Segmentation & Texture Analysis

Sedat Ozer
Department of Computer Engineering
Bilkent University
sedat@cs.bilkent.edu.tr

Texture

- An important approach to image description is to quantify its texture content.
- Texture gives us information about the spatial arrangement of the colors or intensities in an image.

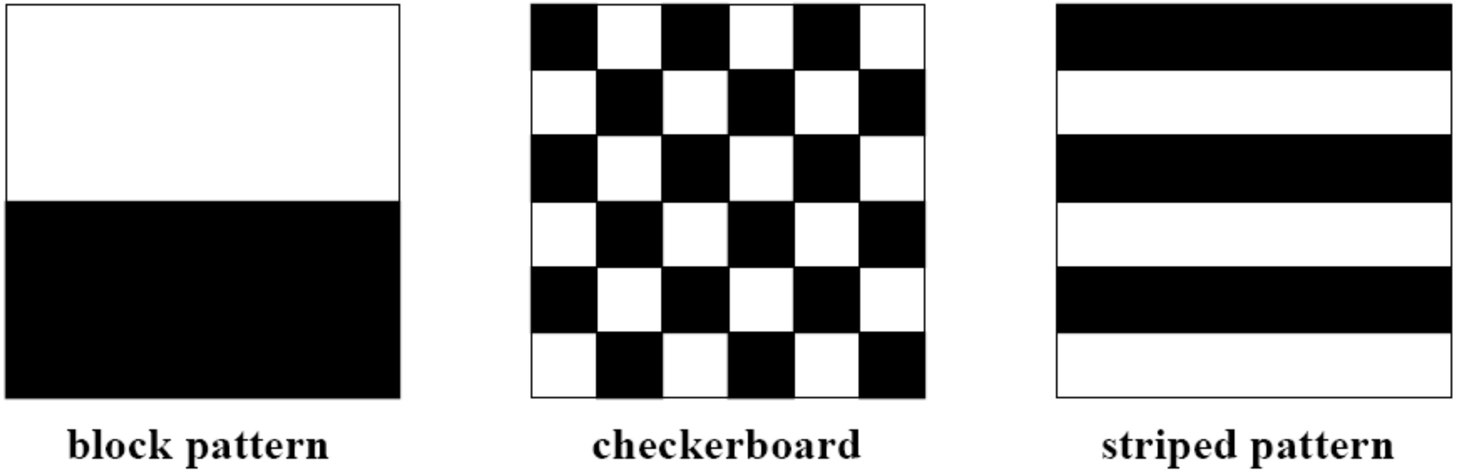
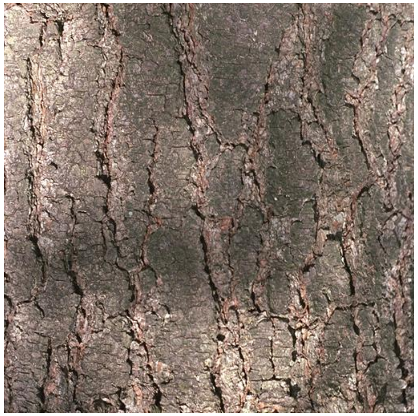


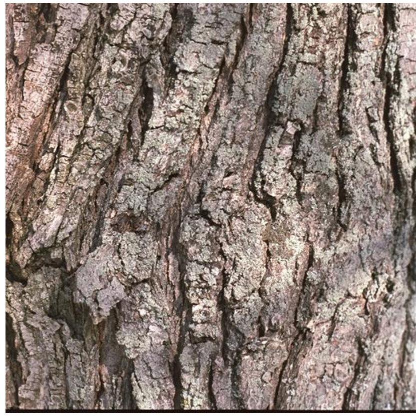
Figure 7.2: Three different textures with the same distribution of black and white.

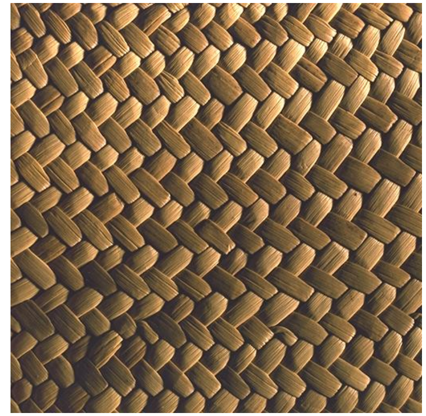
Texture

- Although no formal definition of texture exists, intuitively it can be defined as the uniformity, density, coarseness, roughness, regularity, intensity and directionality of discrete tonal features and their spatial relationships.
- Texture is commonly found in natural scenes, particularly in outdoor scenes containing both natural and man-made objects.

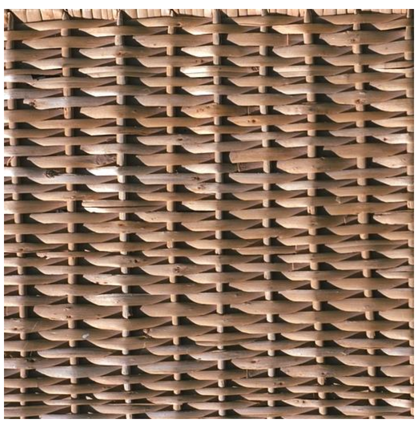
Texture


Bark


Bark


Fabric


Fabric

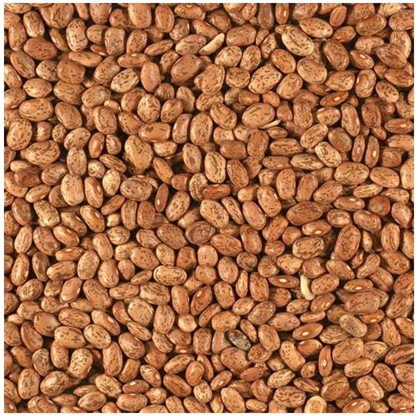

Fabric


Flowers


Flowers


Flowers

Texture


Food


Food


Leaves


Leaves


Leaves

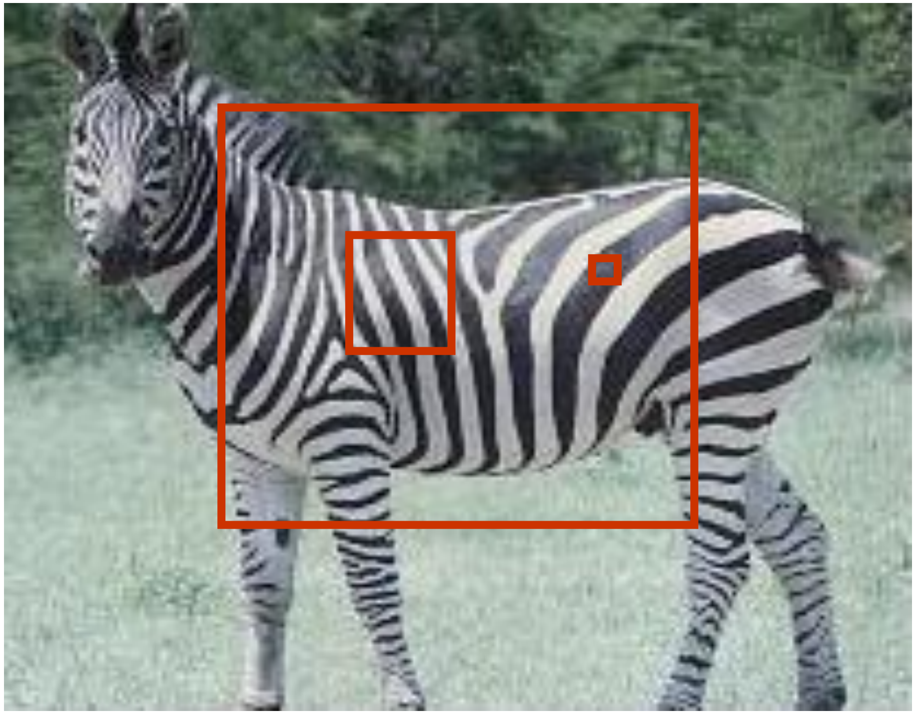

Leaves


Water


Water

Texture

- Whether an effect is a texture or not depends on the scale at which it is viewed.



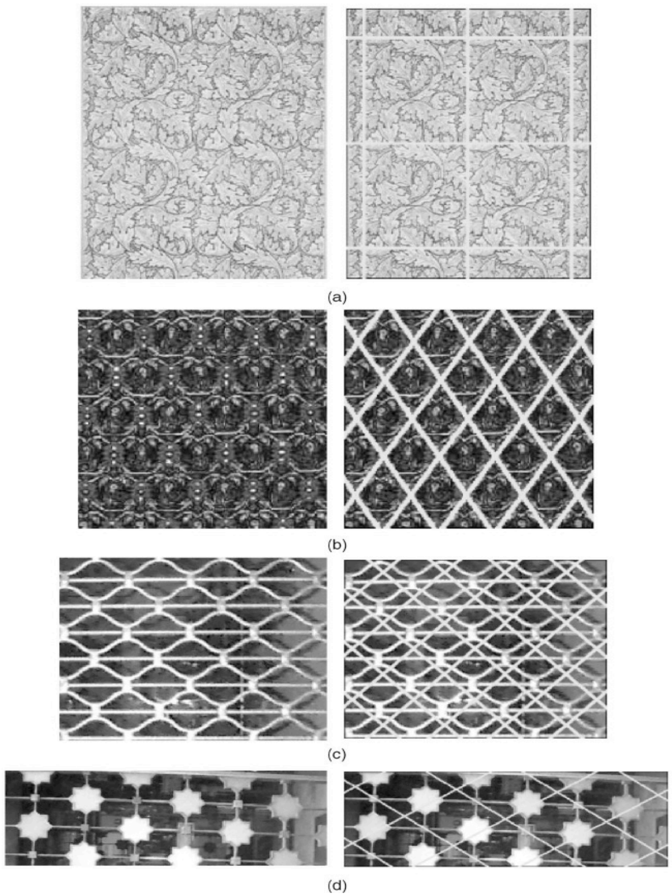
Texture

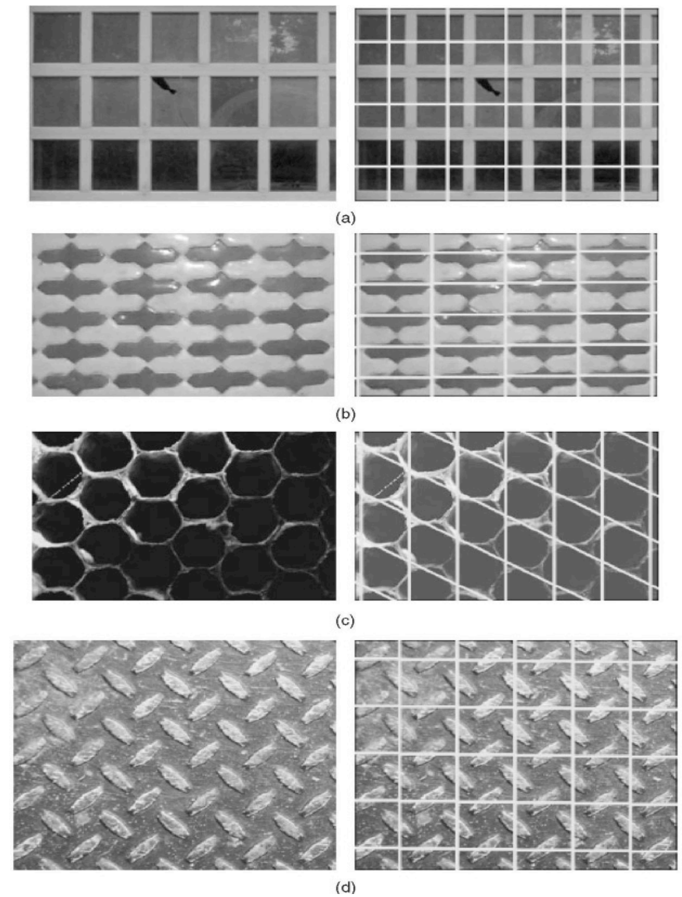
- The approaches for characterizing and measuring texture can be grouped as:
  - structural approaches that use the idea that textures are made up of primitives appearing in a near-regular repetitive arrangement,
  - statistical approaches that yield a quantitative measure of the arrangement of intensities.
- While the first approach is appealing and can work well for man-made, regular patterns, the second approach is more general and easier to compute, and is used more often in practice.

Structural approaches

- Structural approaches model texture as a set of texture primitives (also called texels (texture elements) or textons) in a particular spatial relationship (also called lattice or grid layout).
- A structural description of a texture includes a description of the primitives and a specification of their placement patterns.
- Of course, the primitives must be identifiable and their relationships must be efficiently computable.

Structural approaches

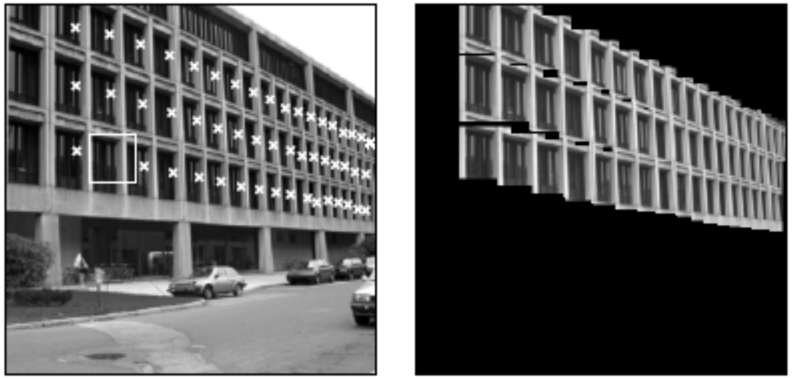




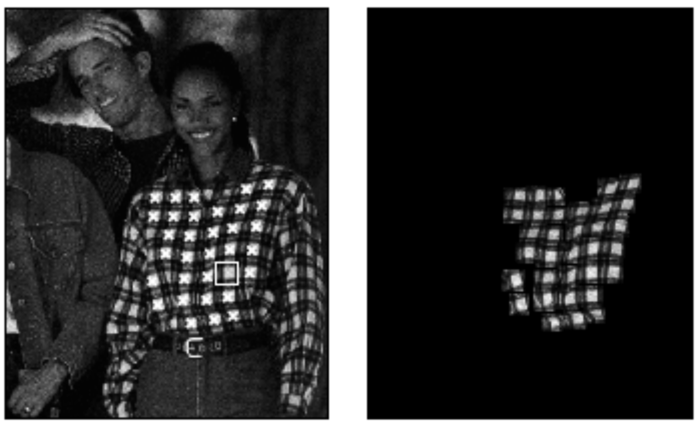
Examples of periodic patterns that are extended in two linearly independent directions to cover the 2D plane. These patterns are also known as wallpaper patterns.

