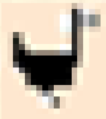
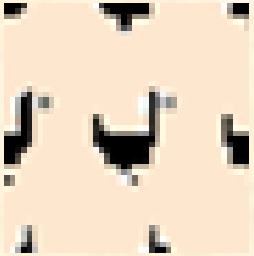
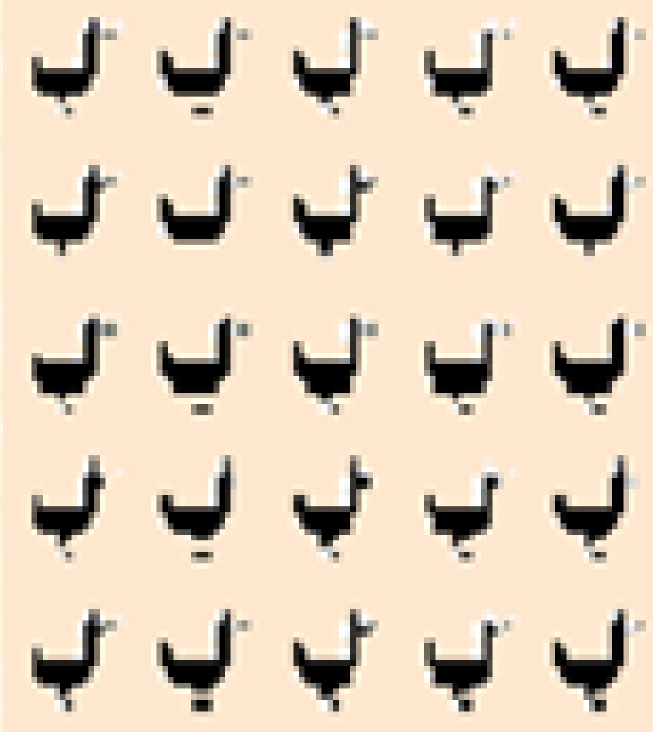
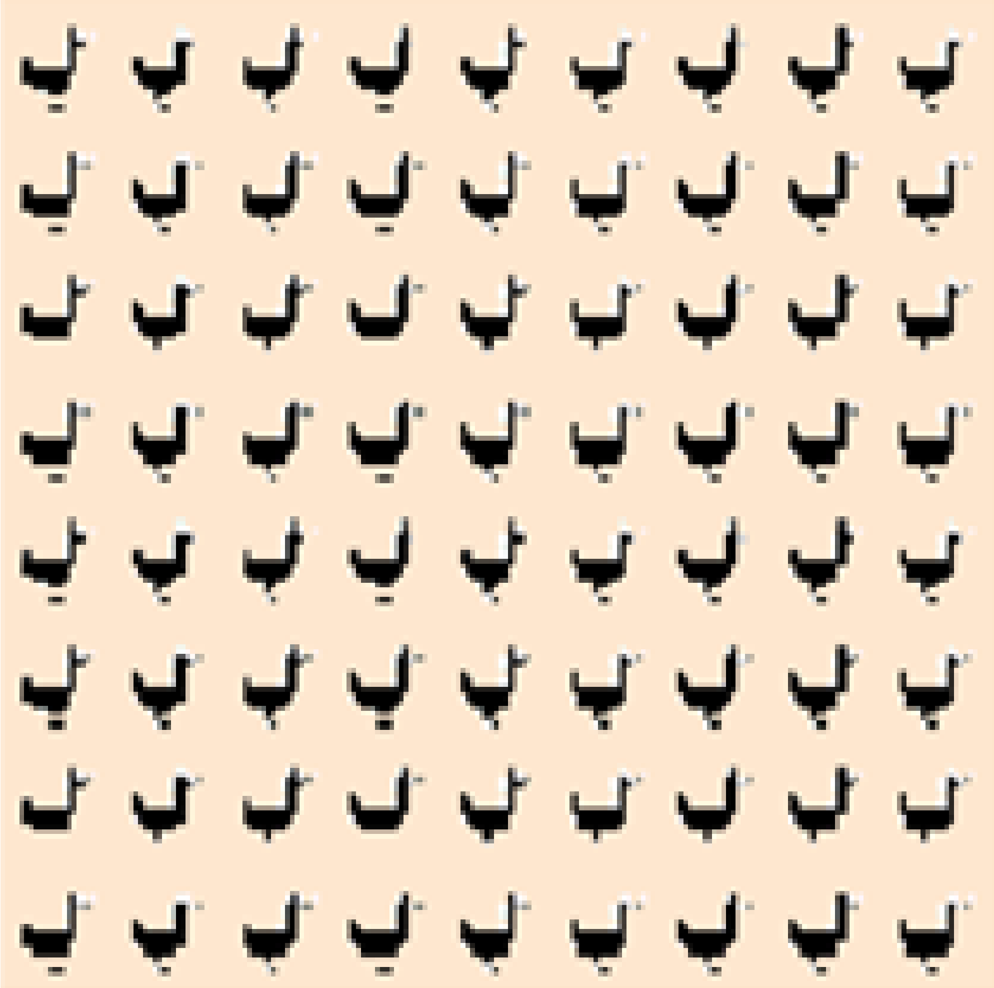
# TEXTURE ANALYSIS

# USING

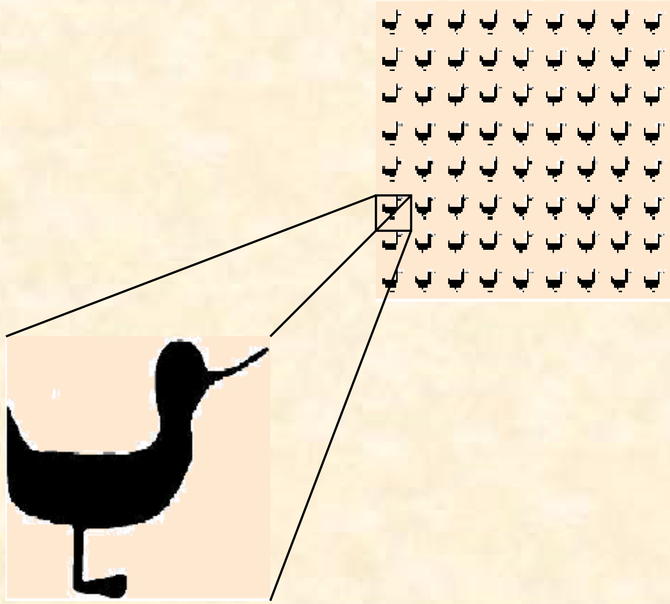
# GABOR FILTERS

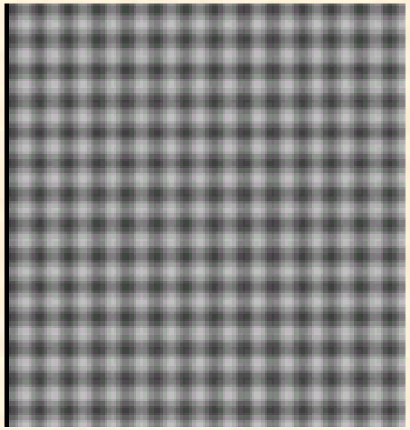



# Texture Types

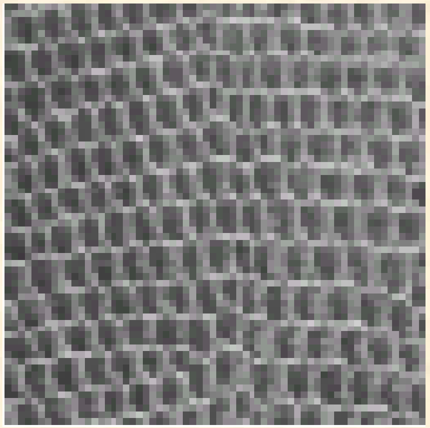
- Definition of Texture



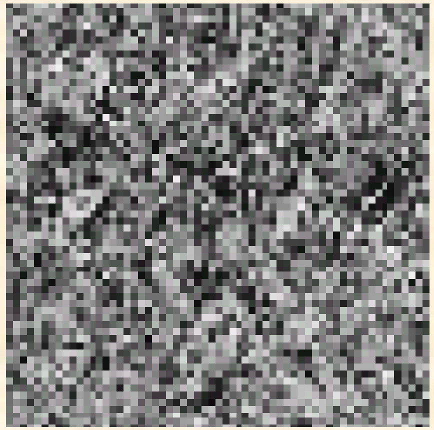
- Texture types



Synthetic



Natural



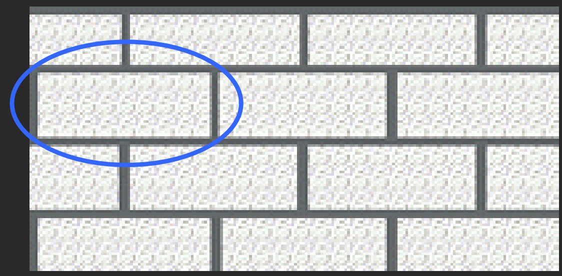
Stochastic

< Prev

Next >

# Texture Definition

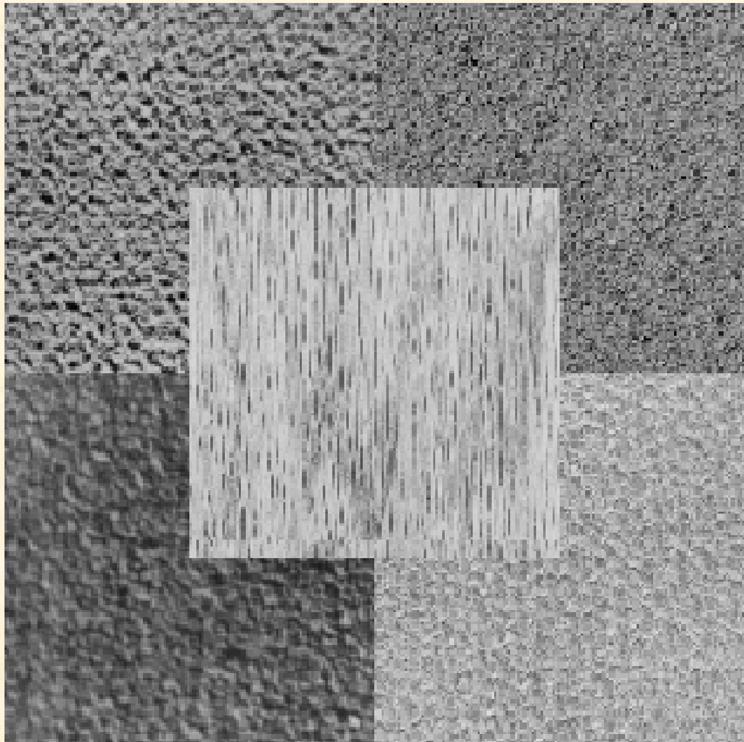
Texture: the regular repetition of an element or pattern on a surface.

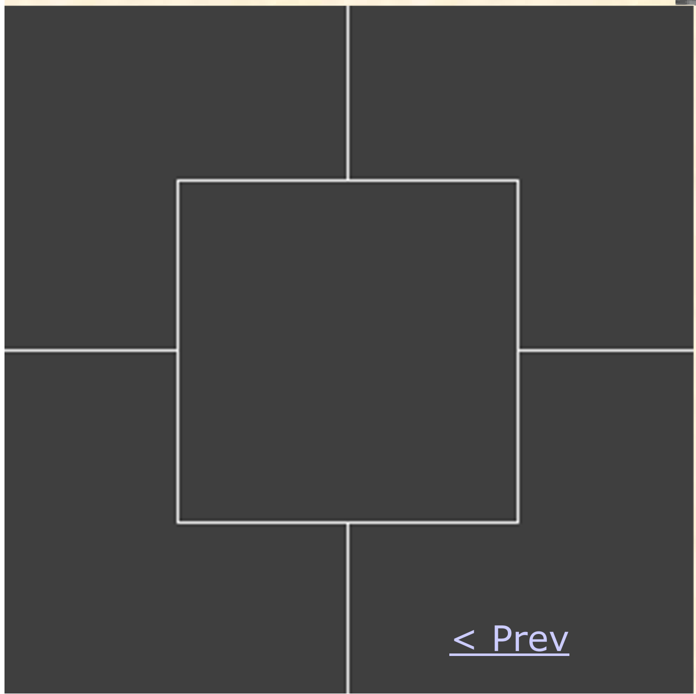


- Purpose of texture analysis:
  - **To identify different textured and non-textured regions in an image.**
  - ***To classify/segment different texture regions in an image.***
  - ***To extract boundaries between major texture regions.***
  - **To describe the texel unit.**
  - **3-D shape from texture**

