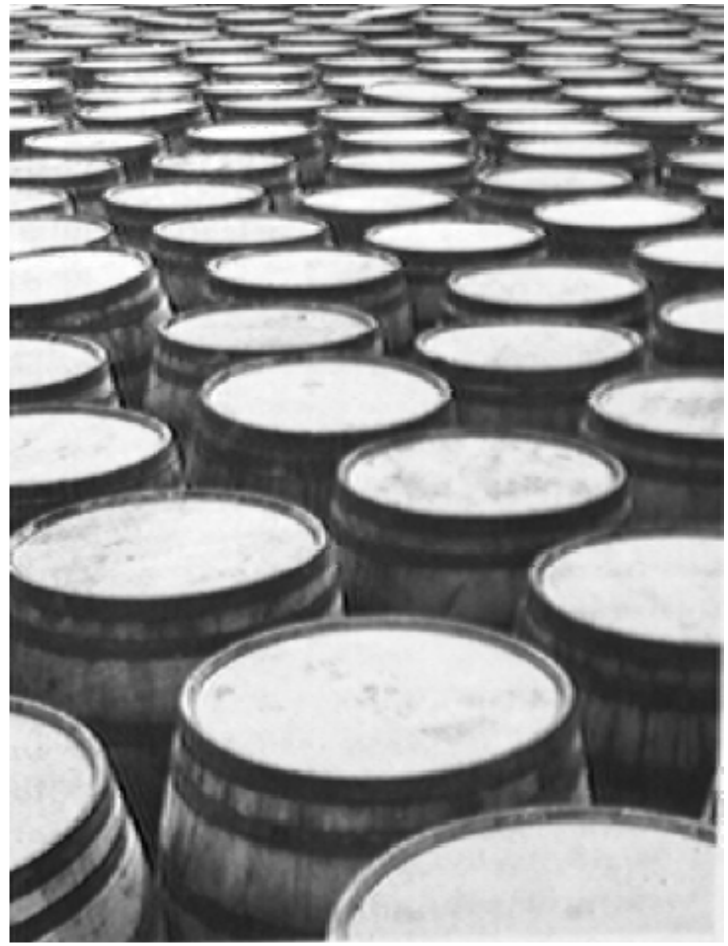
Shows components at each scale and orientation separately. Non-aliased subbands. Good for texture and feature analysis.

Texture

Gradient in the spacing of barrels (Gibson 1957)



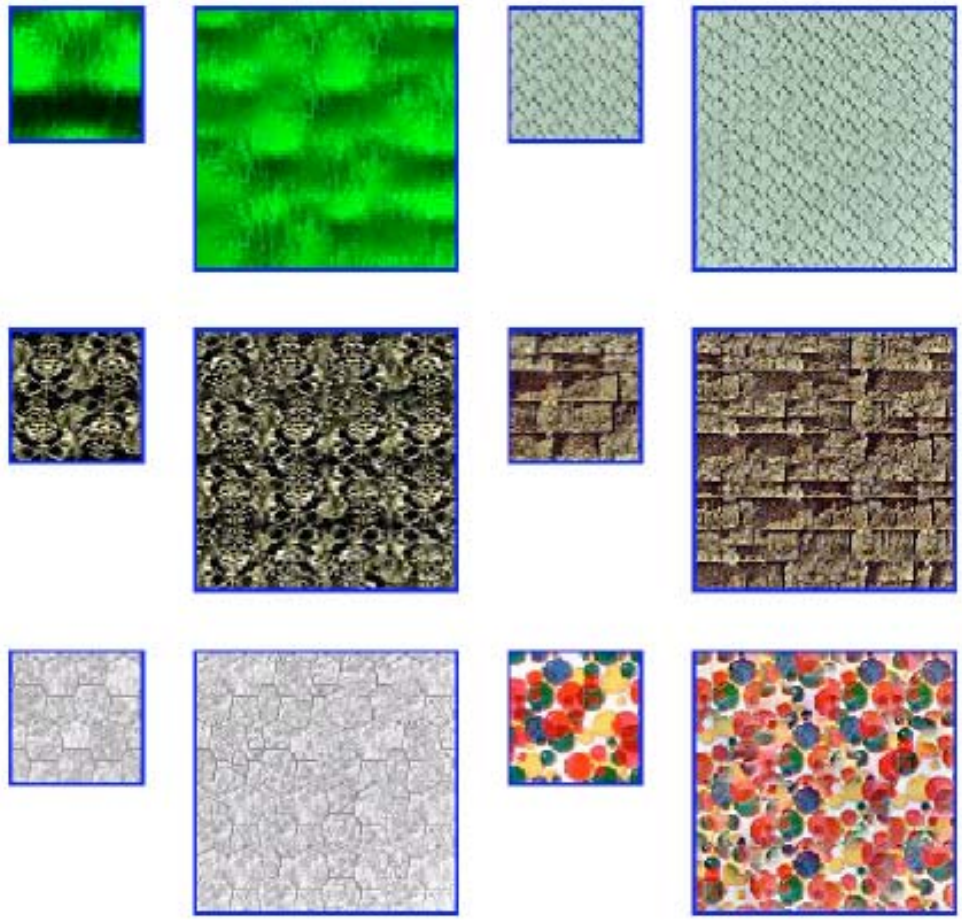
Texture is subject to perspective transformations giving rise to a perspective gradient