Y. Liu, et al., "A Computational Model for Periodic Pattern Perception Based on Frieze and Wallpaper Groups", IEEE Trans. On Pattern Analysis and Machine Intelligence, 2004

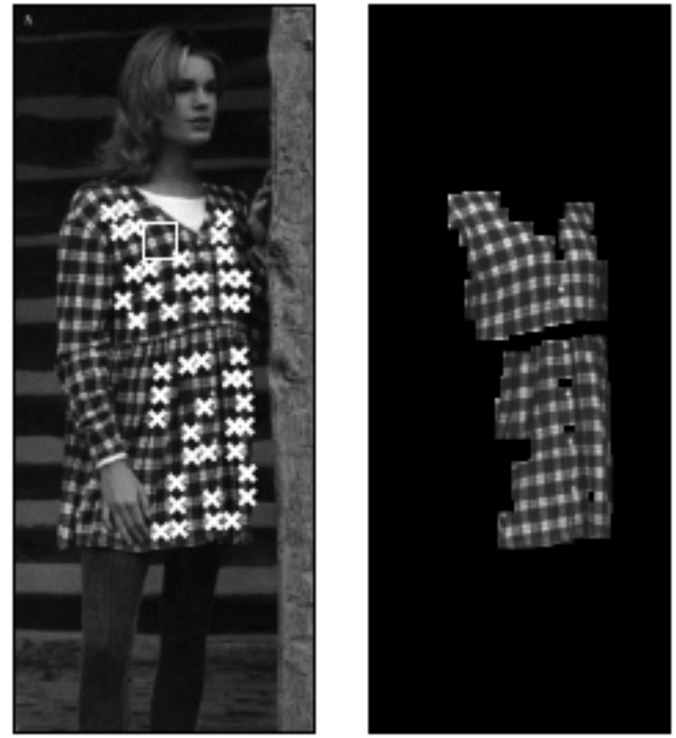
Structural approaches



(a) (b)



(a) (b)

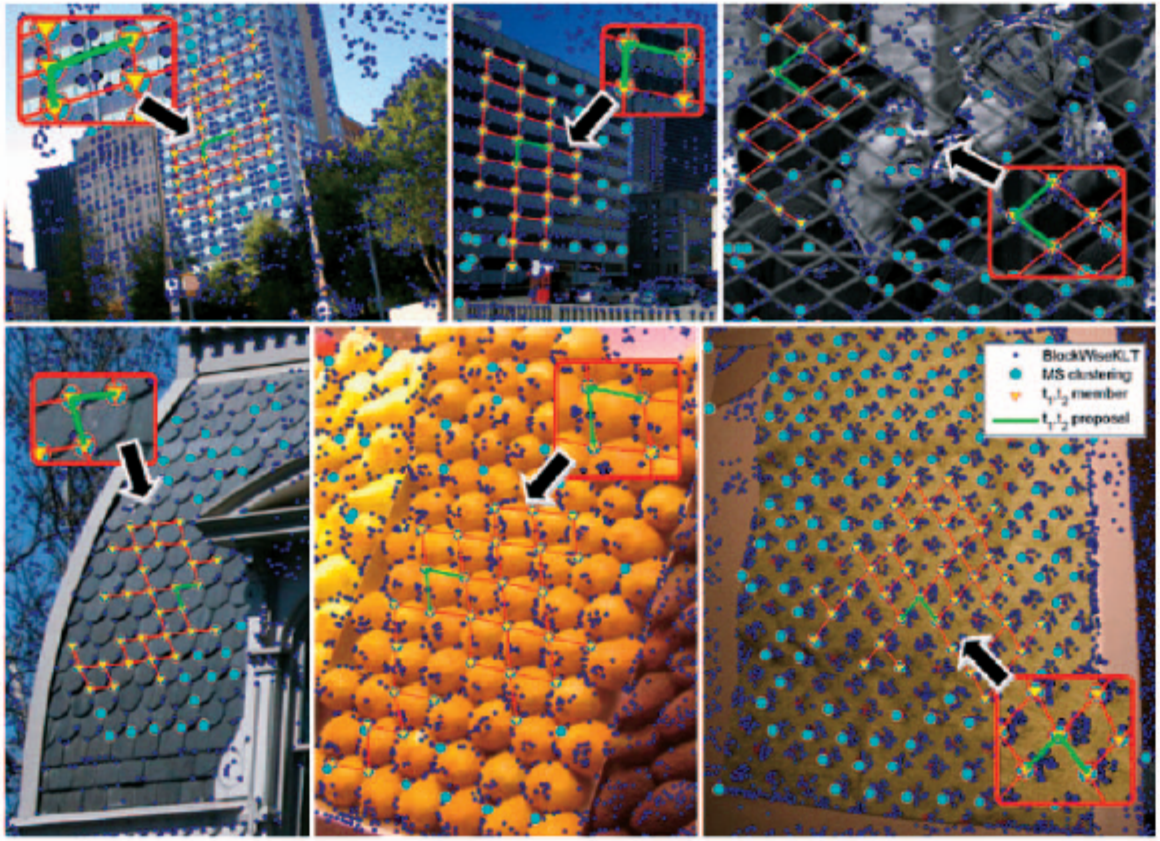


(a) (b)

A structural texture analysis method that involves detecting interesting elements in the image, matching elements with their neighbors, and grouping the elements.

T. Leung, J. Malik, "Detecting, Localizing and Grouping Repeated Scene Elements from an Image", ECCV 2004

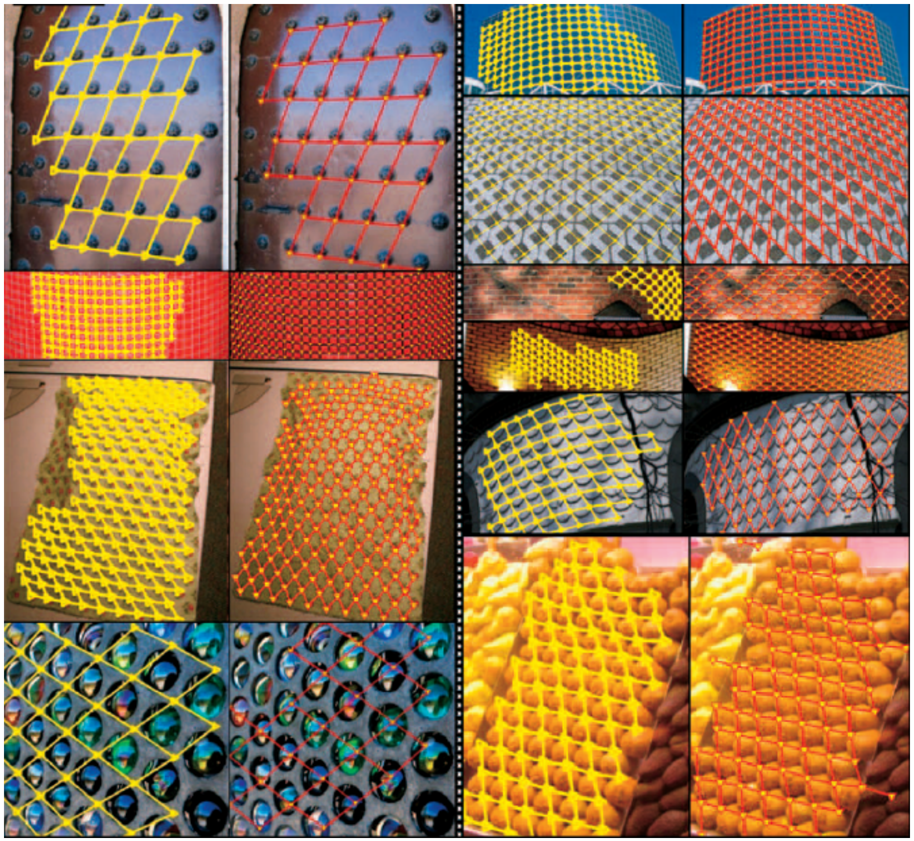
Structural approaches



A method that involves the detection of interest points, clustering of these points, voting for consistent lattice unit proposals, and iterative fitting of a lattice structure.

M. Park, et al., “Deformed Lattice Detection in Real-World Images Using Mean-Shift Belief Propagation”, IEEE Trans. On Pattern Analysis and Machine Intelligence, 2009

Structural approaches



Examples from two different structural texture analysis methods.

M. Park, et al., “Deformed Lattice Detection in Real-World Images Using Mean-Shift Belief Propagation”, IEEE Trans. On Pattern Analysis and Machine Intelligence, 2009

Structural approaches

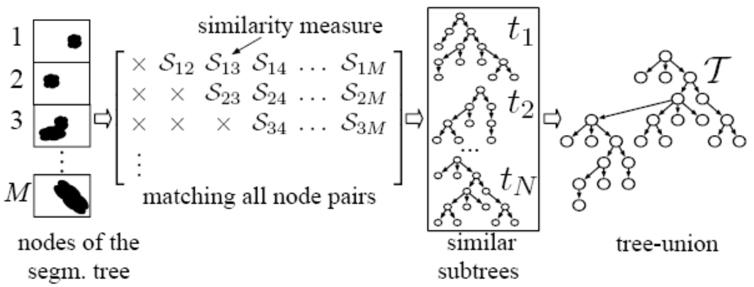
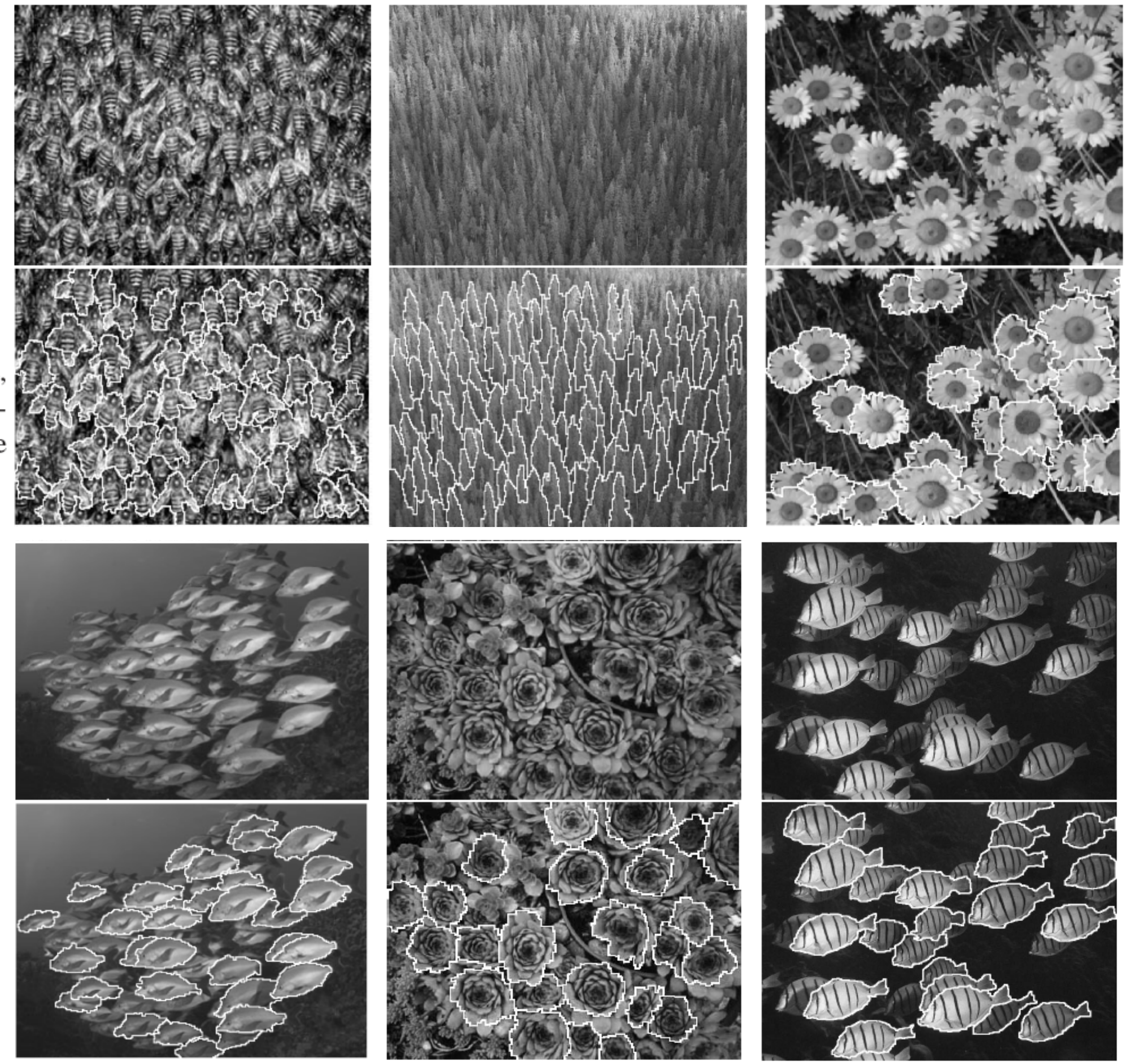