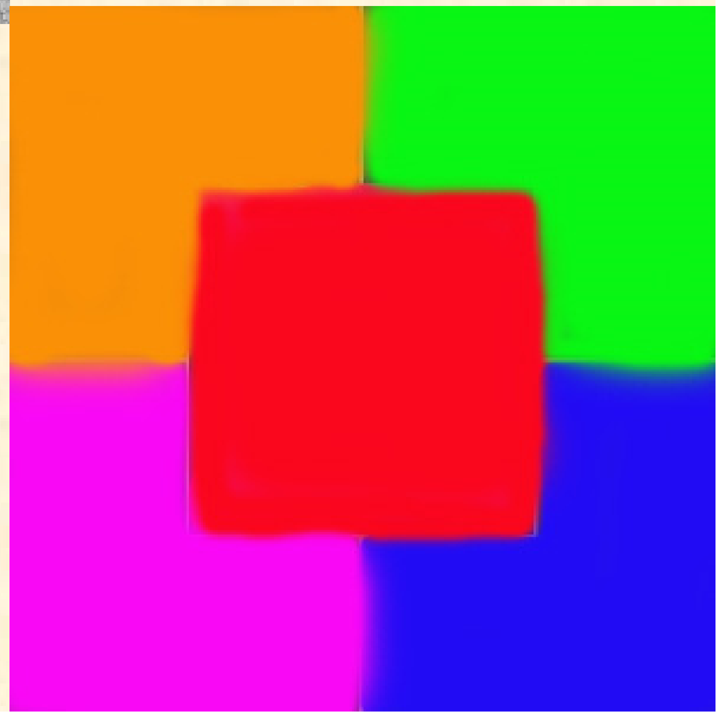
Ref: [Forsyth2003, Raghu95]

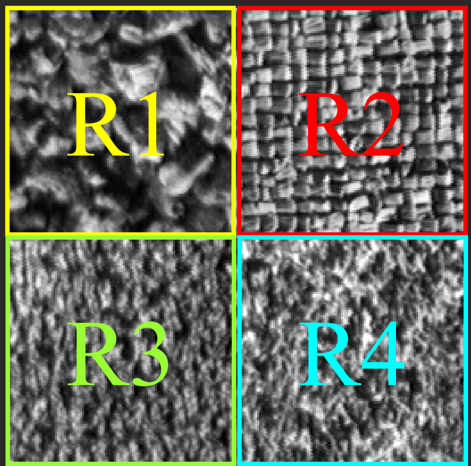
# Texture Regions & Edges





< Prev






Textured image

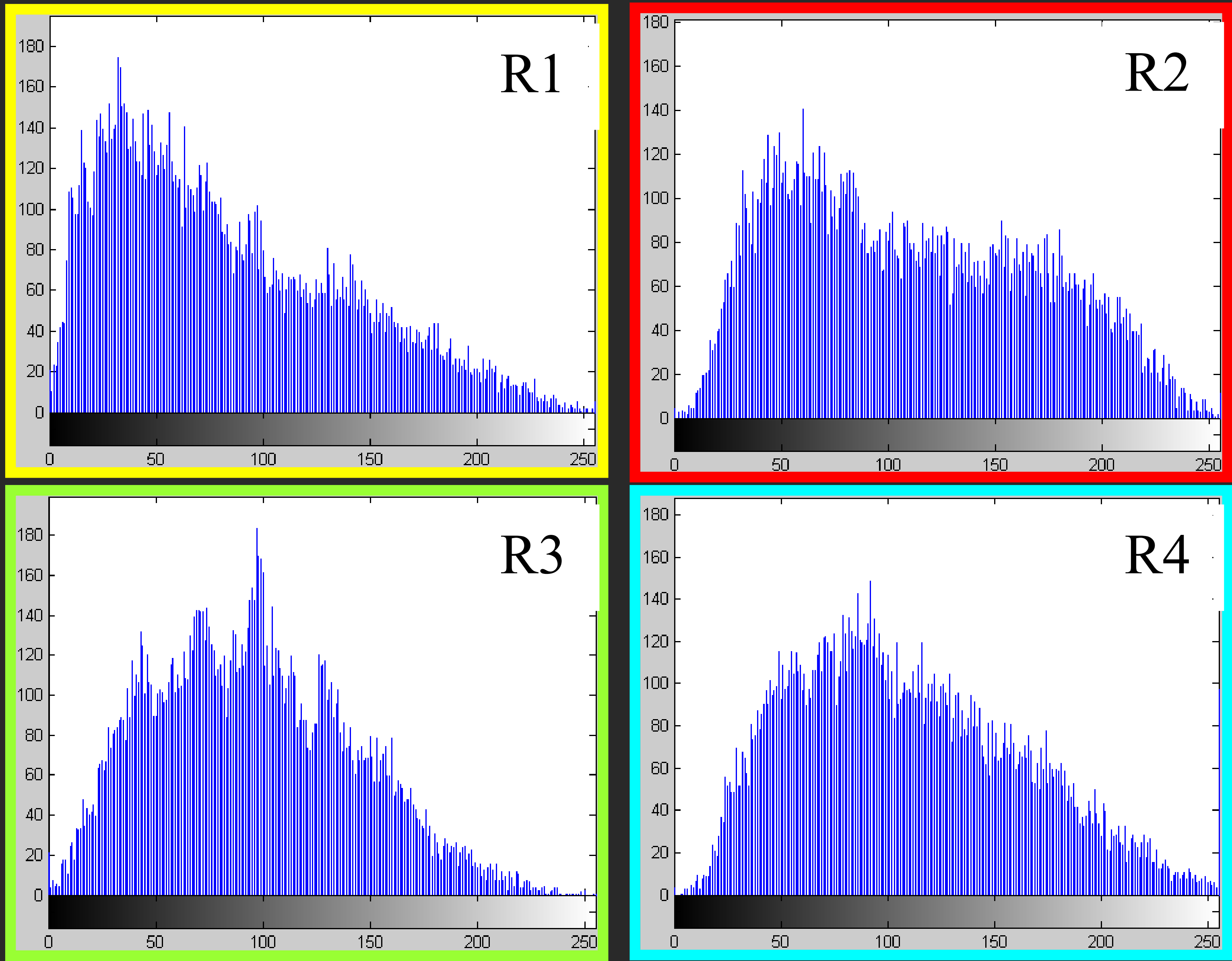
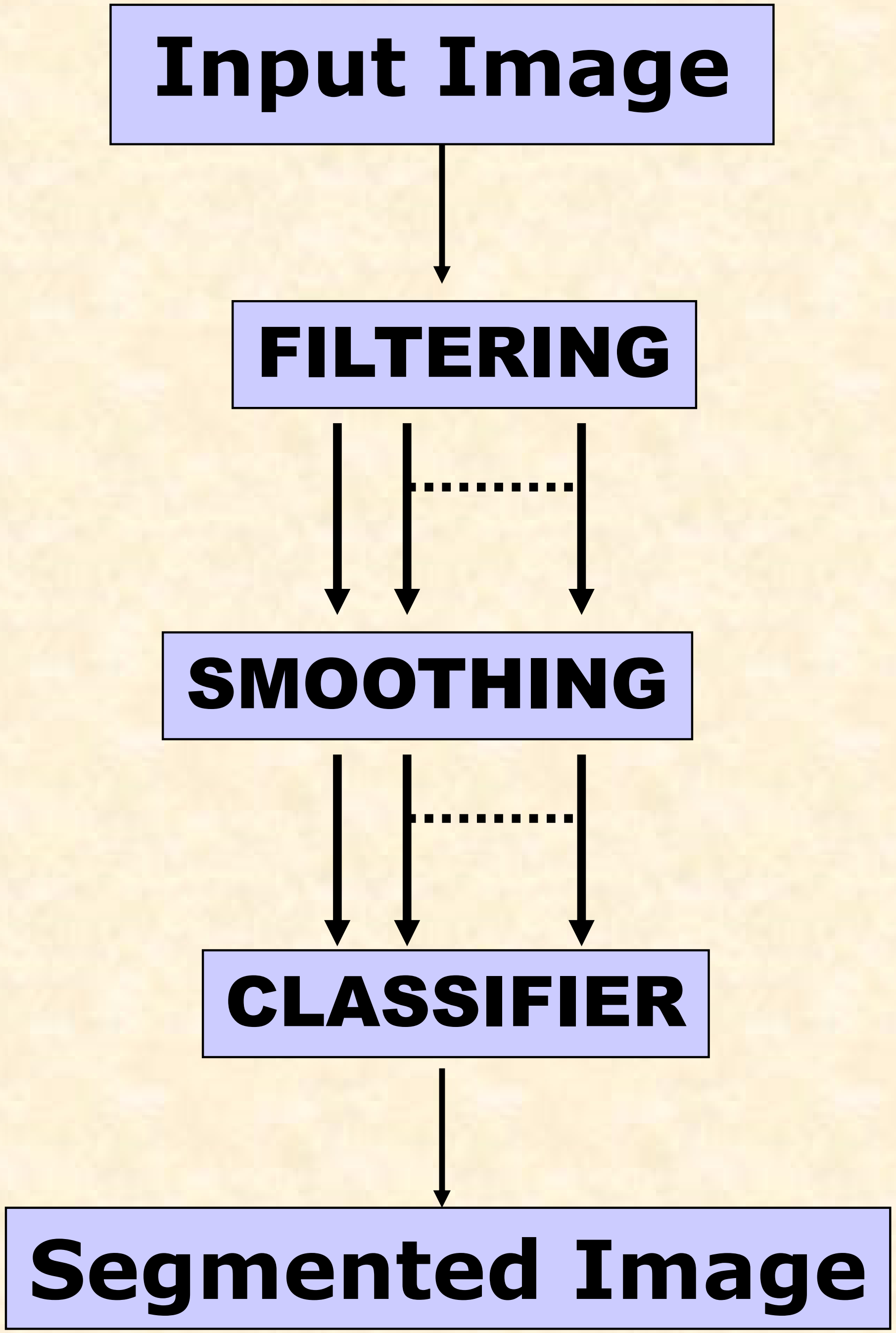



