Feature Extraction

Representing Shape of a Region

Some Basic Shape Descriptors

Some Basic Shape Descriptors

p: perimeter
A: area

$$\text{compactness} = \frac{p^2}{A}$$

a dimensionless measure →
4π for a circle (its minimum value) and 16 for a square

$$\text{circularity} = \frac{4\pi A}{p^2}$$

→ 1 for a perfect circle (its maximum value) and π/4 for a square

Some Basic Shape Descriptors

eigenvectors of the covariance matrix, $\mathbf{C}$, of the data

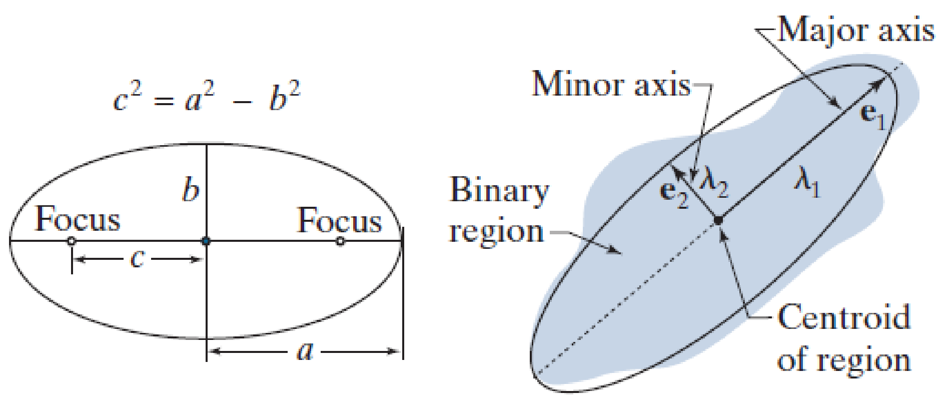
$$\mathbf{C} = \frac{1}{K-1}\sum_{k=1}^{K}(\mathbf{z}_k - \overline{\mathbf{z}})(\mathbf{z}_k - \overline{\mathbf{z}})^T$$

$\mathbf{z}_k$ is a 2-D vector whose elements are the two spatial coordinates of a point in the region, K is the total number of points, and $\overline{\mathbf{z}}$ is the mean vector:

$$\overline{\mathbf{z}} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{z}_k$$

a b

FIGURE 11.21
(a) An ellipse in standard form.
(b) An ellipse approximating a region in arbitrary orientation.



$\mathbf{e}_1$ λ_1 and $\mathbf{e}_2$ λ_2 are the eigenvectors and corresponding eigenvalues of the covariance matrix of the coordinates of the region

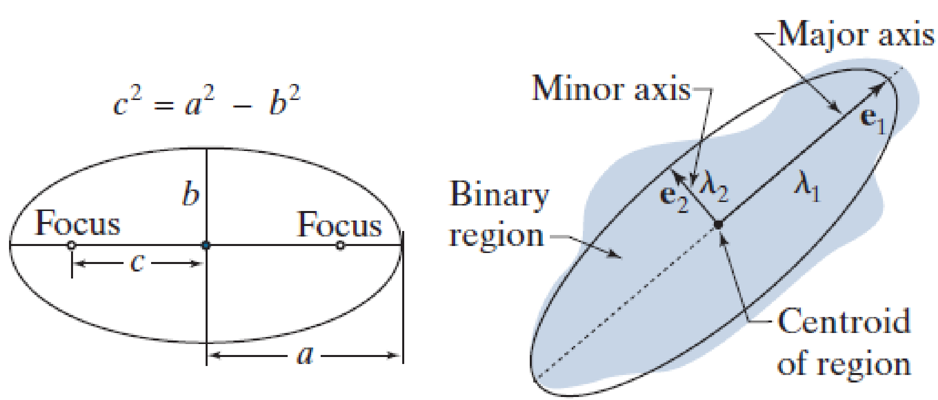
Some Basic Shape Descriptors

$$\text{eccentricity} = \frac{c}{a} = \frac{\sqrt{a^2 - b^2}}{a} = \sqrt{1 - (b/a)^2} \qquad a \geq b$$

eccentricity is 0 for a circle (a = b) and 1 for a line (a = 1, b = 0).

a b

FIGURE 11.21
(a) An ellipse in standard form.
(b) An ellipse approximating a region in arbitrary orientation.

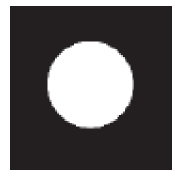


$\mathbf{e}_1$ λ_1 and $\mathbf{e}_2$ λ_2 are the eigenvectors and corresponding eigenvalues of the covariance matrix of the coordinates of the region

Some Basic Shape Descriptors

a b c d

FIGURE 11.22 Compactness, circularity, and eccentricity of some simple binary regions.