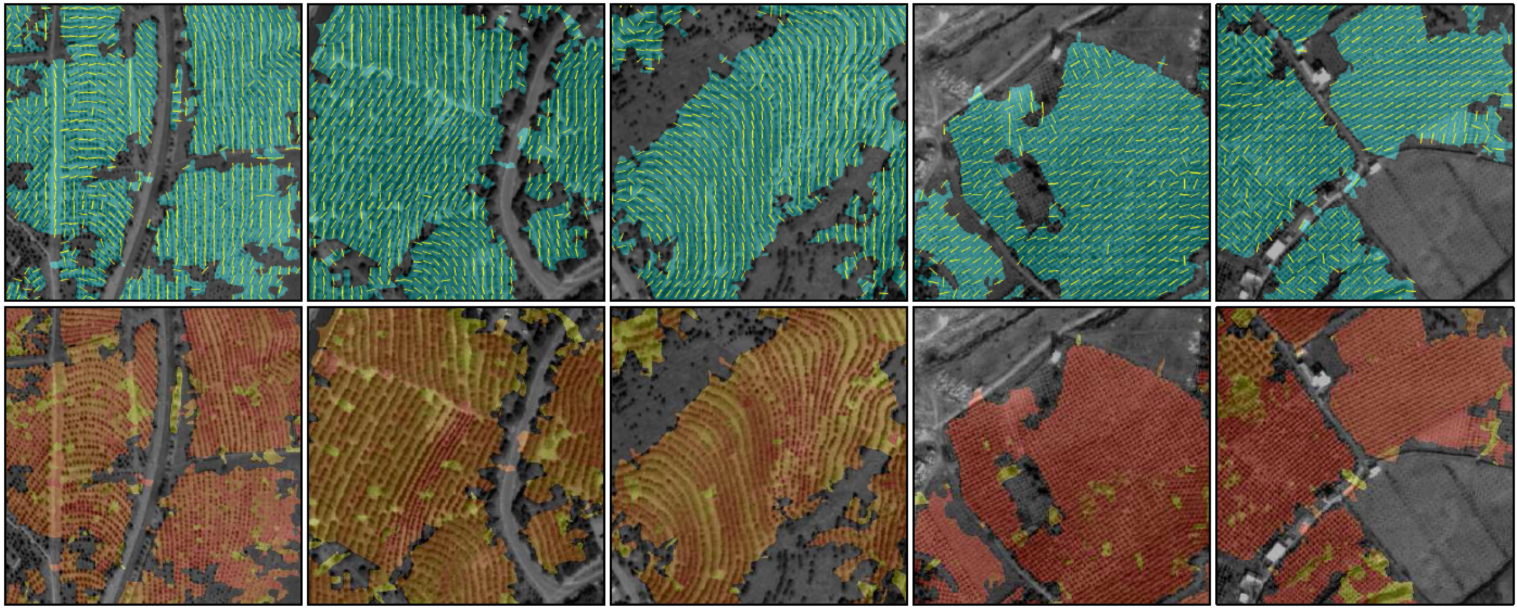
Figure 2. An input image is represented by the segmentation tree, and then all pairs of its M nodes are matched. Frequently occurring, similar subtrees are viewed as candidate texels, which are then fused into the tree-union, representing the texel model.



A method that involves forming a hierarchical representation of the image and searching for texels within this hierarchy.

N. Ahuja, S. Todorovic, "Extracting Texels in 2.1D Natural Textures", ICCV 2007

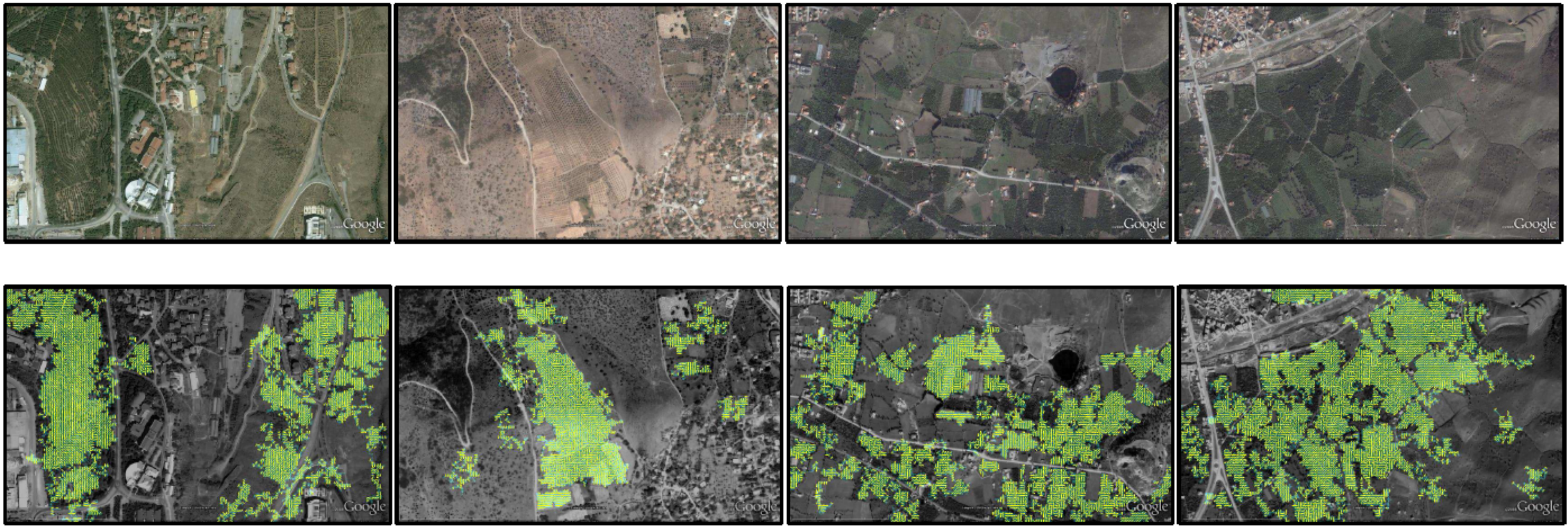
Structural approaches



A method for localization of natural structural textures using multi-orientation and multi-scale regularity analysis of textons detected using Laplacian of Gaussian filters (top: orientation estimates, bottom: scale estimates).

I. Z. Yalniz, S. Aksoy, "Unsupervised Detection and Localization of Structural Textures Using Projection Profiles", Pattern Recognition, 2010

Structural approaches



Examples of natural structural texture detection in images taken from Google Earth (top: input images, bottom: localized structural textures).

I. Z. Yalniz, S. Aksoy, "Unsupervised Detection and Localization of Structural Textures Using Projection Profiles", Pattern Recognition, 2010

Statistical approaches

- Usually, segmenting out the texels is difficult or even impossible in real images.
- Instead, numeric quantities or statistics that describe a texture can be computed from the gray tones or colors themselves.
- This approach can be less intuitive, but is computationally efficient and often works well.

Statistical approaches

- Some statistical approaches for texture:
  - Edge density and direction
  - Co-occurrence matrices
  - Local binary patterns
  - Statistical moments
  - Autocorrelation
  - Markov random fields
  - Autoregressive models
  - Mathematical morphology
  - Interest points
  - Fourier power spectrum
  - Gabor filters

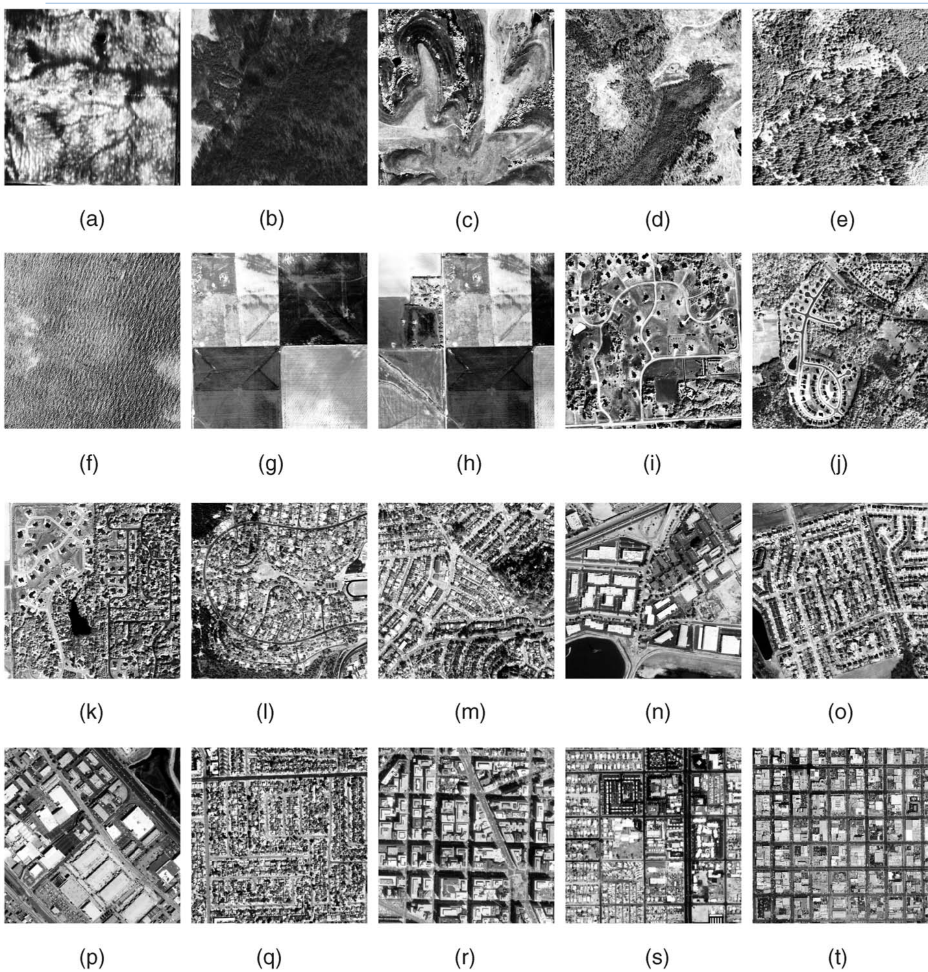
Edge density and direction

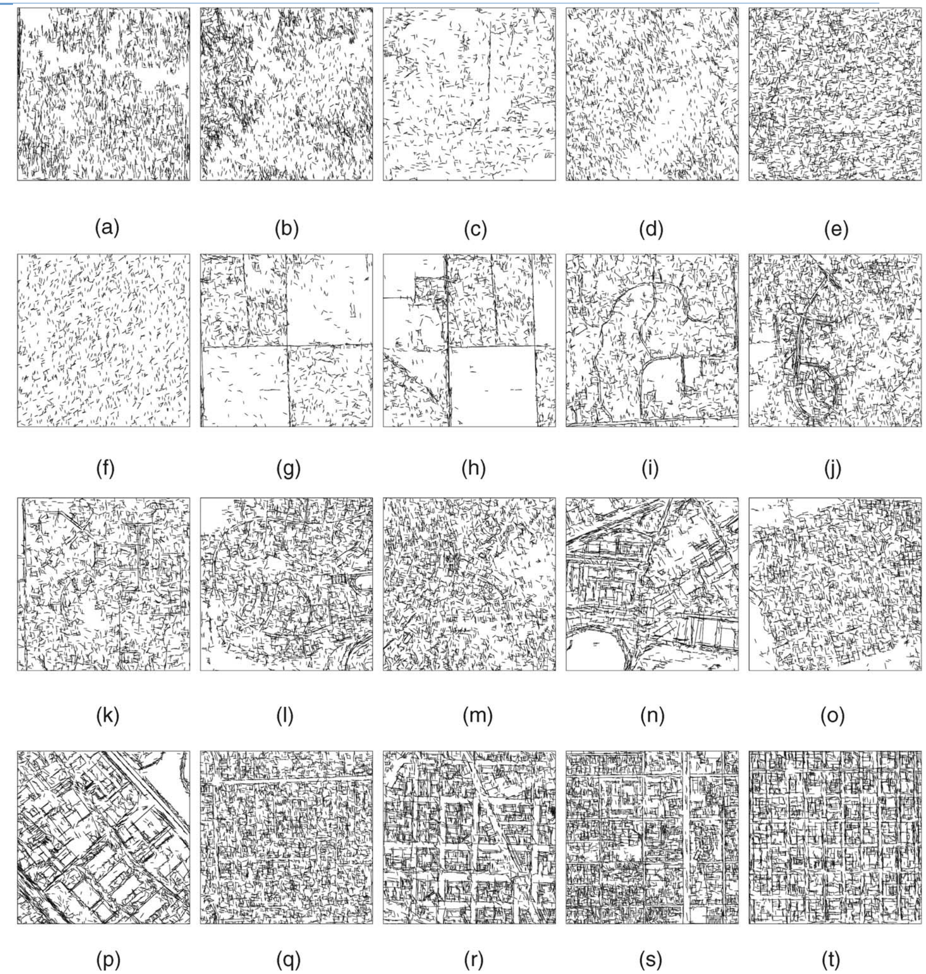
- The number of edge pixels in a fixed-size region tells us how busy that region is.
- Edge directions also help characterize the texture.
- Edge-based texture measures:
  - Edgeness per unit area

    $F_{edgeness} = | \{ p \mid \text{gradient\_magnitude}(p) \geq \text{threshold} \} | / \text{area}$
  - Edge magnitude and direction histograms

    $F_{magdir} = (H_{magnitude}, H_{direction})$
- Two histograms can be compared by computing their L_1 or L_2 distance.