Image histograms




**Flow-chart of a typical method of texture classification**

Synthetic Texture Image

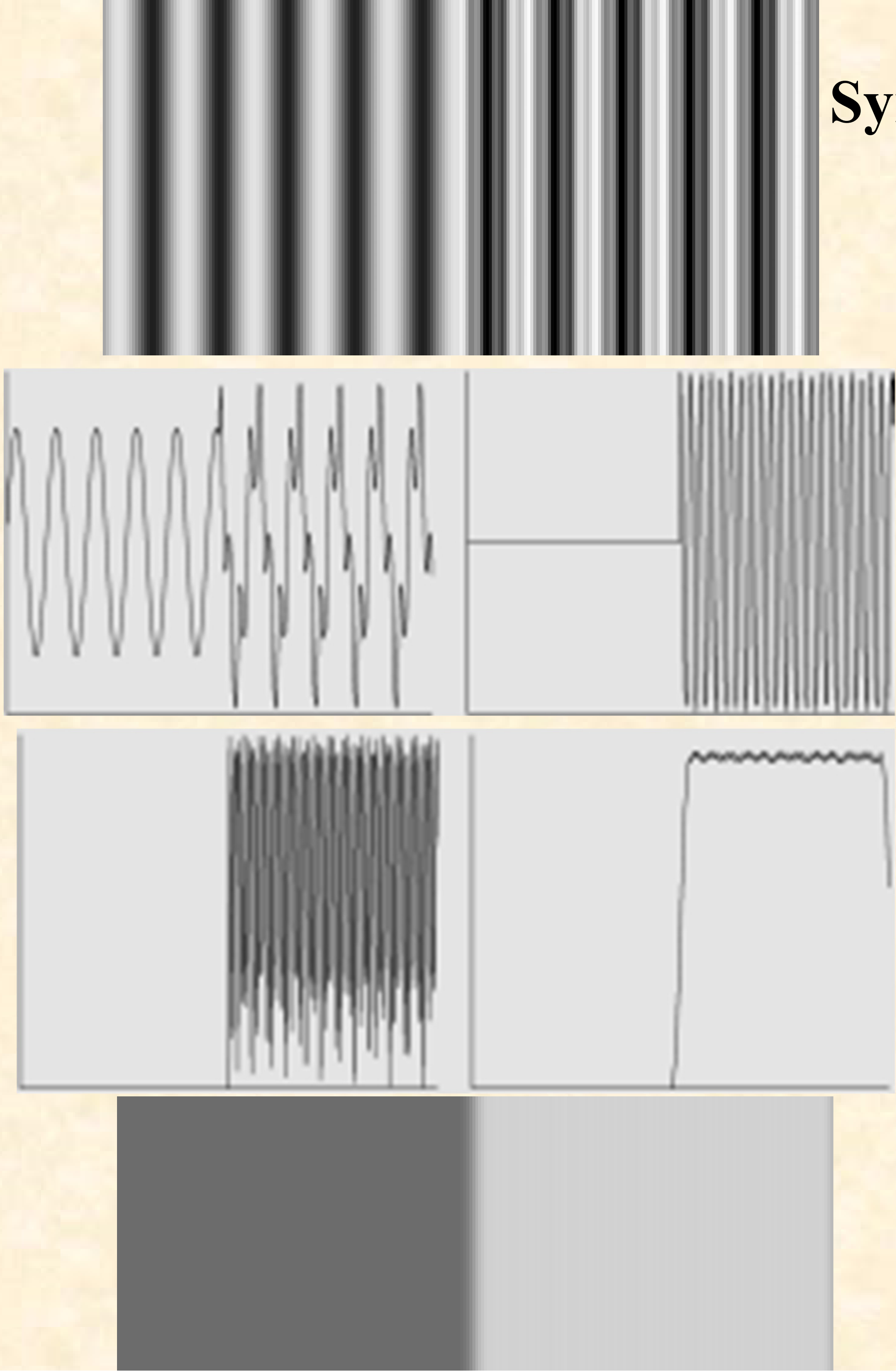
Horizontal intensity profile

Filtered output

Nonlinear transform

Smoothing

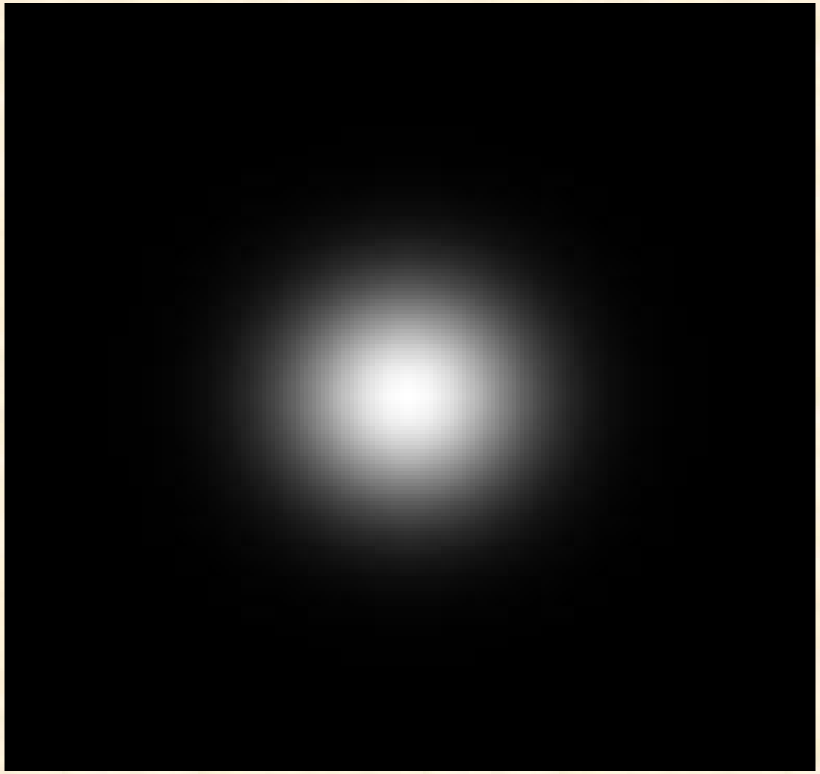
Segmented image



# Processing of Texture-like Images

## 2-D Gabor Filter

$$f(x,y,\omega,\theta,\sigma_x,\sigma_y)=\frac{1}{2\pi\sigma_x\sigma_y}\exp\left[\frac{-1}{2}\left(\left(\frac{x}{\sigma_x}\right)^2+\left(\frac{y}{\sigma_y}\right)^2\right)+j\omega(x\cos\theta+y\sin\theta)\right]$$

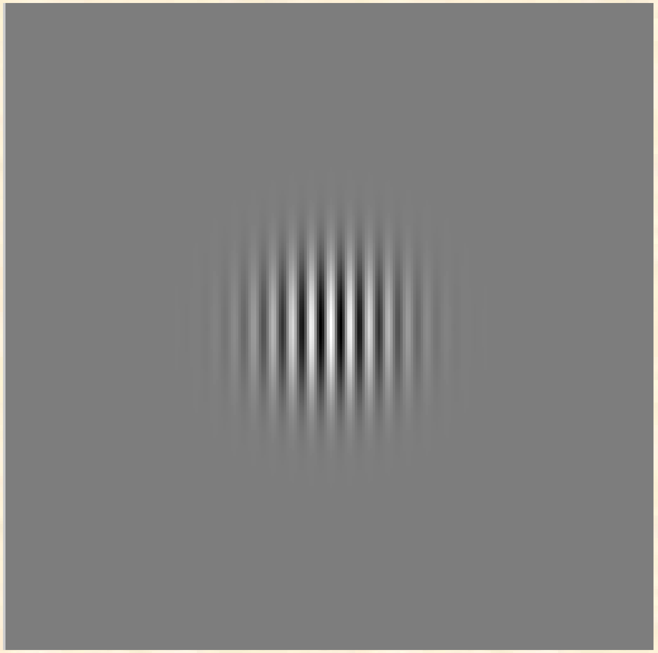


**A typical Gaussian filter with σ=30**

**A typical Gabor filter with σ=30, ω=3.14 and θ=45°**

< Prev

Next >



**Gabor filters with different combinations of**
**spatial width σ, frequency ω and orientation θ.**

**2-D Gabor filter:**

$$f(x,y,\omega,\theta,\sigma_x,\sigma_y)=\frac{1}{2\pi\sigma_x\sigma_y}\exp\left[\frac{-1}{2}\left(\left(\frac{x}{\sigma_x}\right)^2+\left(\frac{y}{\sigma_y}\right)^2\right)+j\omega(x\cos\theta+y\sin\theta)\right]$$

where

σ is the spatial spread

ω is the frequency

θ is the orientation

**1-D Gabor filter:**

$$f(x,\omega,\sigma)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left(\frac{-x^2}{2\sigma^2}+j\omega x\right)$$

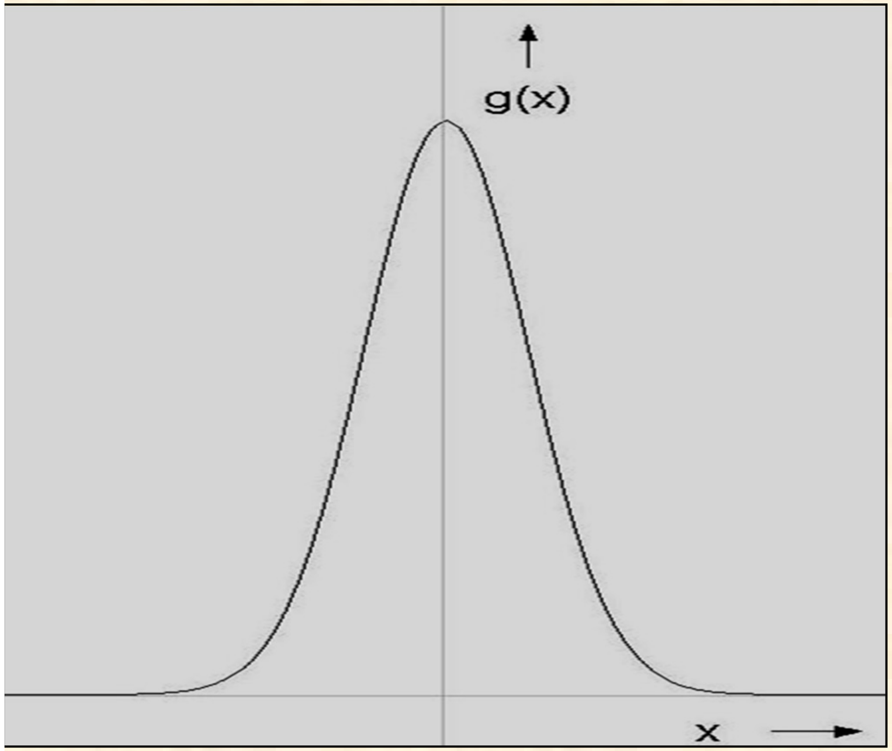
**1-D Gaussian function:**

$$g(x)=\frac{1}{\sqrt{2\pi}\sigma_1}\exp\left(\frac{-x^2}{2\sigma^2}\right)$$

# Processing of Texture-like Images

## 1-D Gaussian Filter

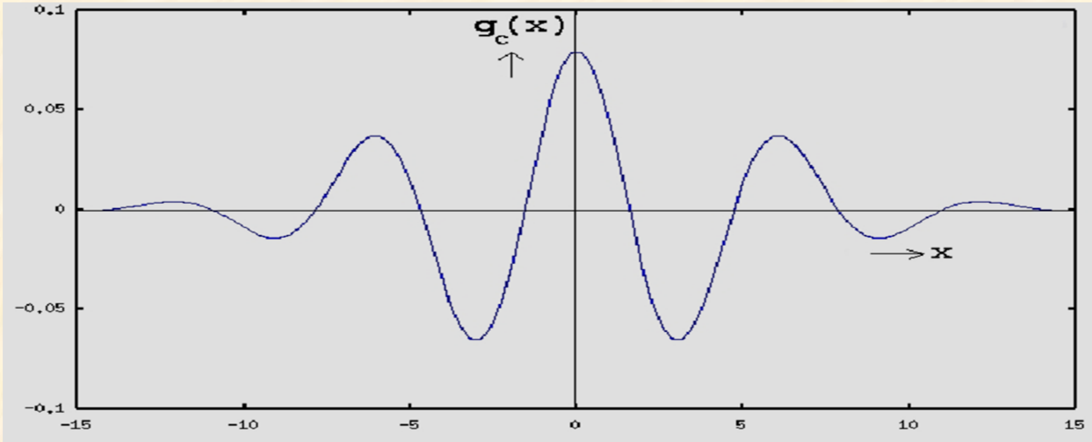
$$g(x) = \frac{1}{\sqrt{2\pi\sigma_1}} \exp\left(\frac{-x^2}{2\sigma^2}\right)$$



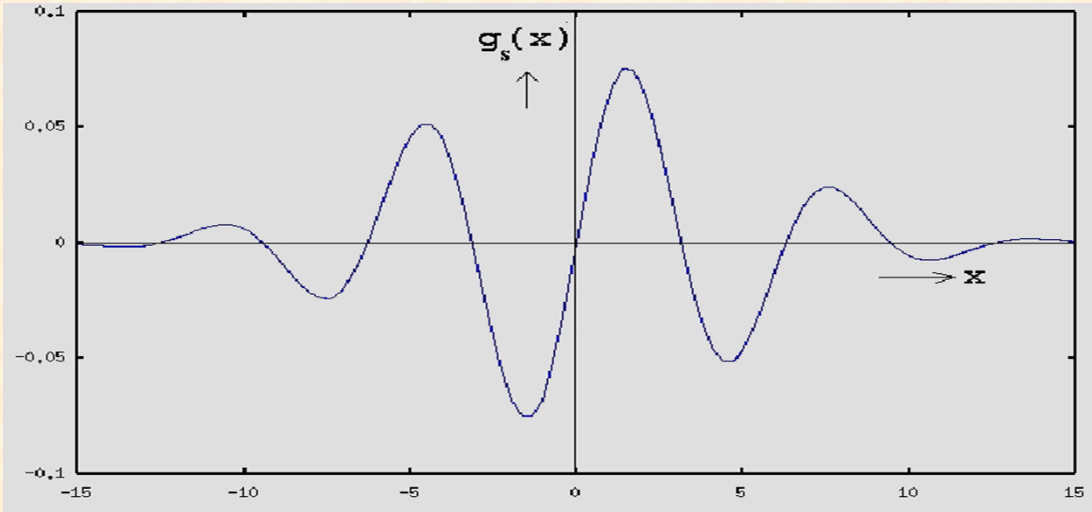

## 1-D Gabor Filter

$$f(x,\omega,\sigma) = \frac{1}{\sqrt{2\pi\sigma}} \exp(\frac{-x^2}{2\sigma^2} + j\omega x)$$