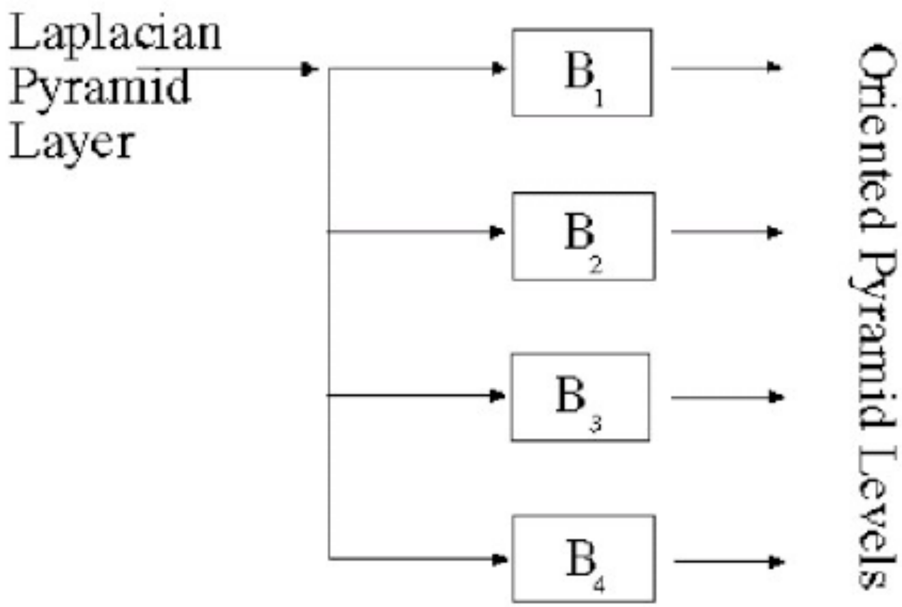
Texture

Texture gradient associated with converging lines (Gibson 1957)