Edge texture





Satellite images sorted according to the amount of land development (left). Properties of the arrangements of line segments can be used to model the organization in an area (right).

Co-occurrence matrices

- Co-occurrence, in general form, can be specified in a matrix of relative frequencies $P(i, j; d, \theta)$ with which two texture elements separated by distance d at orientation θ occur in the image, one with property i and the other with property j.
- In gray level co-occurrence, as a special case, texture elements are pixels and properties are gray levels.

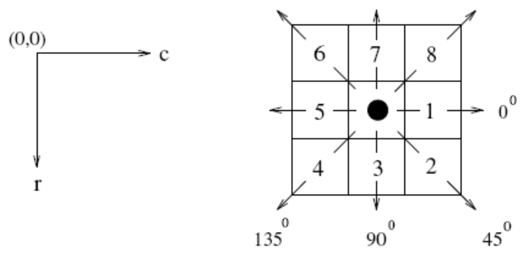
Co-occurrence matrices

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

(a) 4x4 image with gray levels 0-3.

		Gray Level 0	1	2	3
	0	#(0,0)	#(0,1)	#(0,2)	#(0,3)
Gray	1	#(1,0)	#(1,1)	#(1,2)	#(1,3)
Level	2	#(2,0)	#(2,1)	#(2,2)	#(2,3)
	3	#(3,0)	#(3,1)	#(3,2)	#(3,3)

(b) General form of co-occurrence matrices $P(i,j;d,\theta)$ for gray levels 0-3 where $\#(i,j)$ stands for number of times gray levels i and j have been neighbors.



$$P(i,j;1,0^\circ) = \begin{pmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

(c) $(d,\theta) = (1,0^\circ)$

$$P(i,j;1,45^\circ) = \begin{pmatrix} 2 & 1 & 3 & 0 \\ 1 & 2 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{pmatrix}$$

(d) $(d,\theta) = (1,45^\circ)$

$$P(i,j;1,90^\circ) = \begin{pmatrix} 6 & 0 & 2 & 0 \\ 0 & 4 & 2 & 0 \\ 2 & 2 & 2 & 2 \\ 0 & 0 & 2 & 0 \end{pmatrix}$$

(e) $(d,\theta) = (1,90^\circ)$

$$P(i,j;1,135^\circ) = \begin{pmatrix} 4 & 1 & 0 & 0 \\ 1 & 2 & 2 & 0 \\ 0 & 2 & 4 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

(f) $(d,\theta) = (1,135^\circ)$

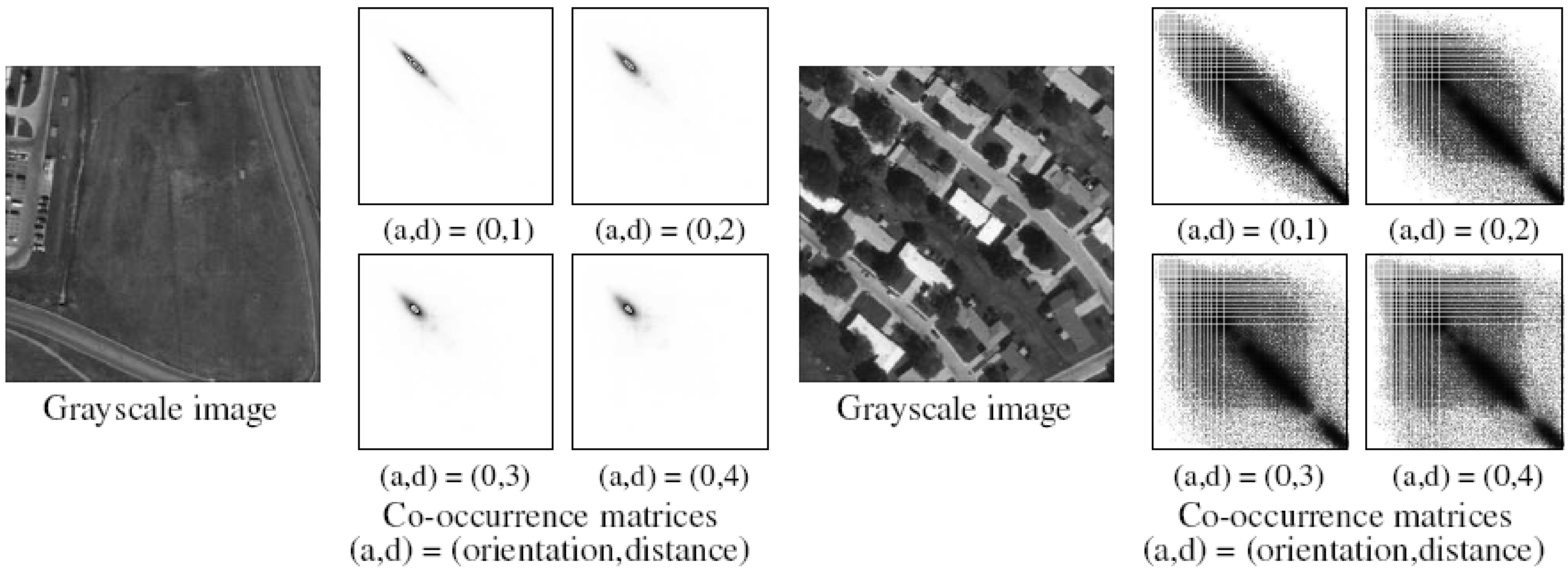
Co-occurrence matrices

- If a texture is coarse and the distance d used to compute the co-occurrence matrix is small compared to the sizes of the texture elements, pairs of pixels at separation d should usually have similar gray levels.
- This means that high values in the matrix $P(i, j; d, \theta)$ should be concentrated on or near its main diagonal.
- Conversely, for a fine texture, if d is comparable to the texture element size, then the gray levels of points separated by d should often be quite different, so that values in $P(i, j; d, \theta)$ should be spread out relatively uniformly.

Co-occurrence matrices

- Similarly, if a texture is directional, i.e., coarser in one direction than another, the degree of spread of the values about the main diagonal in $P(i, j; d, \theta)$ should vary with the orientation θ.
- Thus texture directionality can be analyzed by comparing spread measures of $P(i, j; d, \theta)$ for various orientations.

Co-occurrence matrices



(a) Co-occurrence matrices for an image with a small amount of local spatial variations.

(b) Co-occurrence matrices for an image with a large amount of local spatial variations.

Figure 4. Example co-occurrence matrices.

Co-occurrence matrices

- In order to use the information contained in co-occurrence matrices, Haralick et al. (SMC 1973) defined 14 statistical features that capture textural characteristics such as homogeneity, contrast, organized structure, and complexity.

Co-occurrence matrices

$$Energy = \sum_i \sum_j N_d^2(i,j)$$

$$Entropy = -\sum_i \sum_j N_d(i,j) log_2 N_d(i,j)$$

$$Contrast = \sum_i \sum_j (i-j)^2 N_d(i,j)$$

$$Homogeneity = \sum_i \sum_j \frac{N_d(i,j)}{1+ \mid i-j \mid}$$

$$Correlation = \frac{\sum_i \sum_j (i-\mu_i)(j-\mu_j) N_d(i,j)}{\sigma_i \sigma_j}$$

where μ_i, μ_j are the means and σ_i, σ_j are the standard deviations of the row and column sums $N_d(i)$ and $N_d(j)$ defined by

$$N_d(i) = \sum_j N_d(i,j)$$

$$N_d(j) = \sum_i N_d(i,j)$$

Co-occurrence matrices

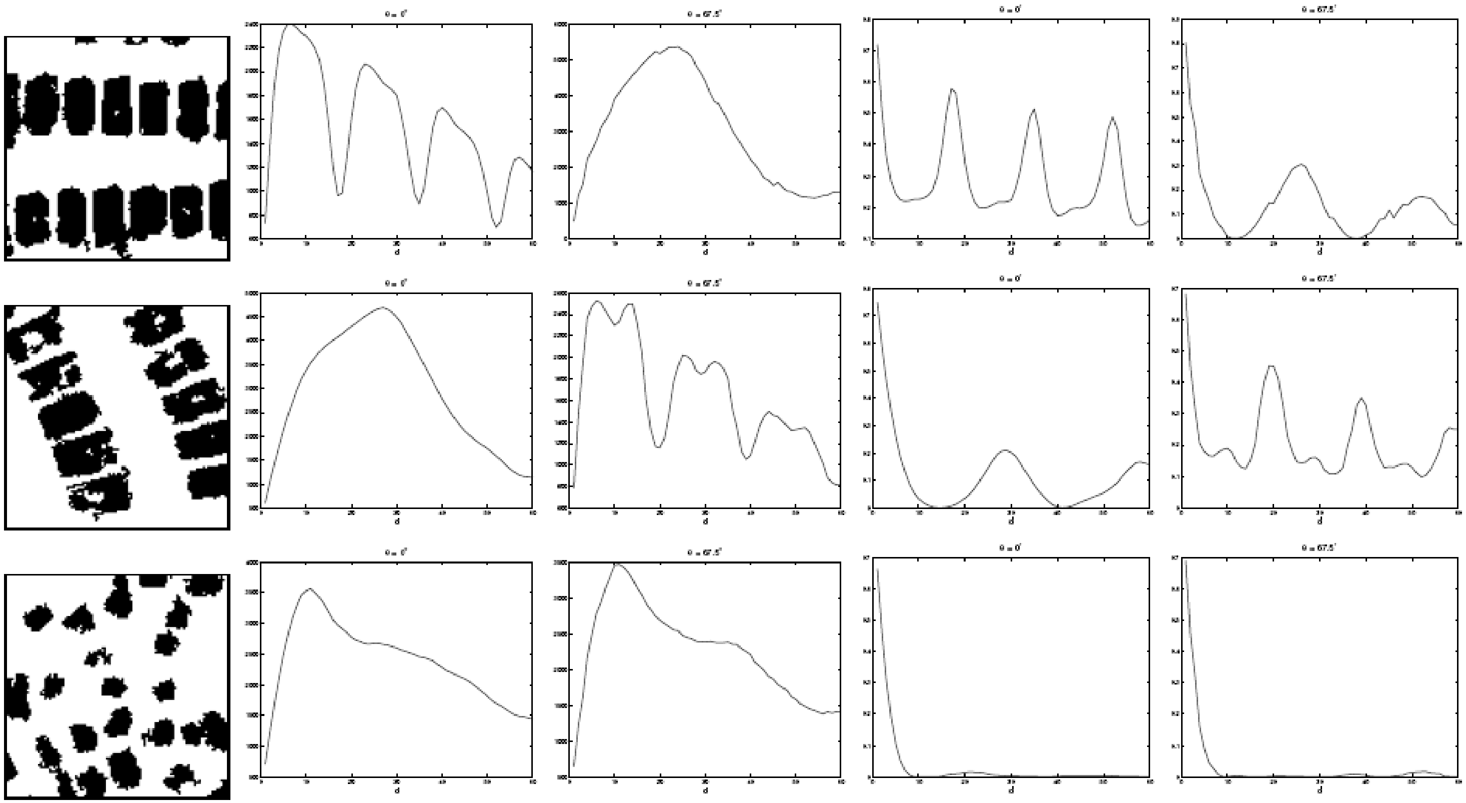
- Zucker and Terzopoulos (CGIP 1980) suggested using a chi-square statistical test to select the values of d that have the most structure for a given class of images.

$$\chi^2(d) = \left(\sum_i \sum_j \frac{N_d^2(i,j)}{N_d(i)N_d(j)} - 1\right)$$

  $N_d(i,j)$: unnormalized co-occurrence of gray level i and j for distance d.

- As N gets closer to a diagonal matrix, the test gives larger values.

Co-occurrence matrices



Example building groups (first column), the contrast features for 0 and 67.5 degree orientations (second and third columns), and the chi-square features for 0 and 67.5 degree orientations (fourth and fifth columns). X-axes represent inter-pixel distances of 1 to 60. The features at a particular orientation exhibit a periodic structure as a function of distance if the neighborhood contains a regular arrangement of buildings along that direction. On the other hand, features are very similar for different orientations if there is no particular arrangement in the neighborhood.

Local binary patterns

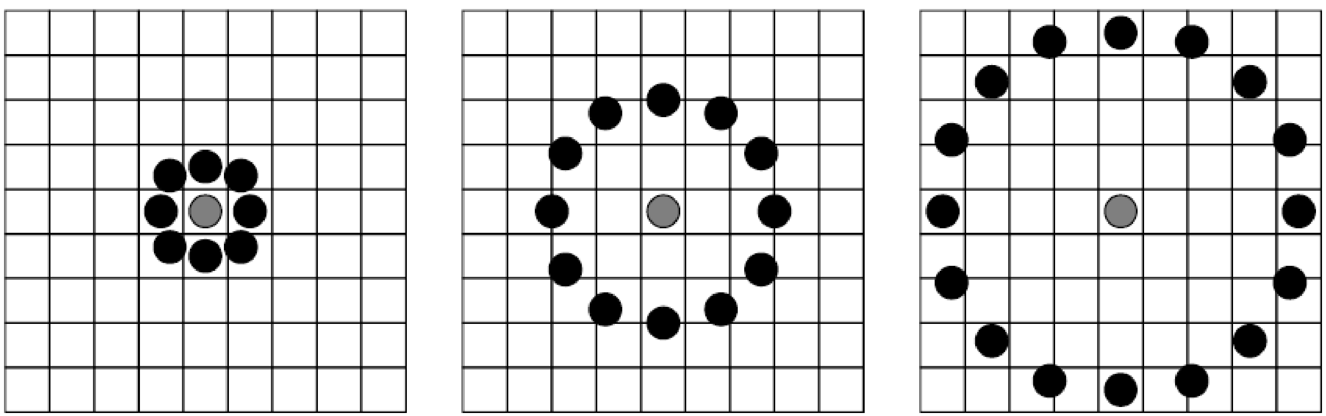
- For each pixel p, create an 8-bit number $b_1\ b_2\ b_3\ b_4\ b_5\ b_6\ b_7\ b_8$, where $b_i = 0$ if neighbor i has value less than or equal to p's value and 1 otherwise.
- Represent the texture in the image (or a region) by the histogram of these numbers.