**Even Component**

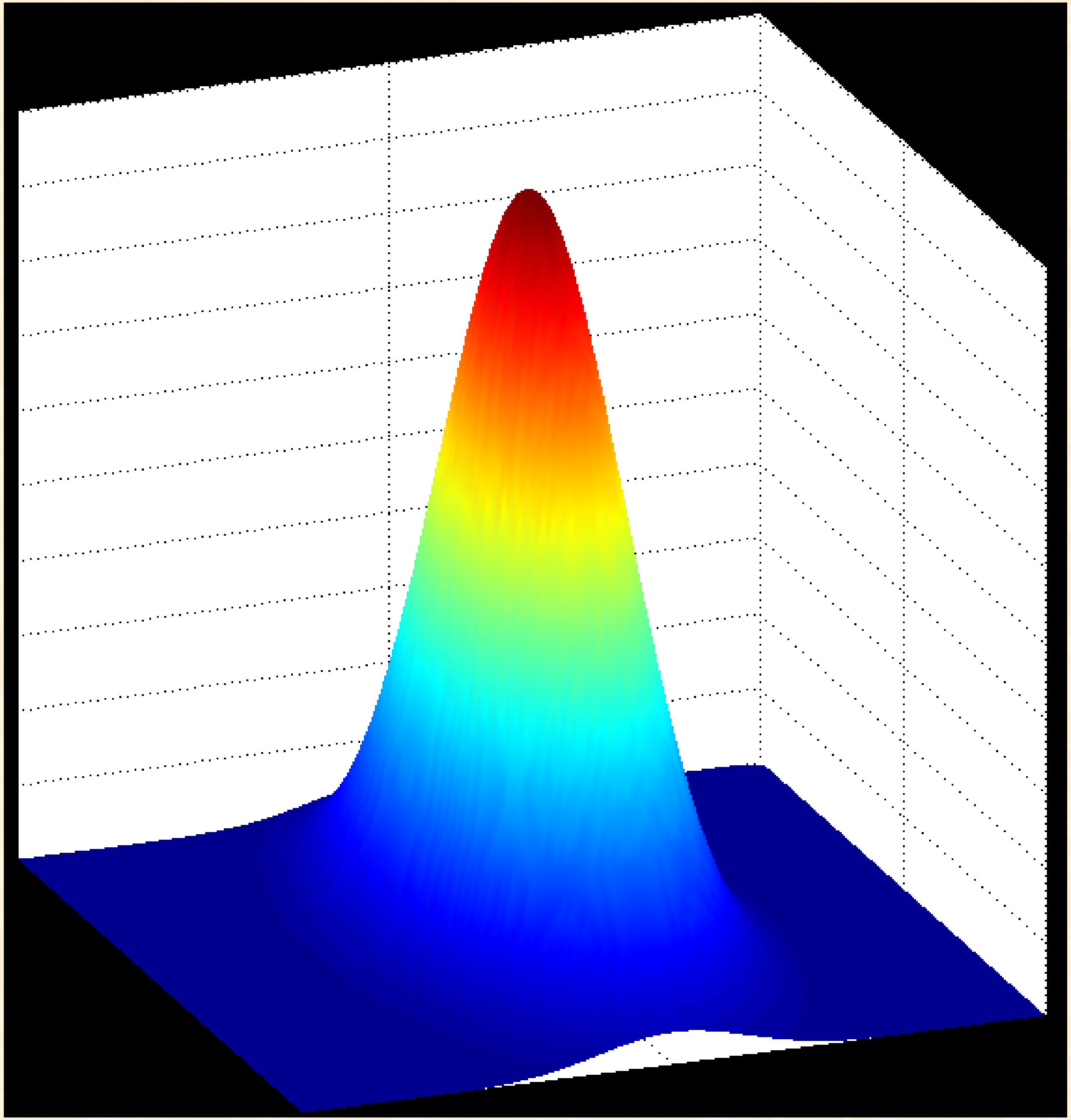



**Odd Component**




< Prev

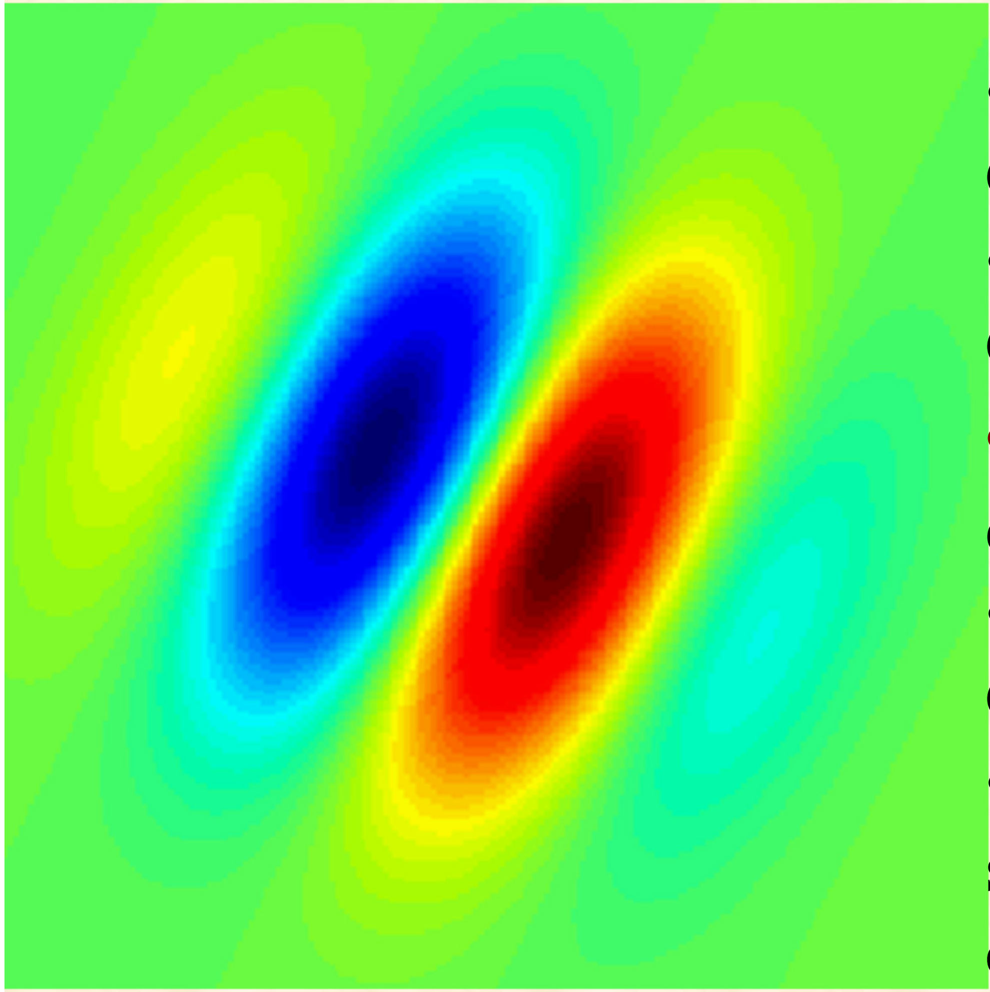
Next >



**Asymmetric 2-D Gaussian function**

$$gab(x,y) = K\exp(-\pi(a^2(x-x_0)_\theta^2 + b^2(y-y_0)_\theta^2))$$

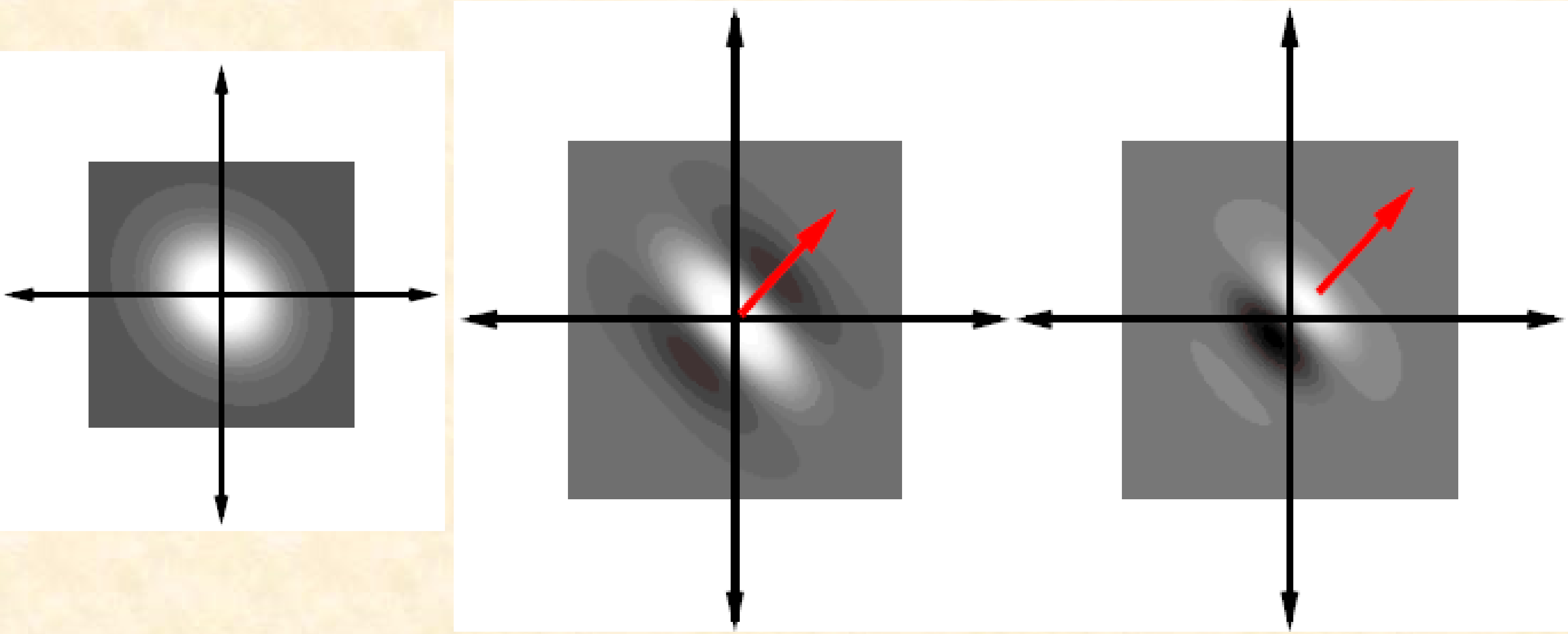
$$\exp(j(2\pi F_0(x\cos\omega_0 + y\sin\omega_0) + P)$$



- **K** : Scales the magnitude of the Gaussian envelop.
- **(a, b)** : Scale the two axis of the Gaussian envelop.
- **θ** : (Rotation) angle of the Gaussian envelop.
- **($x_0$, $y_0$)** : Location of the peak of the Gaussian envelop.
- **($u_0$, $v_0$)** : Spatial frequencies of the sinusoidal carrier in Cartesian coordinates. It can also be expressed in polar coordinates as **($F_0$, $\omega_0$).**
- **P** : Phase of the sinusoidal carrier.

$$gab(x,y) =$$

$$K\exp(-\pi(a^2(x-x_0)_\theta^2 + b^2(y-y_0)_\theta^2) + j(2\pi(u_0x + v_0y) + P)$$



**Asymmetrical Gaussian of 128×128 pixels. The parameters are as follows:**

**$x_0 = y_0 = 0$; a = 1/50 pixels; b = 1/40 pixels; $\theta = -45$ deg.**

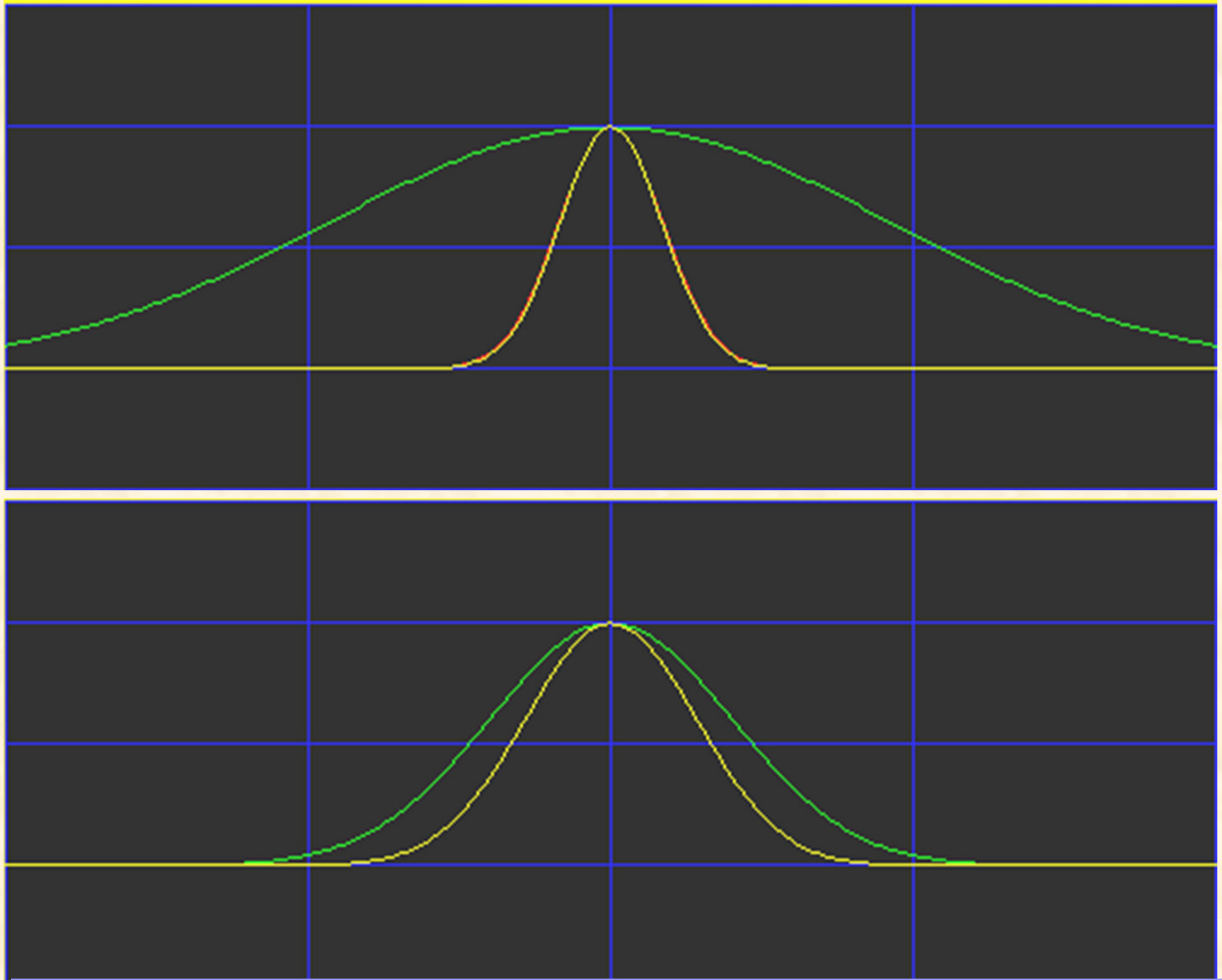
**The real and imaginary parts of a complex Gabor function in space domain, with**