Texture Synthesis

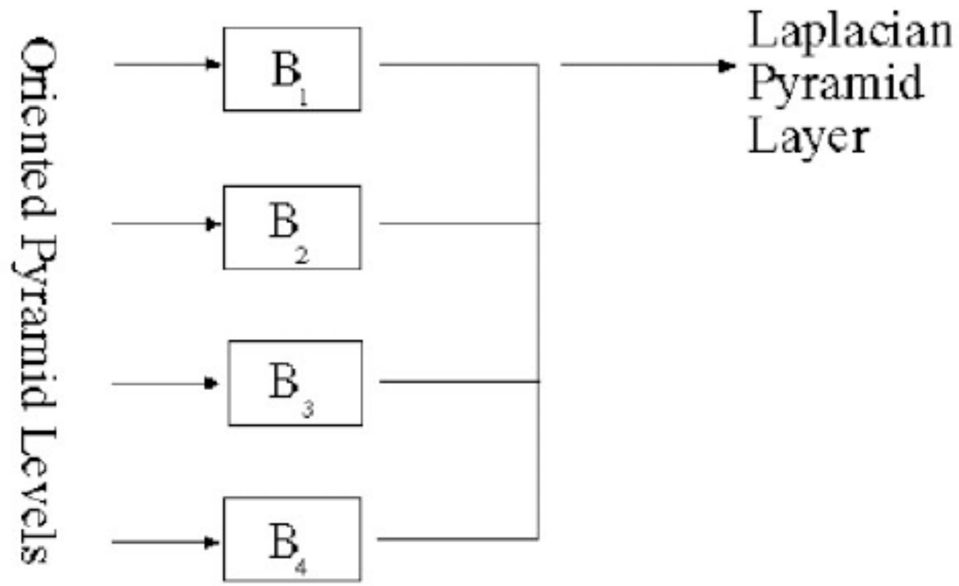


Texture Analysis



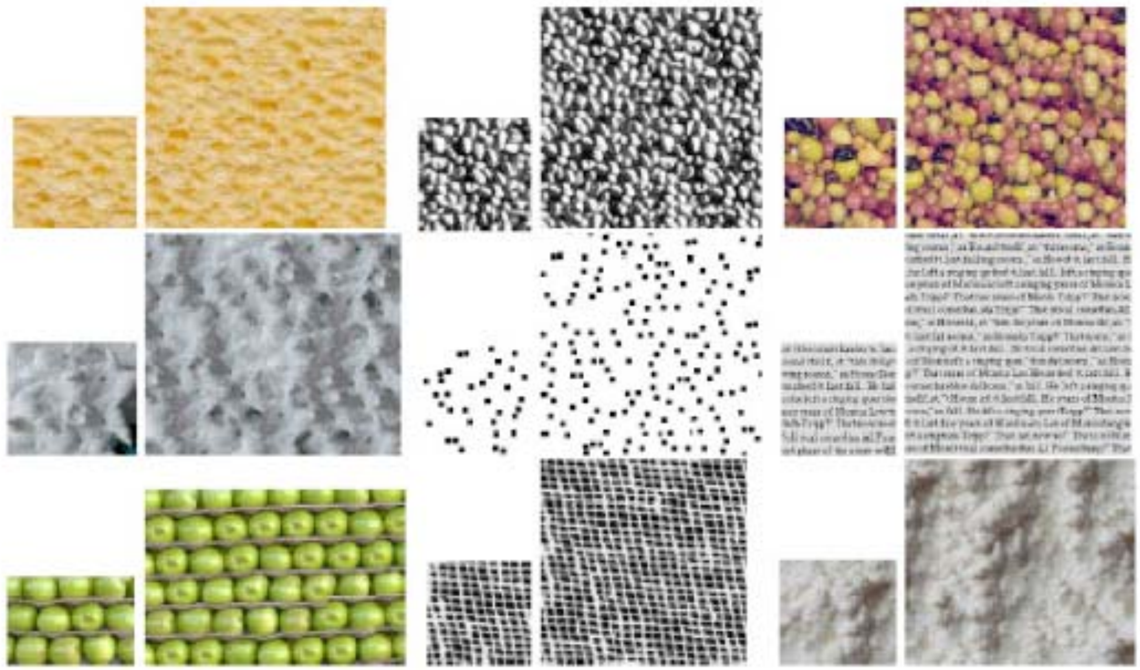
Take the layers of Laplacian Pyramid and apply oriented filters

Texture Synthesis



Re-filter the layers and add them

Texture Synthesis



From "Image quilting for texture synthesis and transfer", Efros and Freeman, SIGGRAPH 2001