	1	2	3	
	100	101	103	
8	40	50	80	4
	50	60	90	
	7	6	5	

→ 1 1 1 1 1 1 0 0

Local binary patterns

- The fixed neighborhoods were later extended to multi-scale circularly symmetric neighbor sets.



$P=8, R=1.0$ $P=12, R=2.5$ $P=16, R=4.0$

Autocorrelation

- The autocorrelation function of an image can be used to
  - detect repetitive patterns of texture elements, and
  - describe the fineness/coarseness of the texture.
- The autocorrelation function ρ(dr,dc) for displacement d=(dr,dc) is given by

$$\rho(dr,dc) = \frac{\sum_{r=0}^{N}\sum_{c=0}^{N} I[r,c]I[r+dr,c+dc]}{\sum_{r=0}^{N}\sum_{c=0}^{N} I^2[r,c]}$$

$$= \frac{I[r,c] \circ I_d[r,c]}{I[r,c] \circ I[r,c]}$$

Autocorrelation

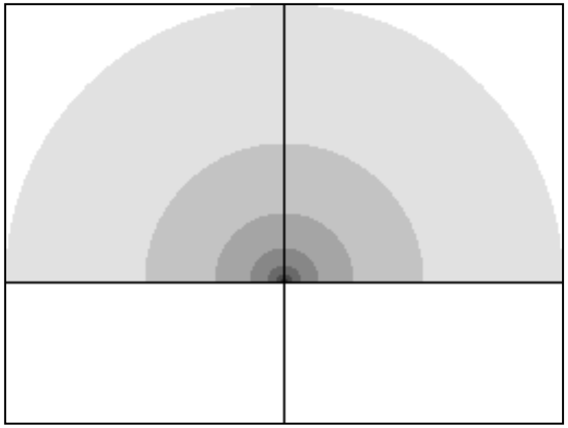
- Interpreting autocorrelation:
  - Coarse texture → function drops off slowly
  - Fine texture → function drops off rapidly
  - Can drop differently for r and c
  - Regular textures → function will have peaks and valleys; peaks can repeat far away from [0,0]
  - Random textures → only peak at [0,0]; breadth of peak gives the size of the texture

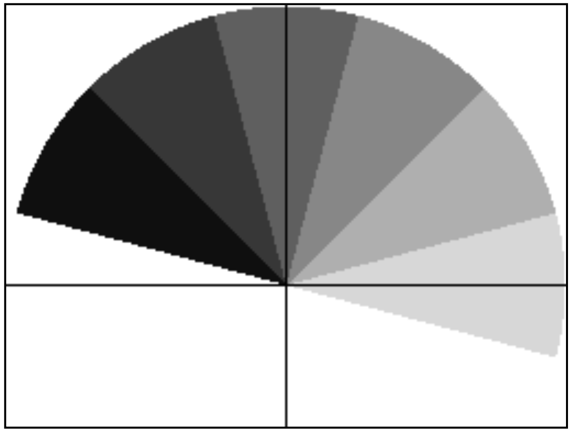
Fourier power spectrum

- The autocorrelation function is related to the power spectrum of the Fourier transform.
- The power spectrum contains texture information because
  - prominent peaks in the spectrum give the principal direction of the texture patterns,
  - location of the peaks gives the fundamental spatial period of the patterns.

Fourier power spectrum

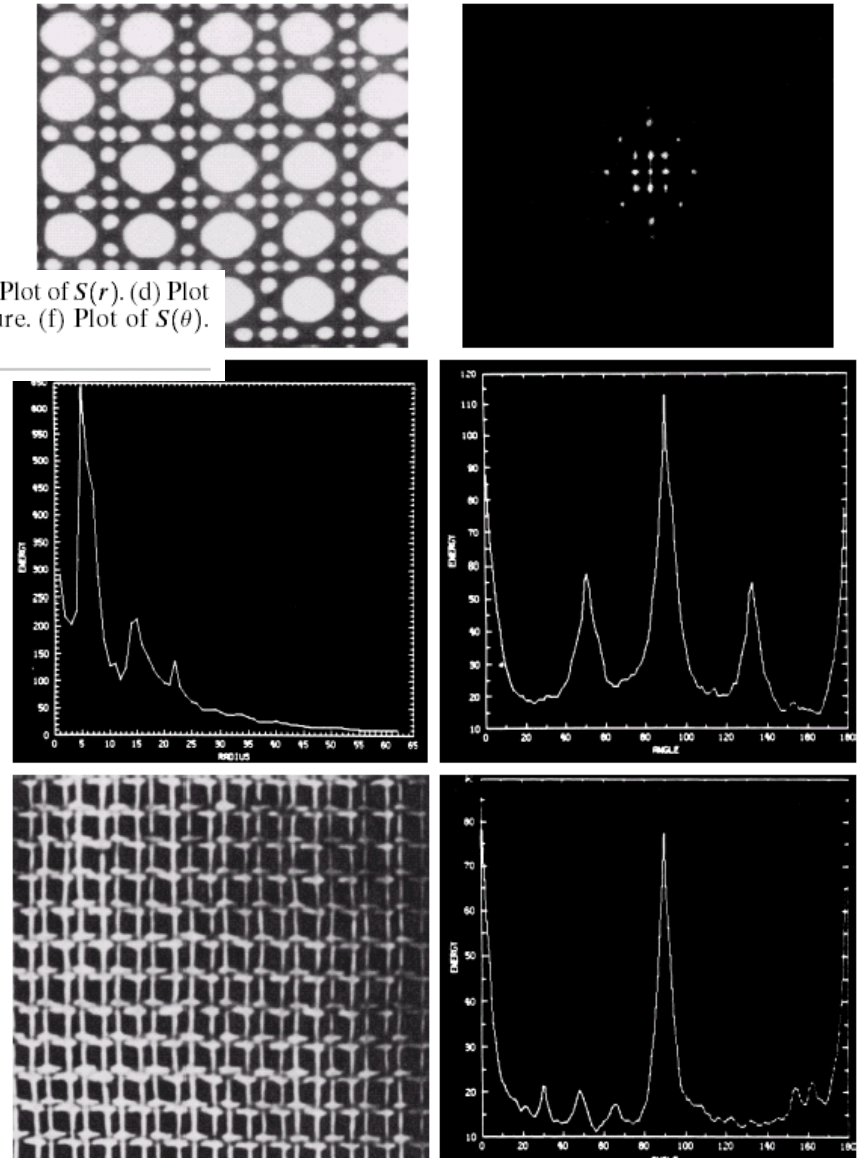
- The power spectrum, represented in polar coordinates, can be integrated over regions bounded by circular rings (for frequency content) and wedges (for orientation content).





$$x_i = \sum_{r=r_i}^{r_{i+1}} \sum_{\theta=0}^{\pi} S(r,\theta) \qquad y_i = \sum_{\theta=\theta_i}^{\theta_{i+1}} \sum_{r=1}^{r_{\max}} S(r,\theta)$$

Fourier power spectrum



a b
c d
e f

FIGURE 11.24 (a) Image showing periodic texture. (b) Spectrum. (c) Plot of $S(r)$. (d) Plot of $S(\theta)$. (e) Another image with a different type of periodic texture. (f) Plot of $S(\theta)$. (Courtesy of Dr. Dragana Brzakovic, University of Tennessee.)

Gabor filters

- Gabor filters can be considered as orientation and scale tunable edge and line detectors.
- A 2D Gabor function g(x,y) and its Fourier transform G(u,v) can be written as

$$g(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)+2\pi jWx\right]$$

$$G(u,v) = \exp\left\{-\frac{1}{2}\left[\frac{(u-W)^2}{\sigma_u^2}+\frac{v^2}{\sigma_v^2}\right]\right\}$$

where $\sigma_u = 1/2\pi\sigma_x$ and $\sigma_v = 1/2\pi\sigma_y$.

Gabor filters

- Let U_l and U_h denote the lower and upper center frequencies of interest, K be the number of orientations, and S be the number of scales, the filter parameters can be selected as

$$a = (U_h/U_l)^{\frac{1}{S-1}}, \quad \sigma_u = \frac{(a-1)U_h}{(a+1)\sqrt{2\ln 2}},$$

$$\sigma_v = \tan\left(\frac{\pi}{2k}\right)\left[U_h - 2\ln\left(\frac{2\sigma_u^2}{U_h}\right)\right]\left[2\ln 2 - \frac{(2\ln 2)^2\sigma_u^2}{U_h^2}\right]^{-\frac{1}{2}}$$

where $W = U_h$ and $m = 0, 1, \ldots, S-1$.

Gabor filters

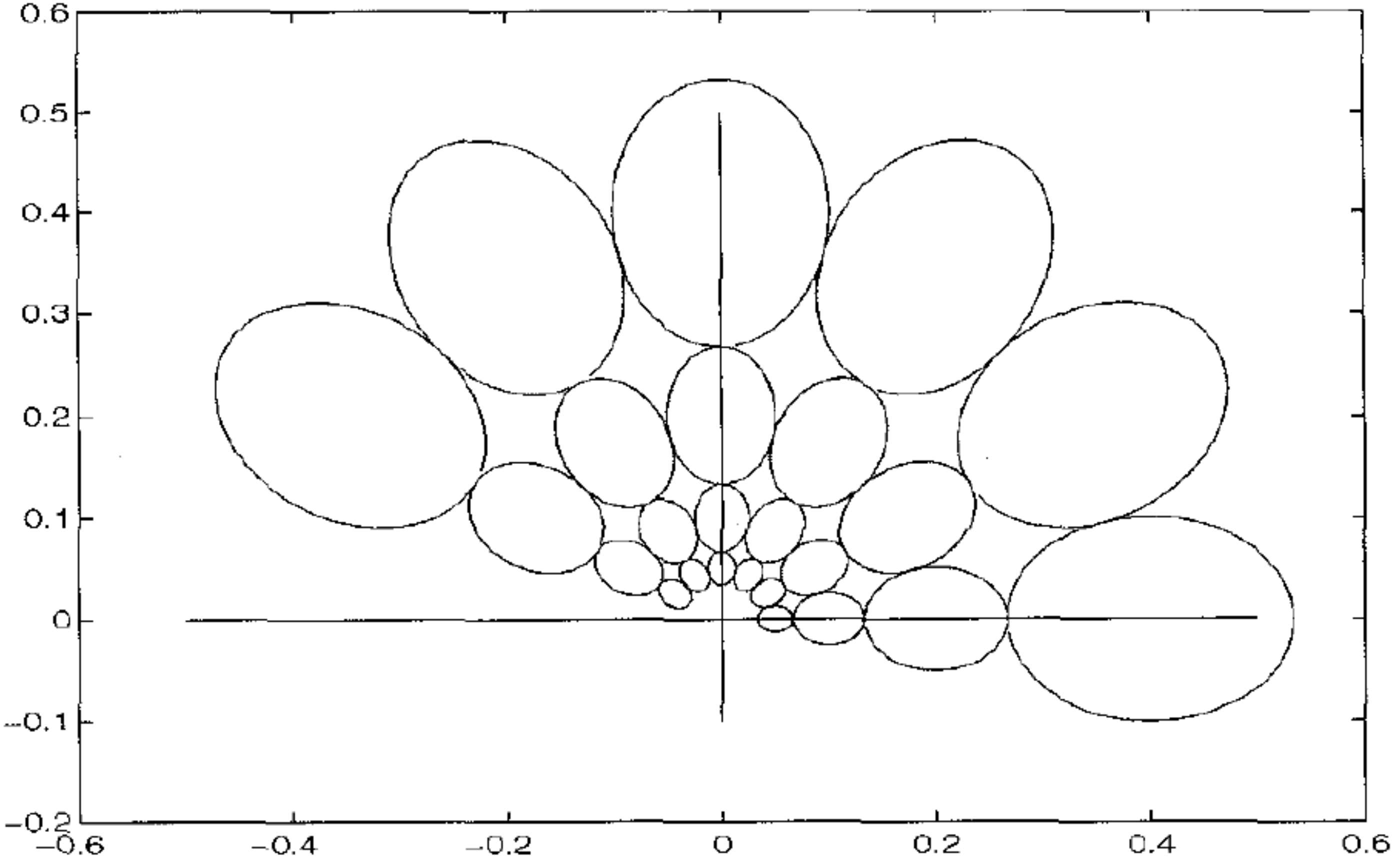
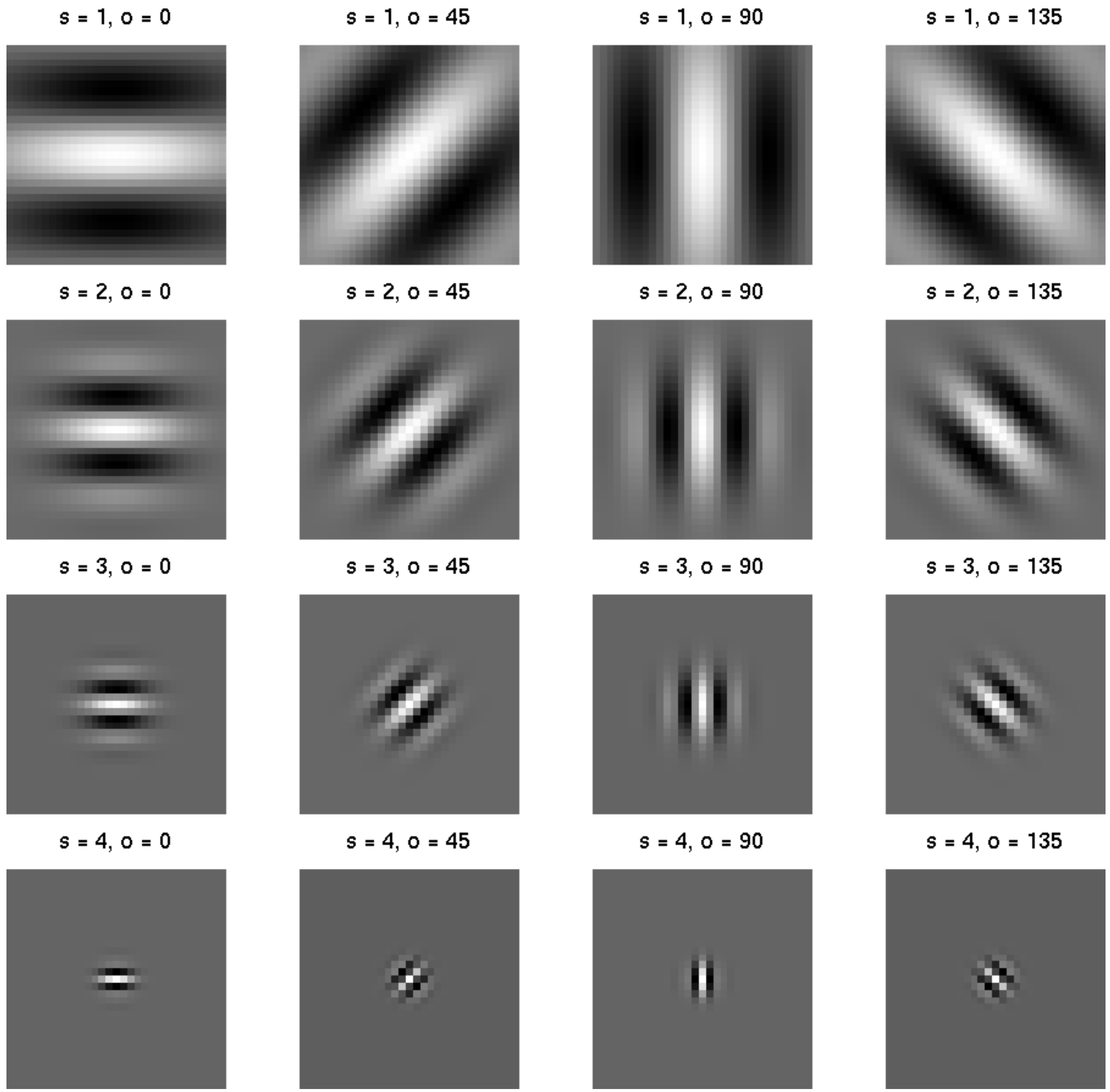