***$F_0$ = sqrt(2)/80 cycles/pixel, $\omega_0$= 45 deg, P = 0 deg.***

## Gaussian and its integral

$$\int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{\pi}$$

$$\int_{-\infty}^{\infty} ae^{-\frac{(x-b)^2}{2c^2}}\,dx = ac \cdot \sqrt{2\pi}.$$



## Fourier transform of the Gaussian

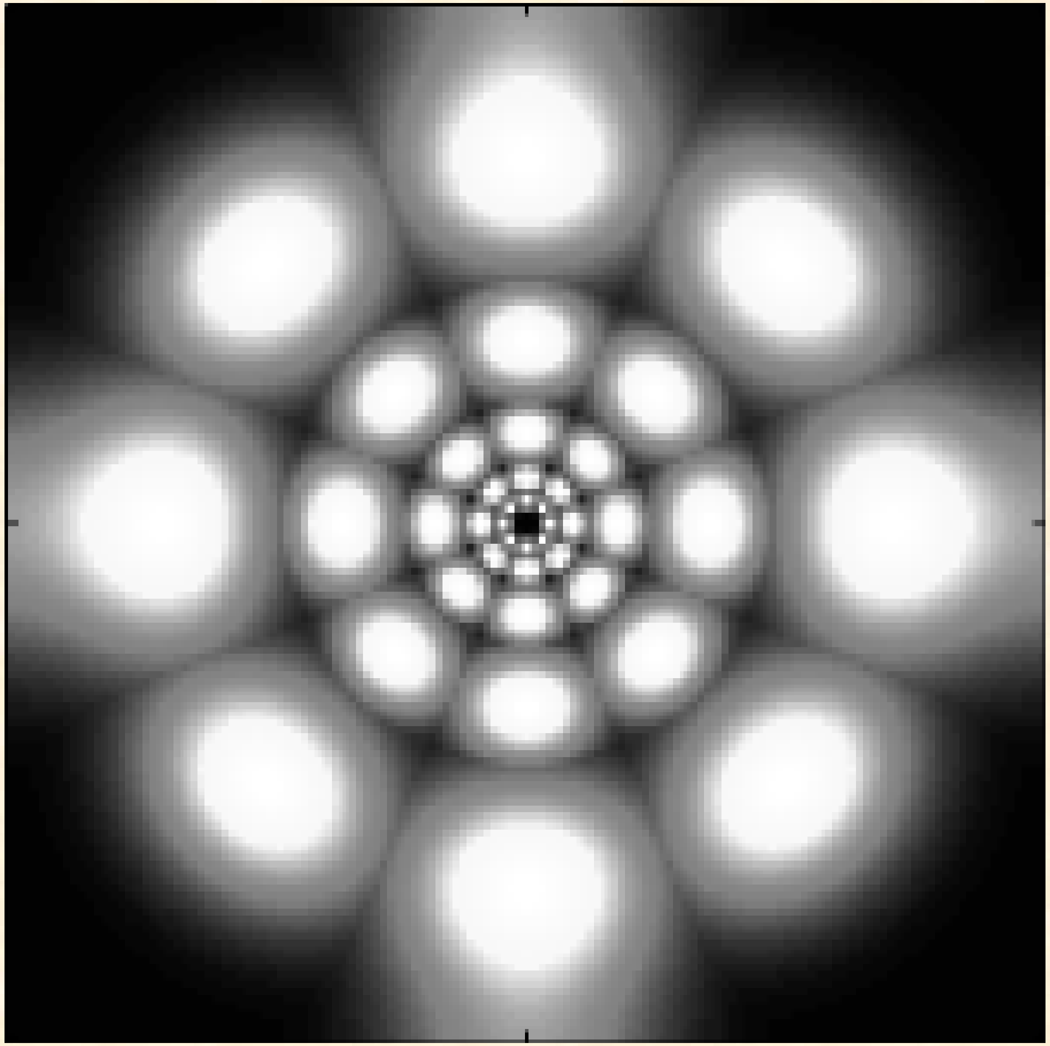
$$g(t) = e^{-t^2/a}$$

$$G(u) =$$

$$(\sqrt{a\pi})e^{-au^2/4}$$

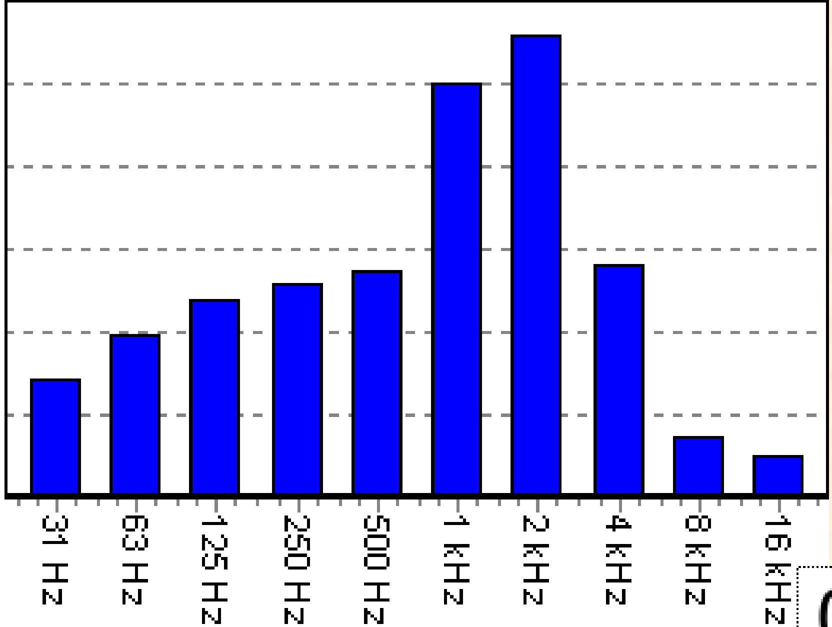
## Fourier transform of the GABOR ??

Green – Gaussian; Yellow - FT of Gaussian

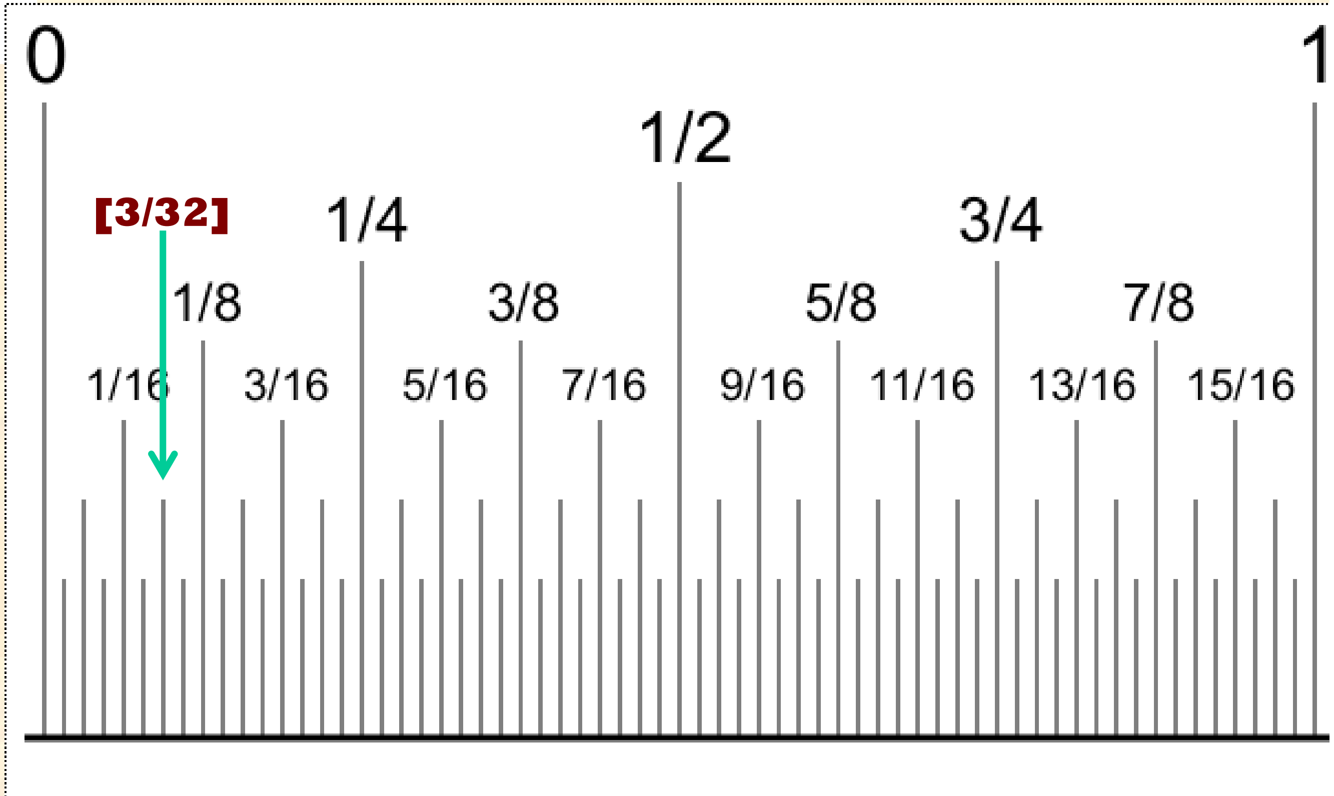


**The frequency response of a typical *dyadic* bank of Gabor filters. One center-symmetric pair of lobes in the illustration represents each filter.**

# Octave bands, due to Dyadic decomposition Filter bank




*Read more about –*
*in wavelets, QMFB*
*and Q-factor*
*etc.*



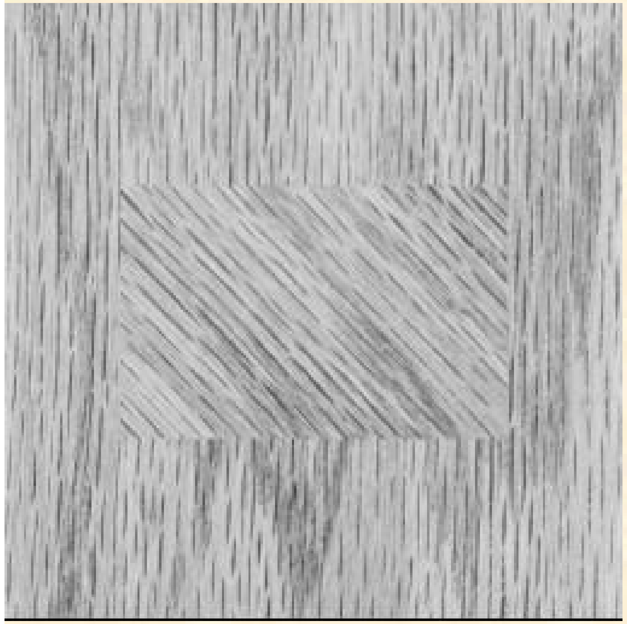

**Octave bands:** ..(.., 0.0625); (0.0625, 0.125); (0.125, 0.25); (0.25, 0.5); (0.5, 1).

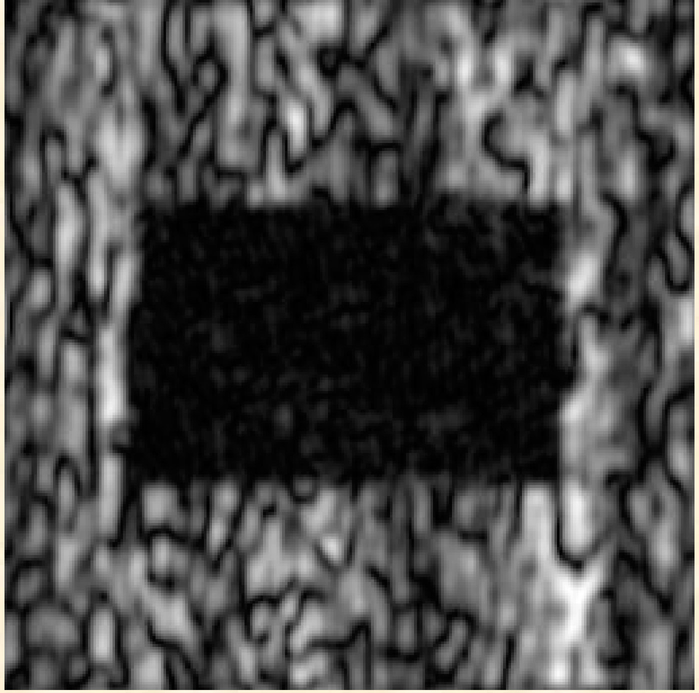
**Center frequencies:** .... 0.0938 (3/32); 0.1875 (3/16); 0.375 (3/8); 0.75 (3/4).