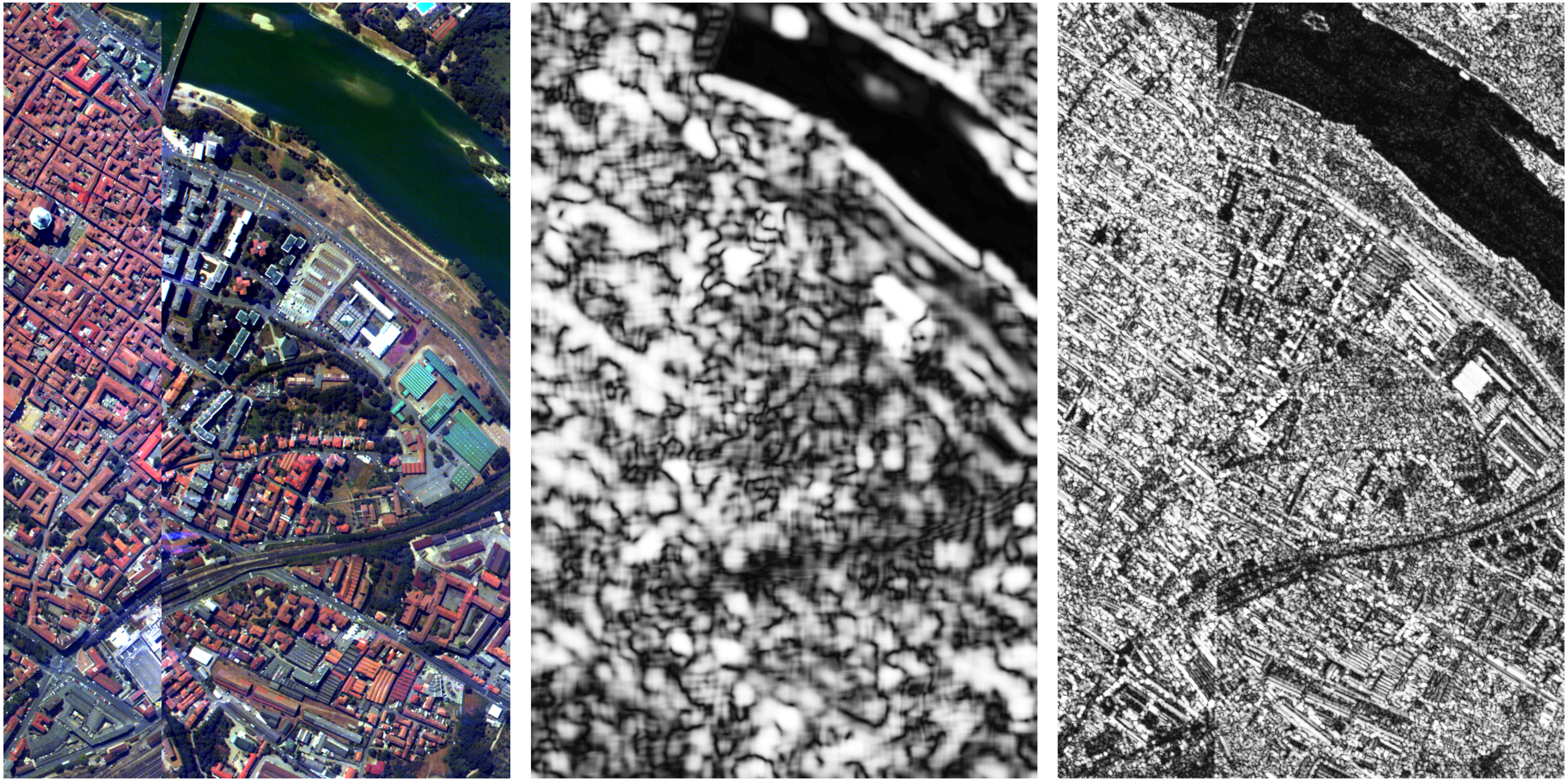
Fig. 1. The contours indicate the half-peak magnitude of the filter responses in the Gabor filter dictionary. The filter parameters used are $U_h = 0.4$, $U_l = 0.05$, $K = 6$, and $S = 4$.

Gabor filters



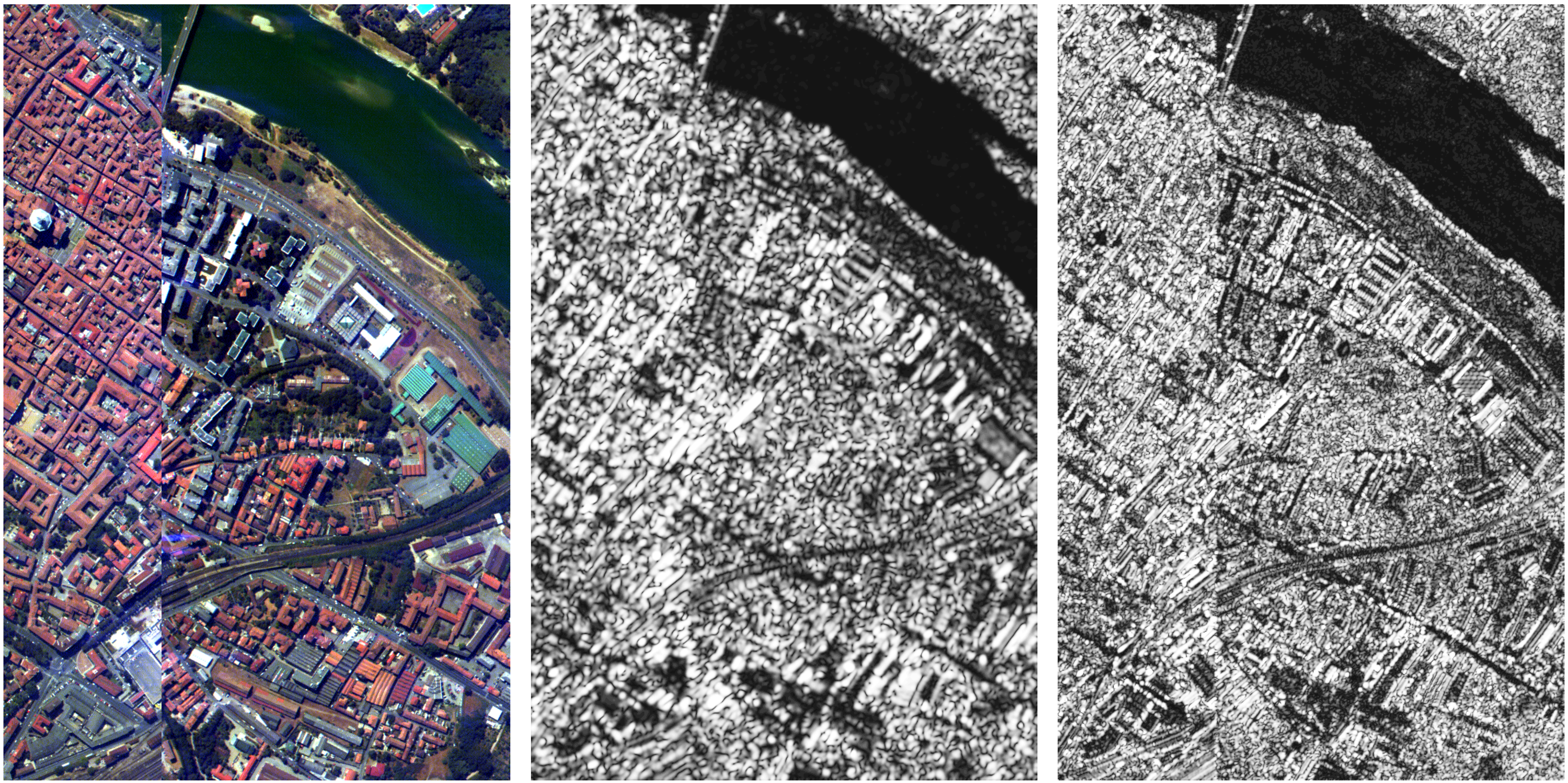
Filters at multiple scales and orientations.

Gabor filters

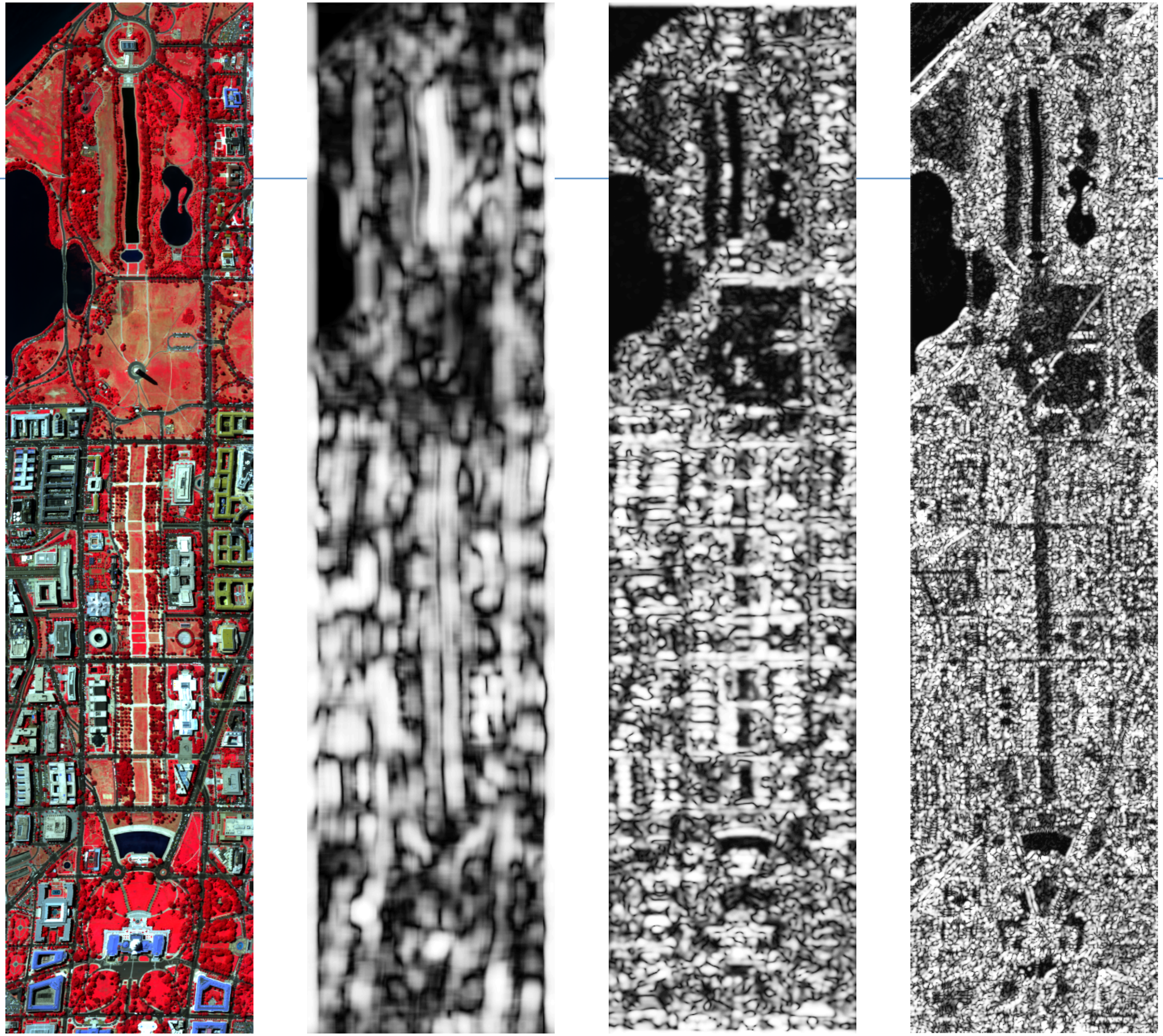


Gabor filter responses for a satellite image.