**Some properties of Gabor filters:**

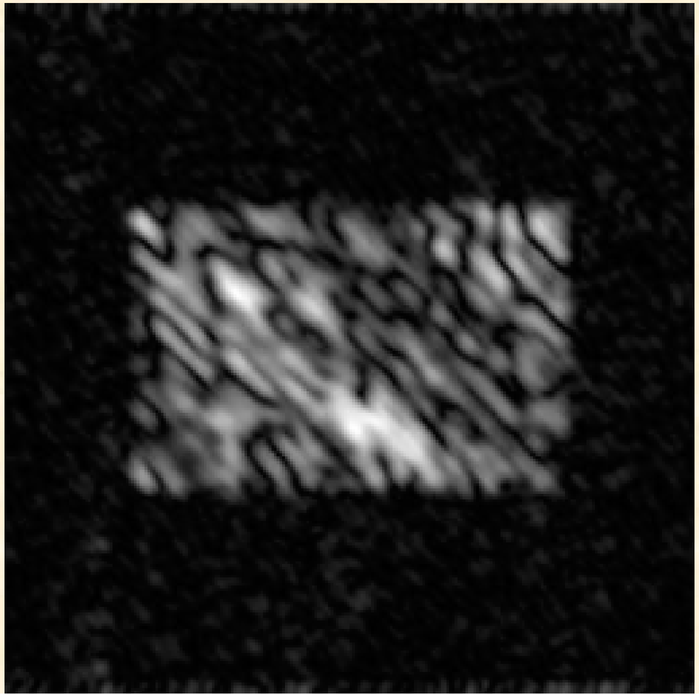
- **A tunable bandpass filter**
- **Similar to a STFT or windowed Fourier transform**
- **Satisfies the lower-most bound of the time-spectrum resolution (uncertainty principle)**
- **It's a multi-scale, multi-resolution filter**
- **Has selectivity for orientation, spectral bandwidth and spatial extent.**
- **Has response similar to that of the Human visual cortex (first few layers of brain cells)**
- **Used in many applications – texture segmentation; iris, face and fingerprint recognition.**
- **Computational cost often high, due to the necessity of using a large bank of filters (or Gabor jet) in most applications**

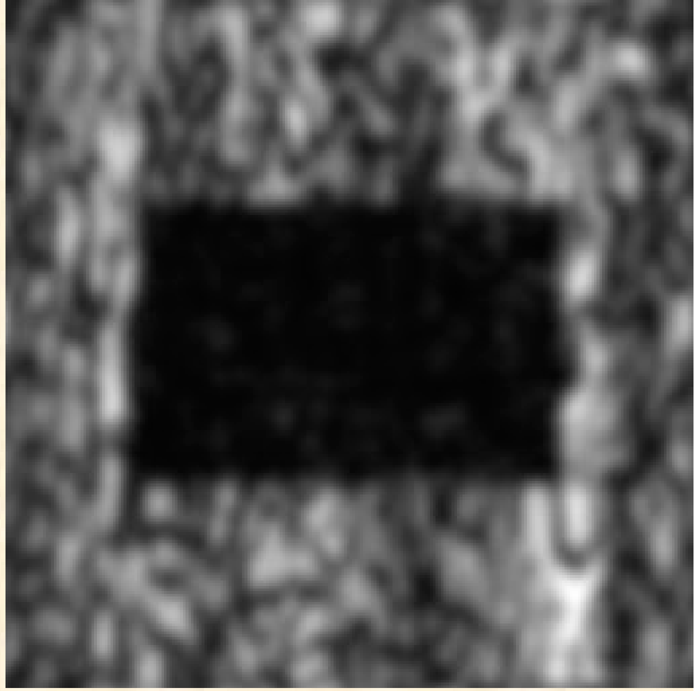
**Texture Image**



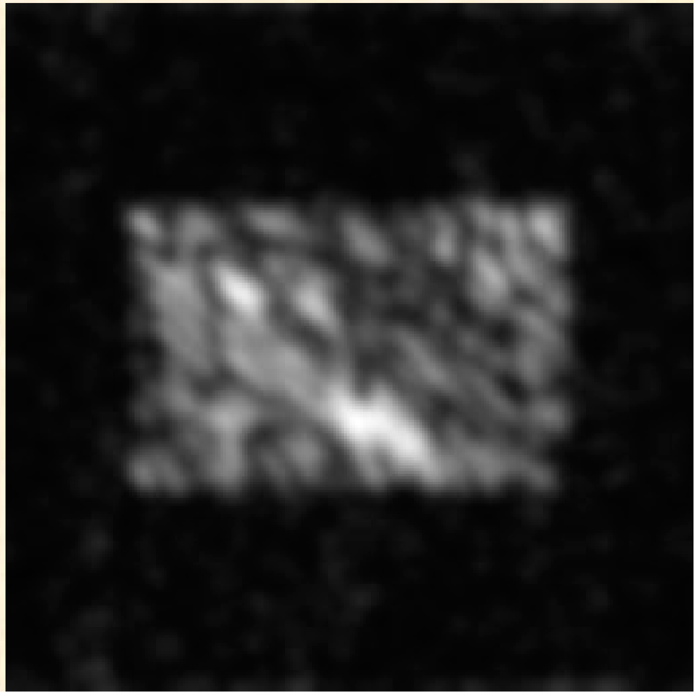


**Magnitude of the Gabor Responses**

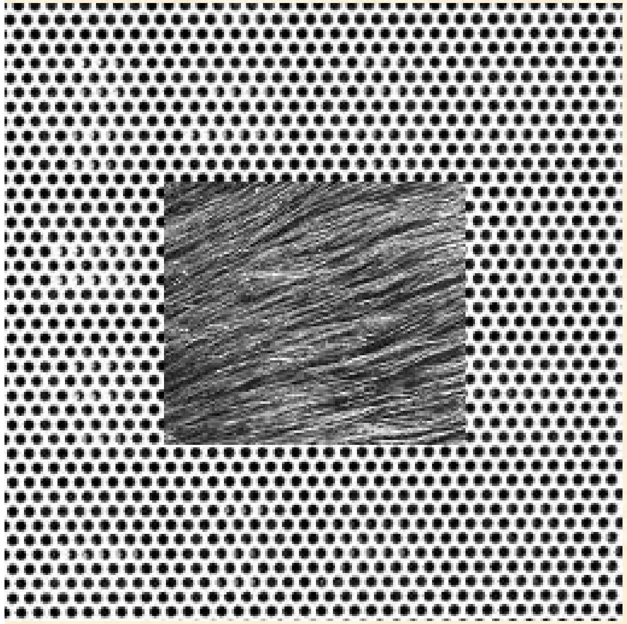


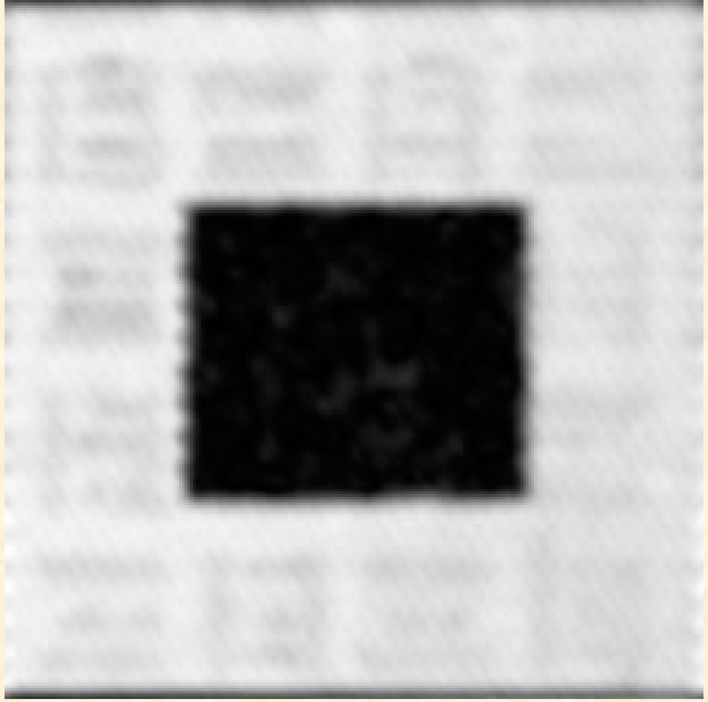


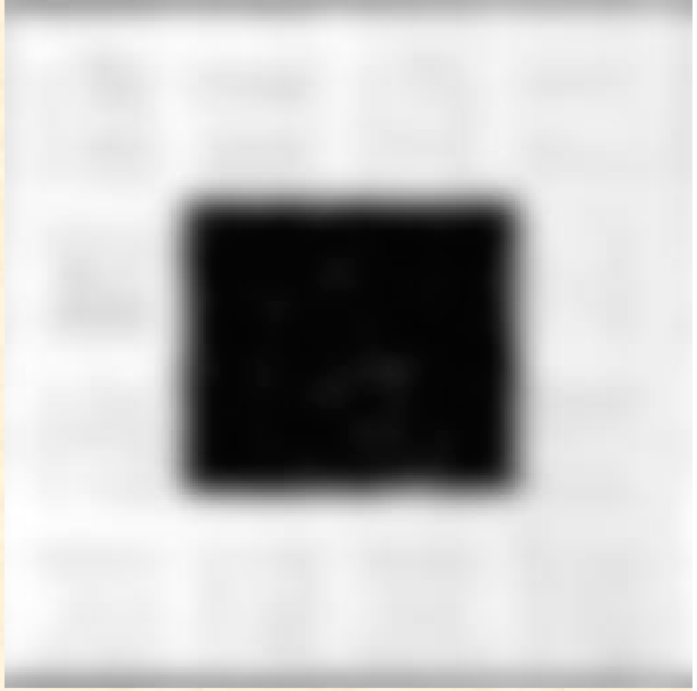
**Smoothed Features**



Texture Image

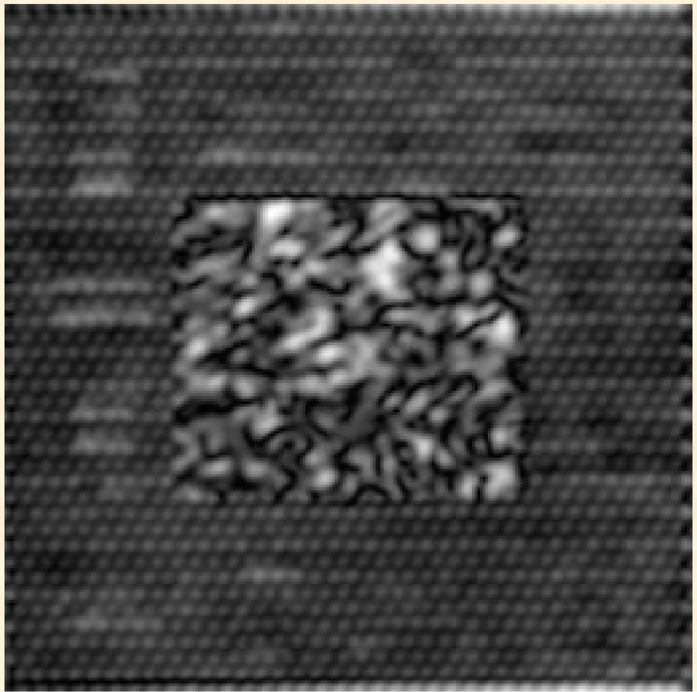


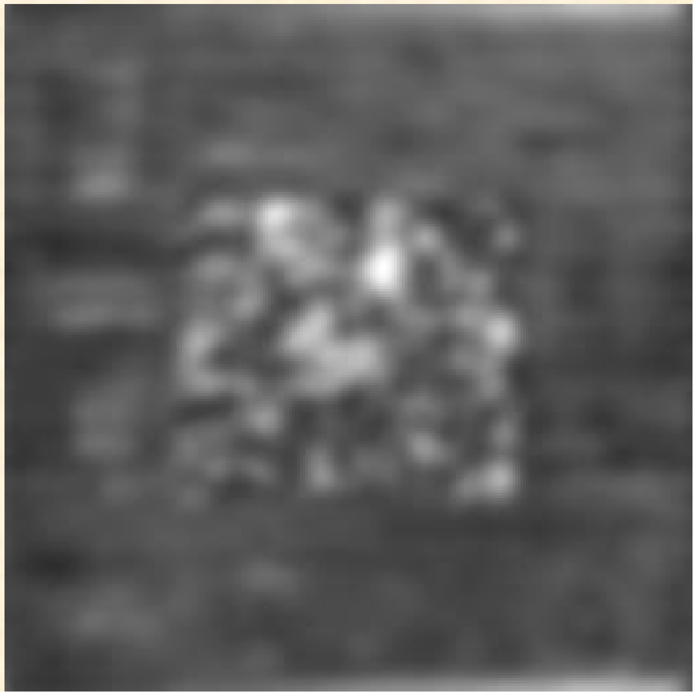




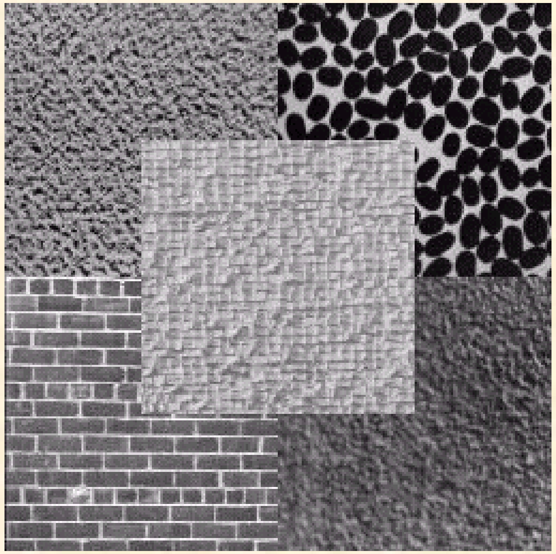
Magnitude of the Gabor Responses

Smoothed Features

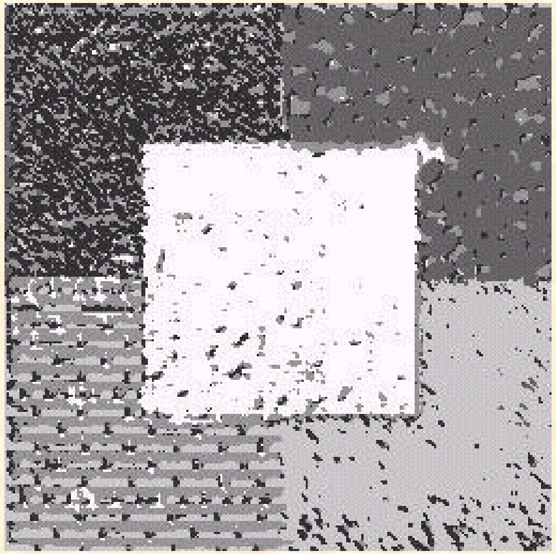




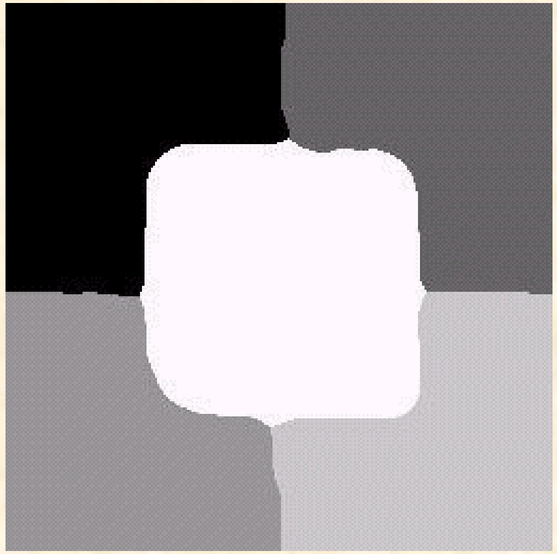
**Natural Textures**



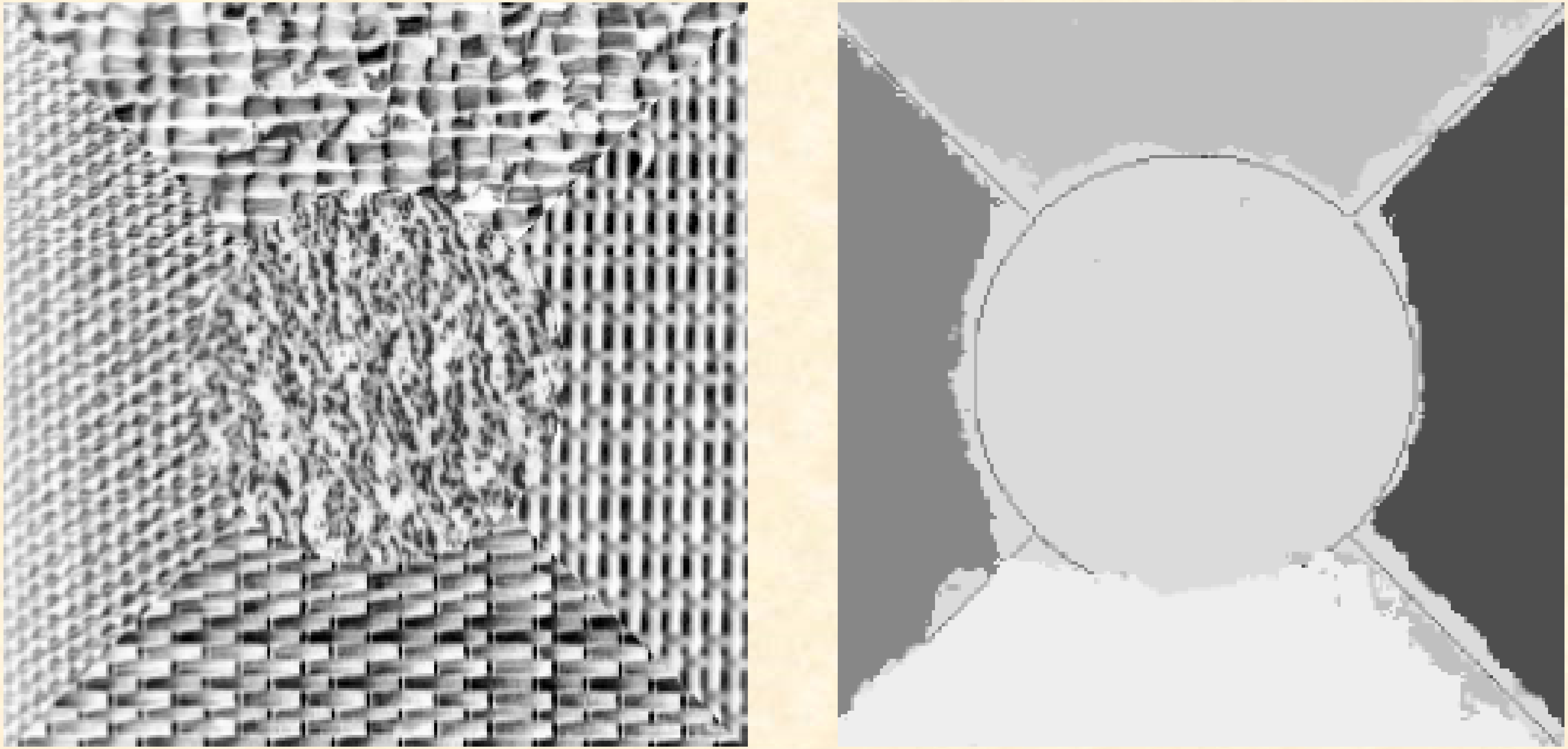
**Initial Classification**



**Final Classification**

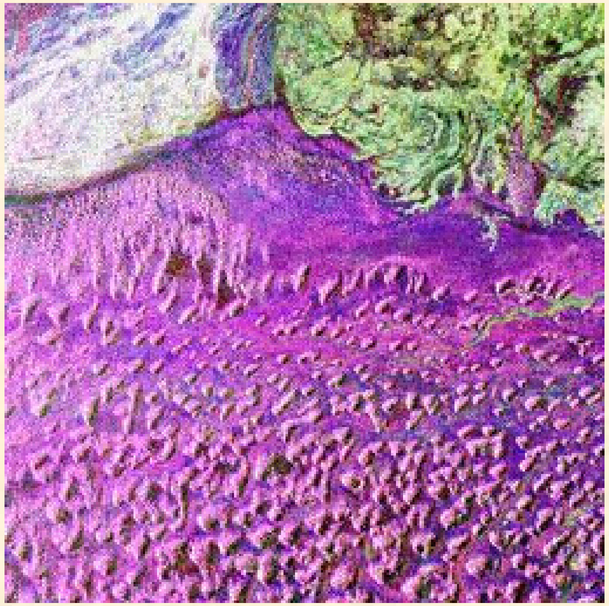


< Back

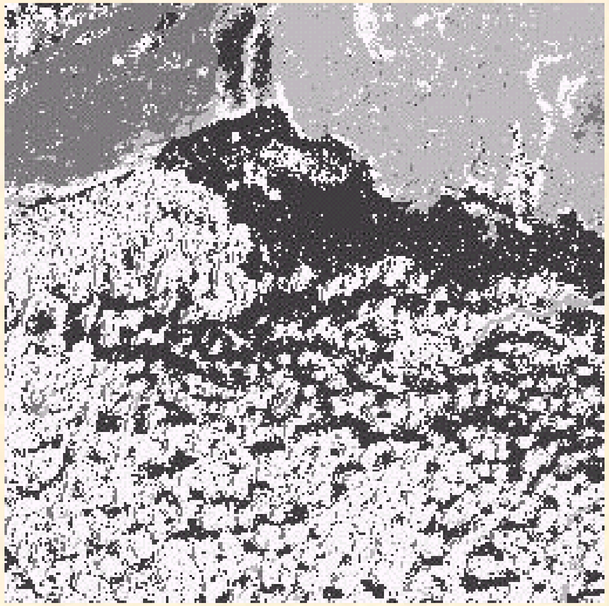


Segmentation using Gabor based features of a texture image containing five regions.

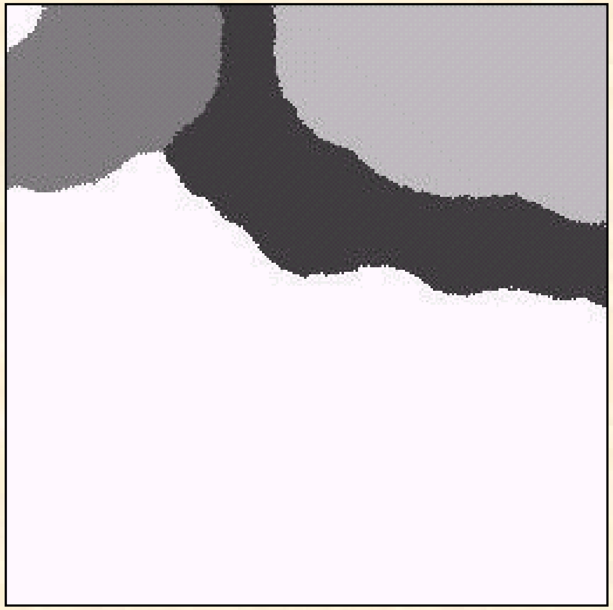
SIR-C/X-SAR image of Lost City of Ubar



Classification using multispectral information



Classification using multispectral and textural information



< Back

- Filtering methods:
  - Discrete Wavelet Transform (DWT) (Daubechies 8-Taps)
  - Gabor Filter (Bank of 8 Gabor filters)
  - Discrete Cosine Transform (DCT) (9 filters) Ref: [Ng 92]
  - Gaussian Markov random field models Ref: [Cesmeli 2001]
  - Combination of DWT and Gabor filter Ref: [Rao 2004]
  - Combination of DWT and MRF Ref: [Wang 99]
- Non-linearity:
  - Magnitude (| . |)
- Smoothing:
  - Gaussian filter
- Feature vectors:
  - Mean (computed in a local window around a pixel)
- Classification:
  - Fuzzy-*C* Means (FCM) (unsupervised classifier)