Gabor filters



Gabor filter responses for a satellite image.



Gabor filter responses for a satellite image.

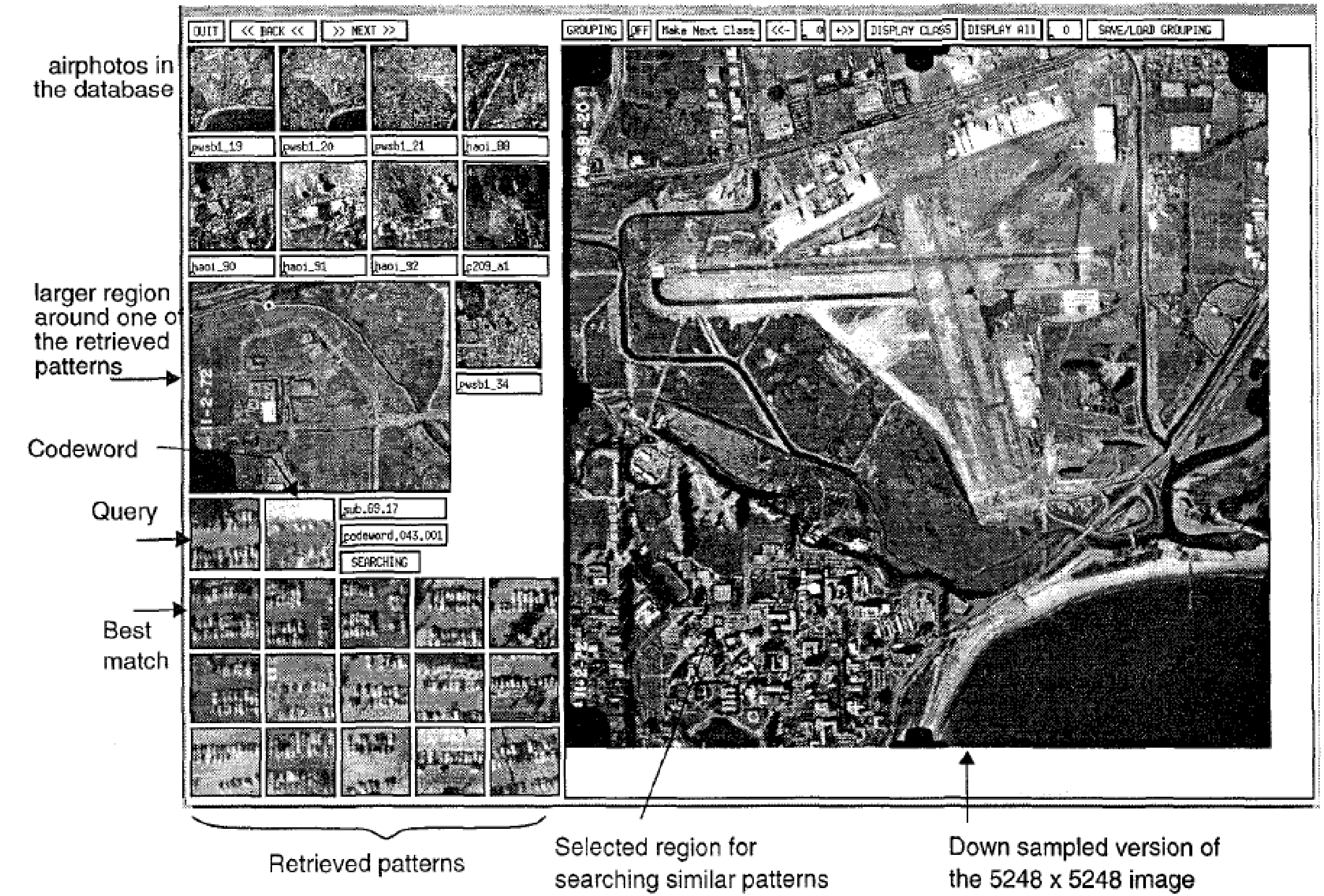
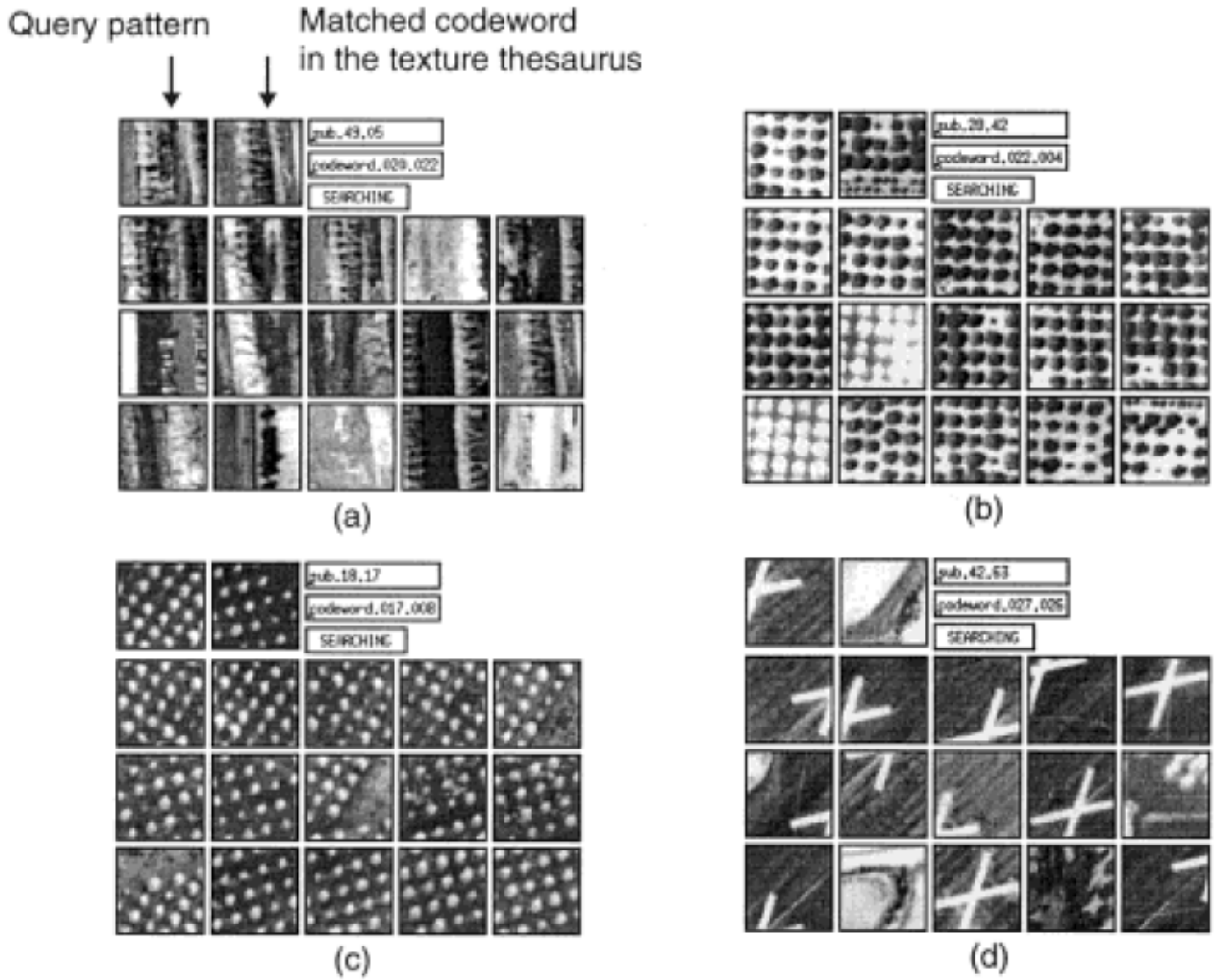


Figure 4: Snapshot of an aerial photograph browsing demonstration. The example shown indicates a query pattern containing a parking lot. Next to the query is the image codeword used to index the database. The browser can retrieve almost 99% of all the parking lots in the aerial photo database.

Gabor filters



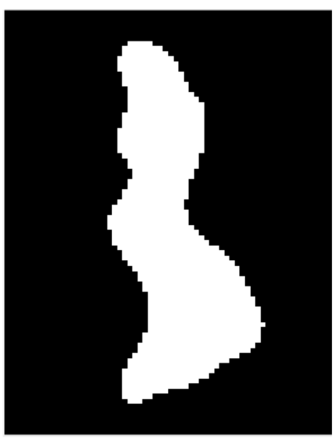
How well the kernel machines can model/learn shapes?

SVM:
(libSVM)

83 x 64 = **5312 pixels**
(the training data size)

Use pixel coordinates in 2D as input:
$X=[(x_1,y_1)\dots(x_{5312},y_{5312})]$
Y={black, white}

T = 0.05
#SVs = **1082 (~20%)**
PixelError = 0



σ = 100
#SVs = **1036 (~20%)**
PixelError = 309

σ = 20
#SVs = **1084 (~20%)**
PixelError = 119

σ = 3.3
#SVs = **3261 (~61%)**
PixelError = 51

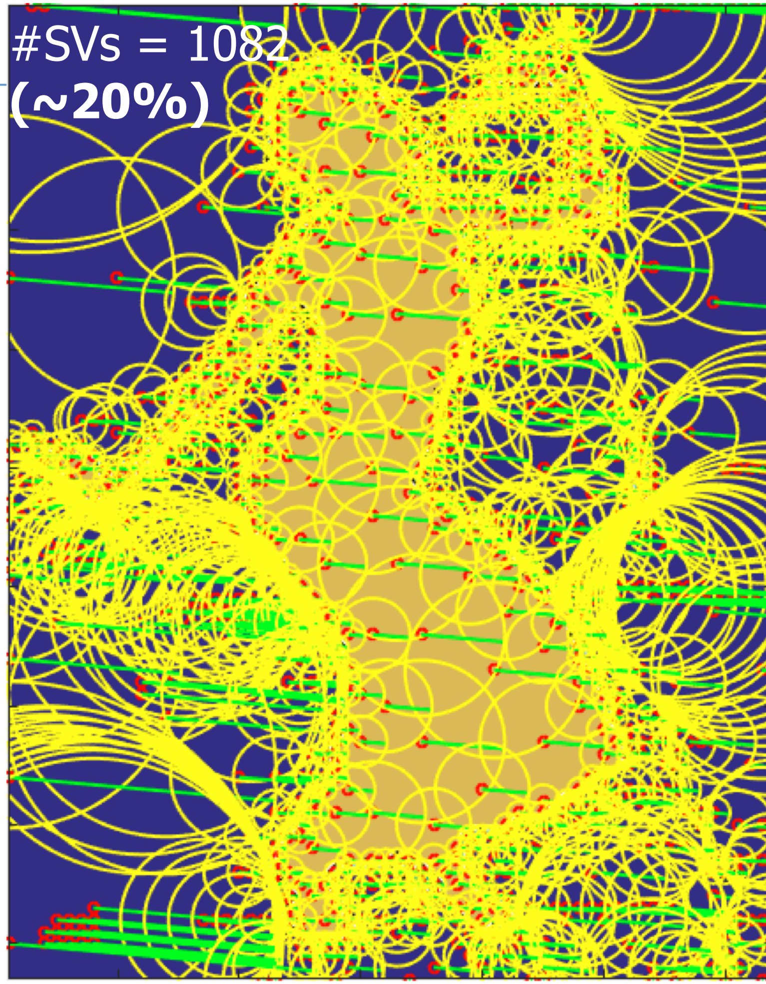
σ = 1.4
#SVs = **4778 (~90%)**
PixelError = 31

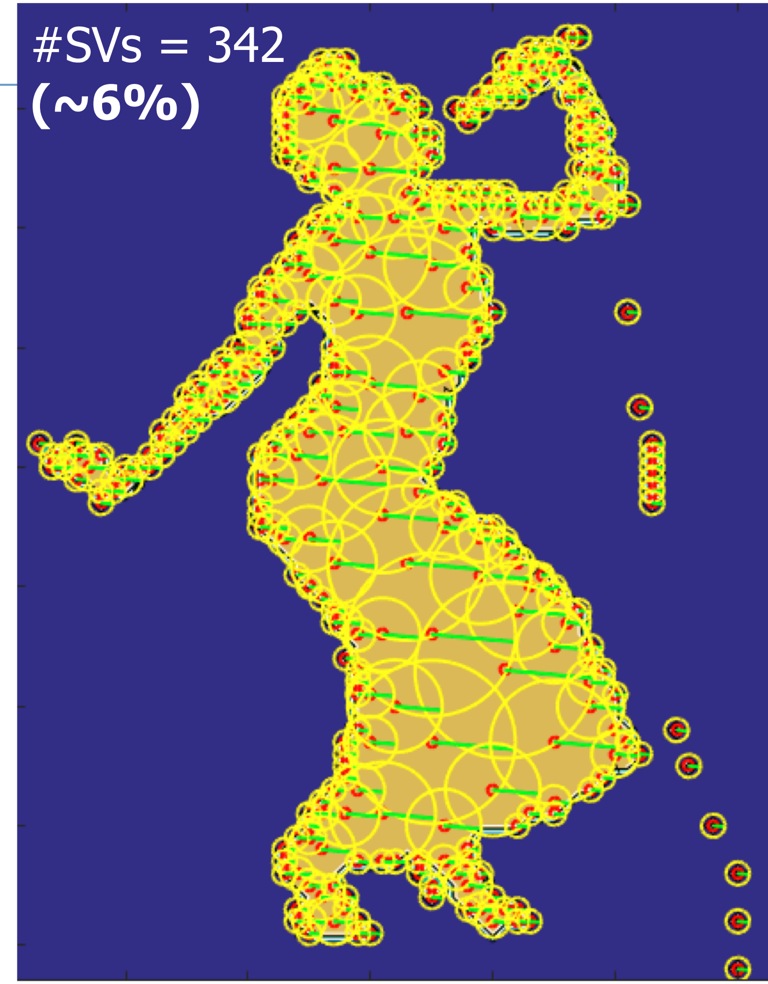
σ = 1.0
#SVs = **4974 (~94%)**
PixelError = 17

Visualization of the kernel parameters

T = 0.05
#SVs = **1082**
PixelError = 0

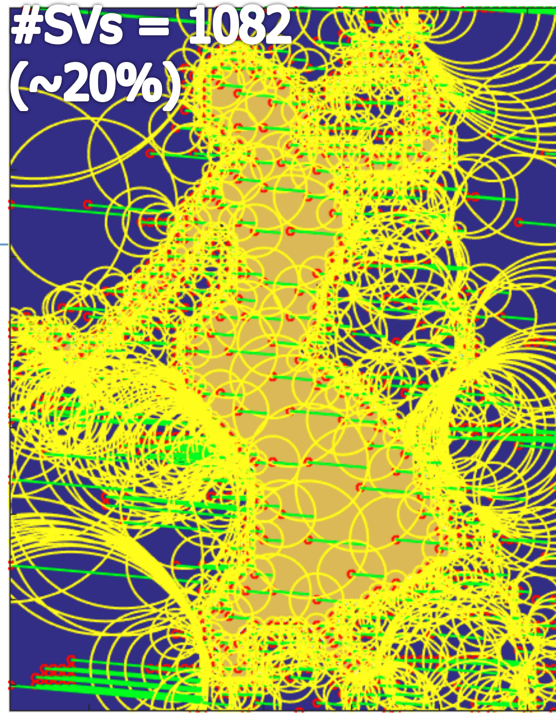
83 x 64 =
5312 pixels
(the training data size)

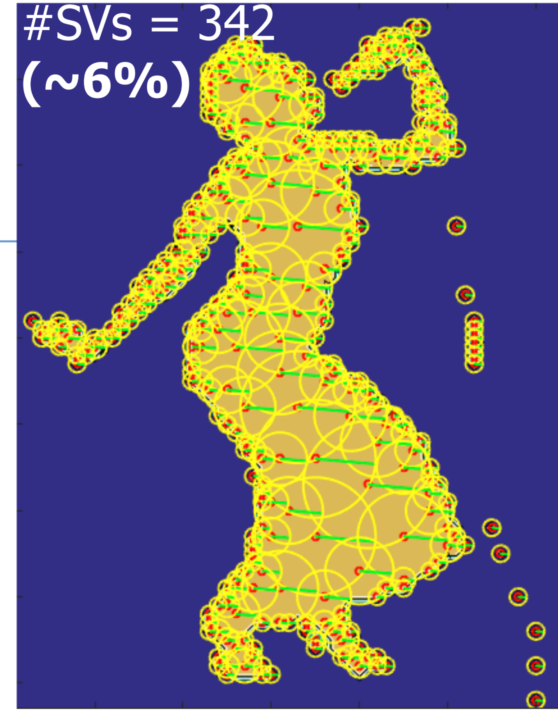




T value:	0.05	0.1	0.15	0.2	0.3	0.35	0.4
total SVs:	1082	917	898	850	791	730	728
total pixel error:	0	0	0	2	10	9	14

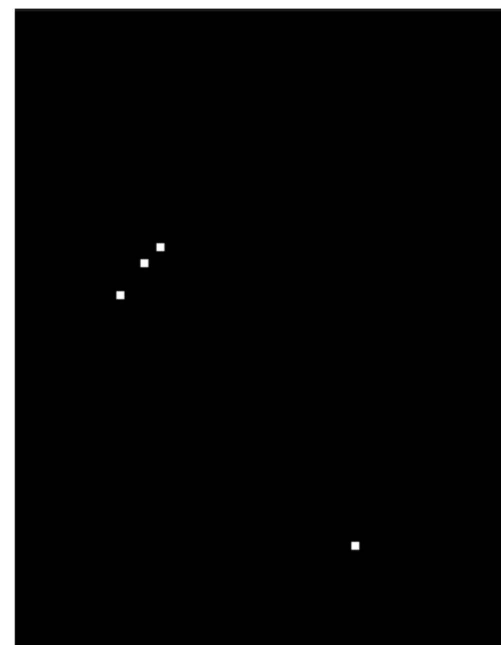
One-Class Approximation





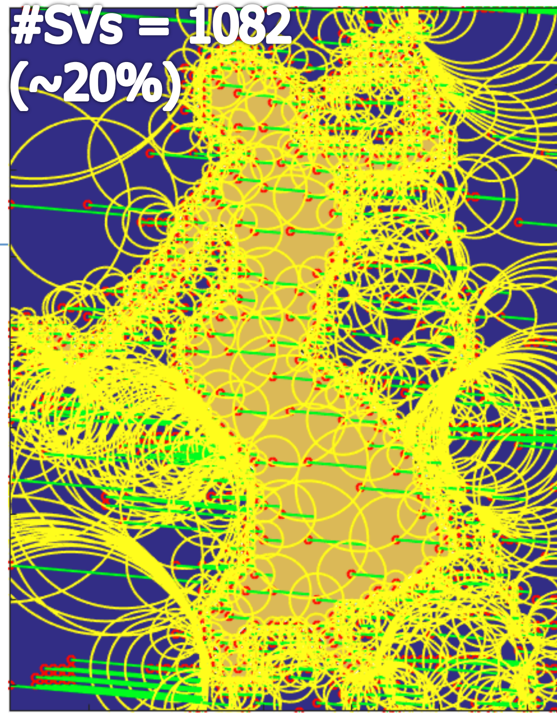
Original Image

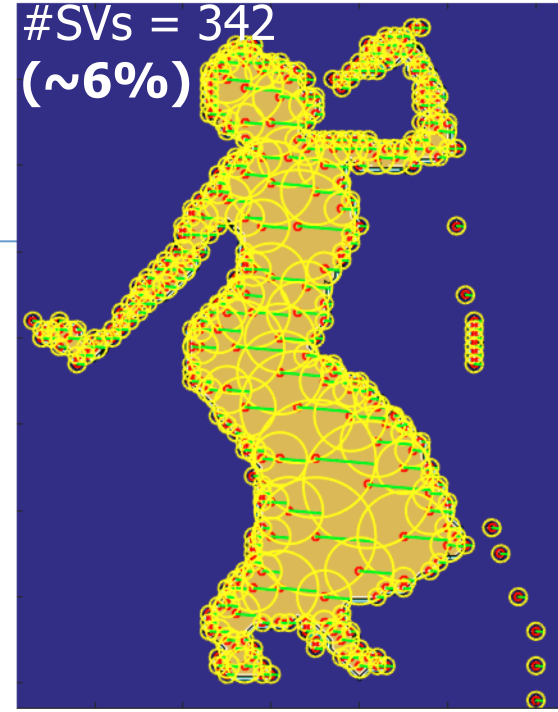
Approximation



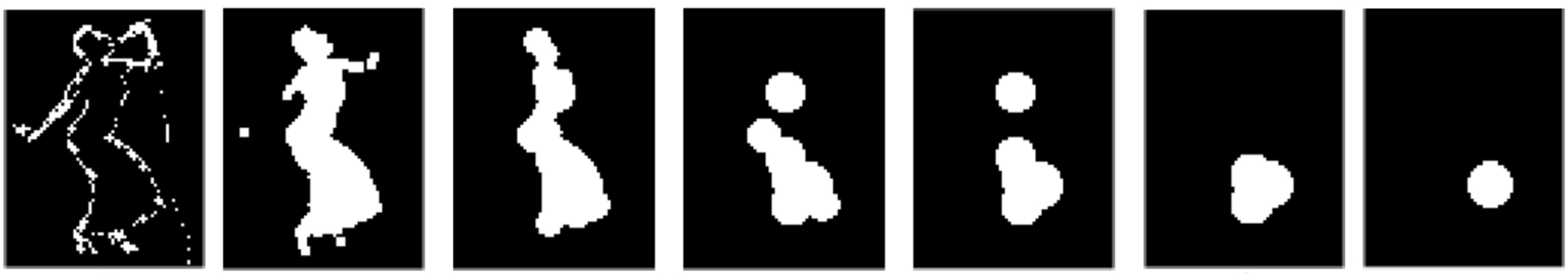
Error

Data Analysis





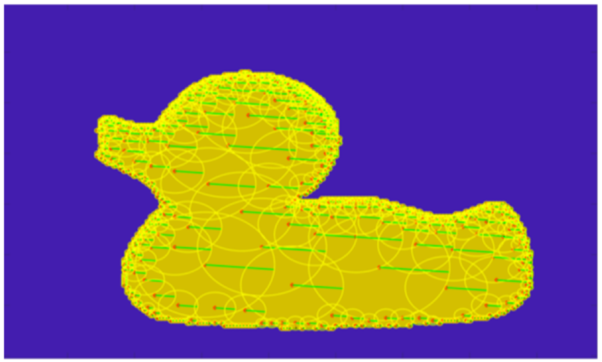
Bin Center:	1.29	3.19	5.09	6.99	8.90	10.80	12.70	14.60	16.51	18.41
Total Counts:	303	16	12	2	3	1	0	2	2	1



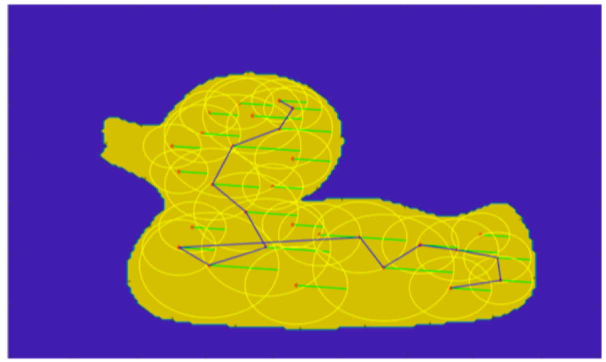
(a) (b) (c) (d) (e) (f) (g)

$\sigma_{i2} < 1.29$ $\sigma_{i2} > 1.29$ $\sigma_{i2} > 5.09$ $\sigma_{i2} > 8.90$ $\sigma_{i2} > 14.60$ $\sigma_{i2} > 16.51$ $\sigma_{i2} > 18.41$

Computing Skeleton with SDs

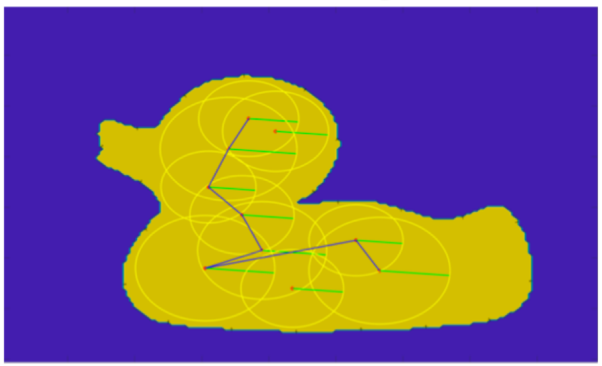


(a) All of the foreground r_i

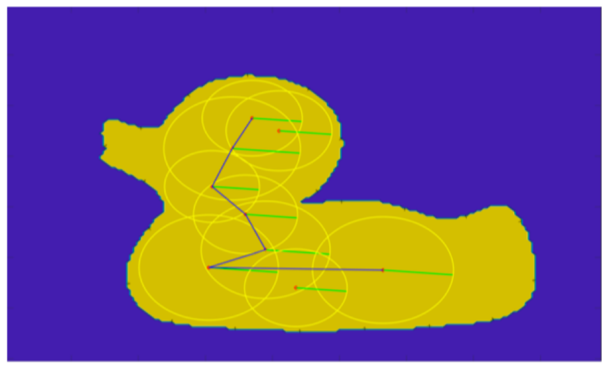


(b) Skeleton for $\sigma_i^2 > 29.12$

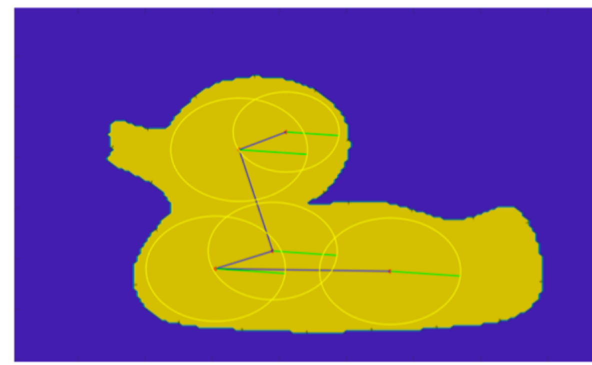
(c) Skeleton for $\sigma_i^2 > 48.32$



(d) Skeleton for $\sigma_i^2 > 67.51$



(e) Skeleton for $\sigma_i^2 > 86.71$



(f) Skeleton for $\sigma_i^2 > 105.90$

Use only SDs to compute skeletons

- Save computational time