• Combine Edge and region map using a CSNN

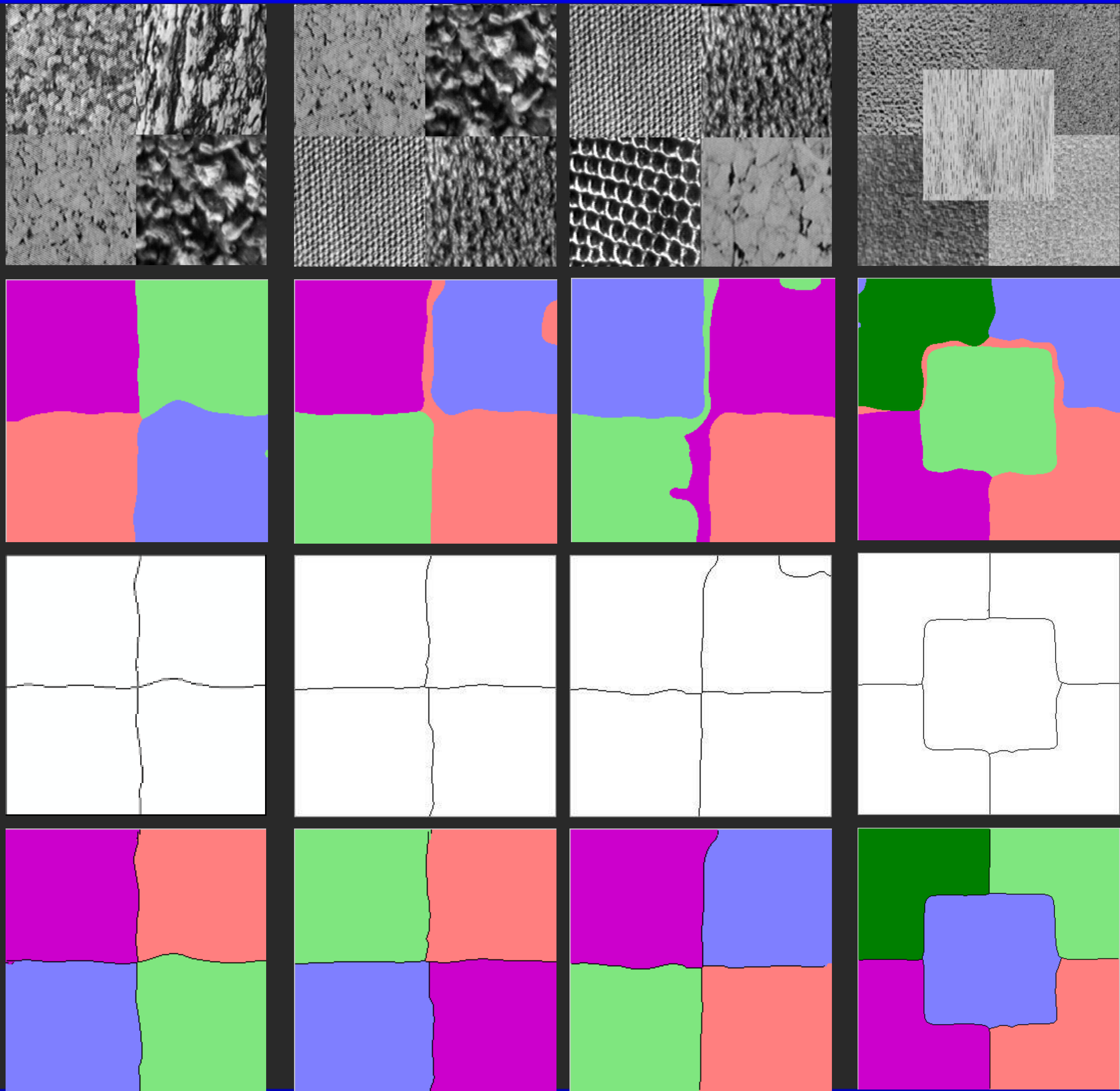
# Results

Input Image

Segmented map before integration

Edge map before integration

Segmented map and Edge map after integration



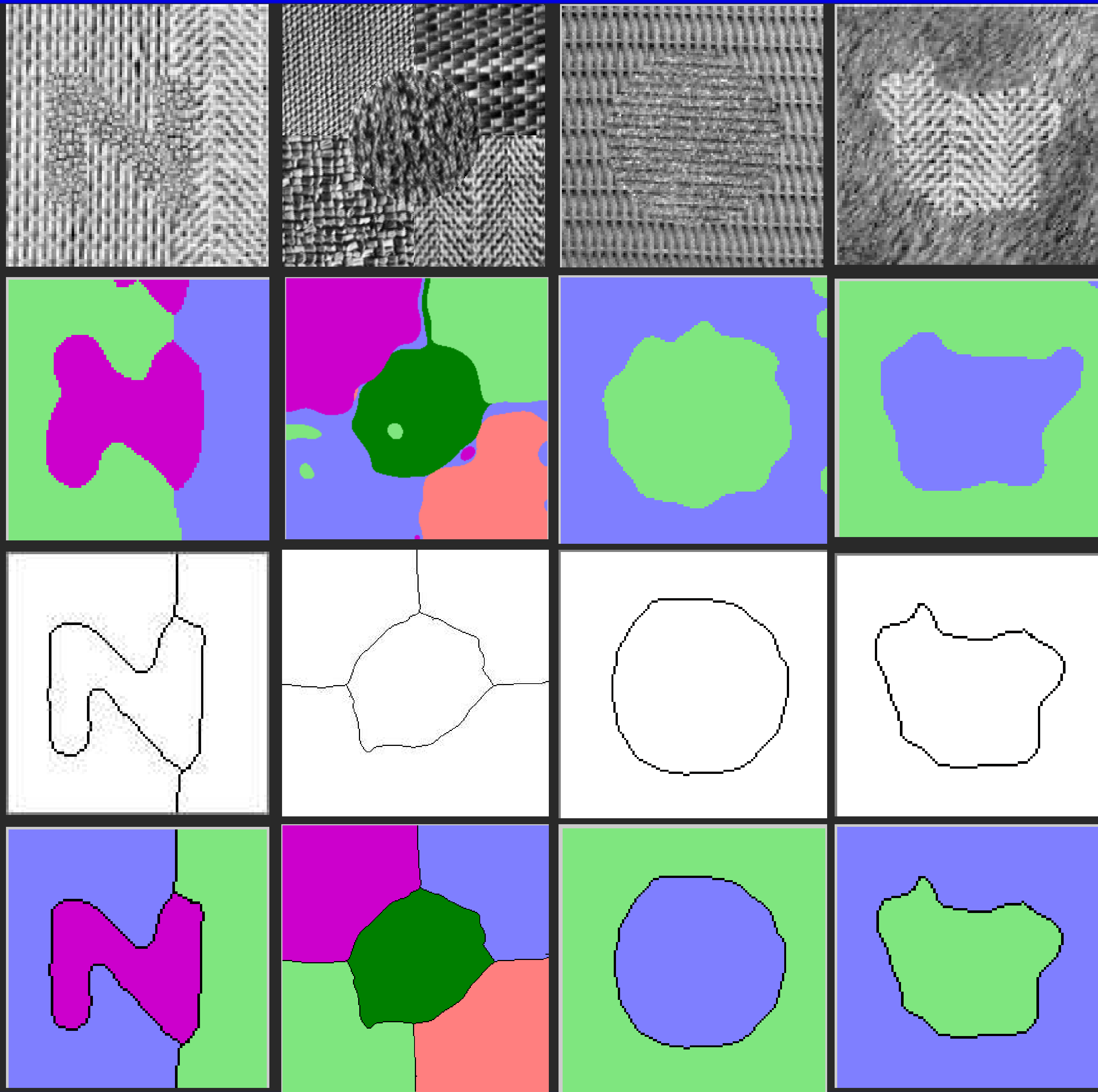
# Results

Input Image

Segmented map before integration

Edge map before integration

Segmented map and Edge map after integration



## GLCM based texture feature (statistical)

The **Grey Level Co-occurrence Matrix, GLCM**
(also called the **Grey Tone Spatial Dependency Matrix**)

**The GLCM is a tabulation of how often different combinations of pixel brightness values (grey levels) occur in an image.**

**The GLCM is usually defined for a series of "second order" texture calculations.**

**Second order means they consider the relationship between groups of two pixels in the original image.**

**First order texture measures are statistics calculated from the original image values, like variance, and do not consider pixel relationships. Third and higher order textures (considering the relationships among three or more pixels) are theoretically possible but not implemented due to calculation time and interpretation difficulty.**

| | | | |
|---|---|---|---|
| 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 2 | 2 | 2 |
| 2 | 2 | 3 | 3 |

*A small image neighborhood*

| NPV> RPV | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | | | | |
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |

*Neighbor/Reference Pixel Value*

**Spatial relationship between two pixels:**

**GLCM texture considers the relation between two pixels at a time, called the reference and the neighbour pixel. Let, the neighbour pixel is chosen to be the one to the east (right) of each reference pixel. This can also be expressed as a (1,0) relation: *(i, j) -> (i+1, j).***

**Each pixel within the window becomes the reference pixel in turn, starting in the upper left corner and proceeding to the lower right.**

| | | | |
|---|---|---|---|
| 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 2 | 2 | 2 |
| 2 | 2 | 3 | 3 |

*A small image neighborhood*

**(-1,0) relation:** ***(i, j) -> (i-1, j).***

***EAST GLCM***

| NPV> RPV | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 2 | 2 | 1 | 0 |
| 1 | 0 | 2 | 0 | 0 |
| 2 | 0 | 0 | 3 | 1 |
| 3 | 0 | 0 | 0 | 1 |

***WEST GLCM (why transpose ?)***

| NPV> RPV | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 2 | 0 | 0 | 0 |
| 1 | 2 | 2 | 0 | 0 |
| 2 | 1 | 0 | 3 | 0 |
| 3 | 0 | 0 | 1 | 1 |

**Symmetrical (1,0) GLCM**

| NPV> RPV | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 4 | 2 | 1 | 0 |
| 1 | 2 | 4 | 0 | 0 |
| 2 | 1 | 0 | 6 | 1 |
| 3 | 0 | 0 | 1 | 2 |

## Expressing the GLCM as a probability:

Symmetrical (1,0) GLCM

| NPV> / RPV | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 4 | 2 | 1 | 0 |
| 1 | 2 | 4 | 0 | 0 |
| 2 | 1 | 0 | 6 | 1 |
| 3 | 0 | 0 | 1 | 2 |

This is the number of times this outcome occurs, divided by the total number of possible outcomes.

This process is called normalizing the matrix. Normalization involves dividing by the sum of values.

$$P_{i,j} = \frac{V_{i,j}}{\sum_{i,j=0}^{N-1} V_{i,j}}$$

| | | | |
|---|---|---|---|
| .166 (4/24) | .083 (2/24) | .042 (1/24) | 0 (0/24) |
| .083 | .166 | 0 | 0 |
| .042 | 0 | .25 | .042 |
| 0 | 0 | .042 | .083 |

*A small image neighborhood*

| | | | |
|---|---|---|---|
| 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 2 | 2 | 2 |
| 2 | 2 | 3 | 3 |

**Find,**
**(1,0), South GLCM (solve it):**

| | | | |
|---|---|---|---|
| 3 | 0 | 2 | 0 |
| 0 | 2 | 2 | 0 |
| 0 | 0 | 1 | 2 |
| 0 | 0 | 0 | 0 |

**Normalized symmetrical vertical GLCM**

| | | | |
|---|---|---|---|
| .250 | 0 | .083 | 0 |
| 0 | .166 | .083 | 0 |
| .083 | .083 | .083 | .083 |
| 0 | 0 | .083 | 0 |

***Any reason for Diagonal dominance ?***

# References


1. Haralick, R.M. 1979. Statistical and Structural Approaches to Texture. Proceedings of the IEEE, 67:786-804; (also 1973, IEEE-T-SMC)
2. Simona E. Grigorescu, Nicolai Petkov, and Peter Kruizinga; Comparison of Texture Features Based on Gabor Filters; IEEE Transactions on Image Processing, Vol. 11, No. 10, OCT. 2002; pp 1160-1167.
3. J. Randen and J. S. Husoy, Texture Segmentation using filters with Optimized Energy Separation, IEEE Transactions on Image Processing, Vol. 8, No. 4, April 1999, pp 571-582.
4. M. L. Comer and E. J. Delp, Segmentation of Textured Images using a Multiresolution Gaussian Autoregressive Model, IEEE Transactions on Image Processing, Vol. 8, No. 3, March 1999, pp 408-420.
5. B. Thai and G. Healey, Optimal Spatial Filter Selection for Illumination-Invariant Color Texture Discrimination, IEEE Trans. Systems, Man and Cybernetics, Part B: Cybernetics, Vol. 30, No. 4, August 2000, pp 610-615.
6. J. Randen and J. S. Husoy, "Optimal Filter-bank Design for Multiple texture Discrimination", Proceedings of the International Conference on Image Processing (ICIP '97), pp 215-218.
7. "Integrating Region and Edge Information for Texture Segmentation using a modified Constraint Satisfaction Neural Network", Lalit Gupta, Utthara Gosa Mangai and Sukhendu Das, Image and Vision Computing, Vol. 26, No. 8, pp. 1106-1117, August 2008; #Citations – 16.