Descriptor				
Compactness	10.1701	42.2442	15.9836	13.2308
Circularity	1.2356	0.2975	0.7862	0.9478
Eccentricity	0.0411	0.0636	0	0.8117

Representing Shape of a Region

Shape Context

Shape Context

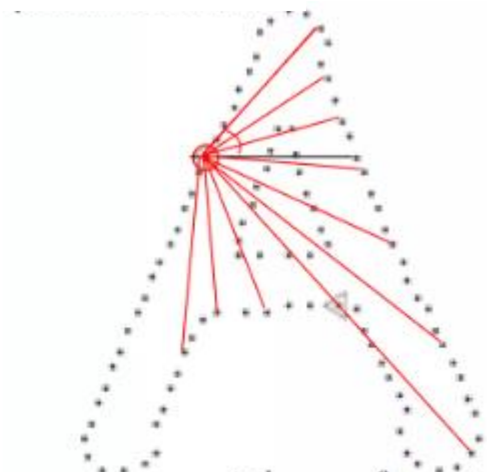
- *Step 1:* Given a shape P, obtain n samples uniformly spaced along the boundary.

$$P = \{p_1, \dots, p_n\}, p_i \in \mathbb{R}^2$$

Belongie, Malik and Puzicha. Shape Matching and Object Recognition Using Shape Contexts, IEEE PAMI 2002.

Shape Context

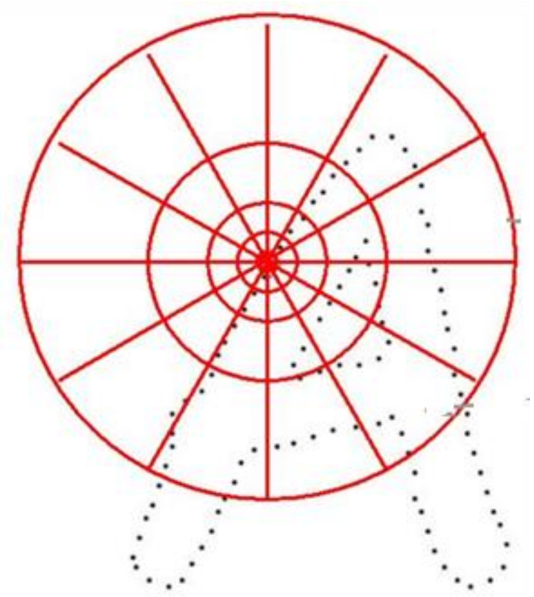
- *Step 2:* Compute Euclidean distance r and angle θ from each point to all the other n-1 points.
- Normalize r by the median distance λ and measure the angle relative to the positive x-axis.



$$Vector\_d_{p_x} = \begin{bmatrix} d_{p_x-p_1} & \theta_{p_x,p_1} \\ d_{p_x-p_2} & \theta_{p_x,p_2} \\ d_{p_x-p_3} & , \ \theta_{p_x,p_3} \\ \vdots & \vdots \\ d_{p_x-p_{n-1}} & \theta_{p_x,p_{n-1}} \end{bmatrix}$$

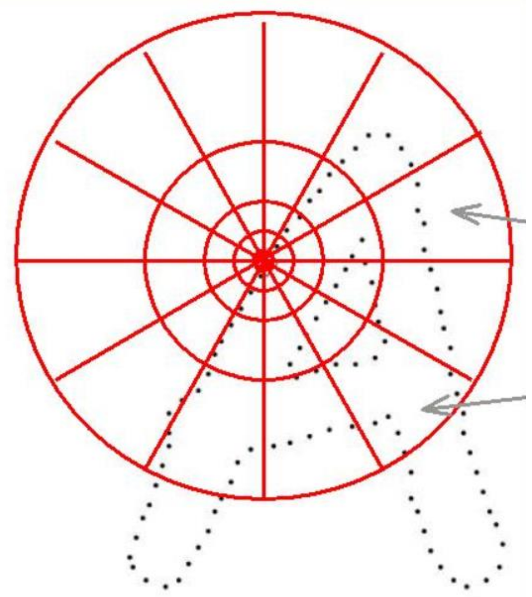
Shape Context

- *Step 3:* Compute log of r vector. Discretize the distance and angle measurements.



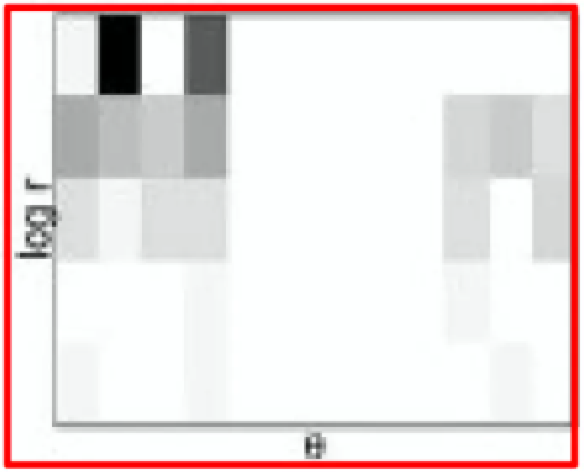
Shape Context

- *Step 4:* For each point, capture number of points that lie in a given θ, r bin.

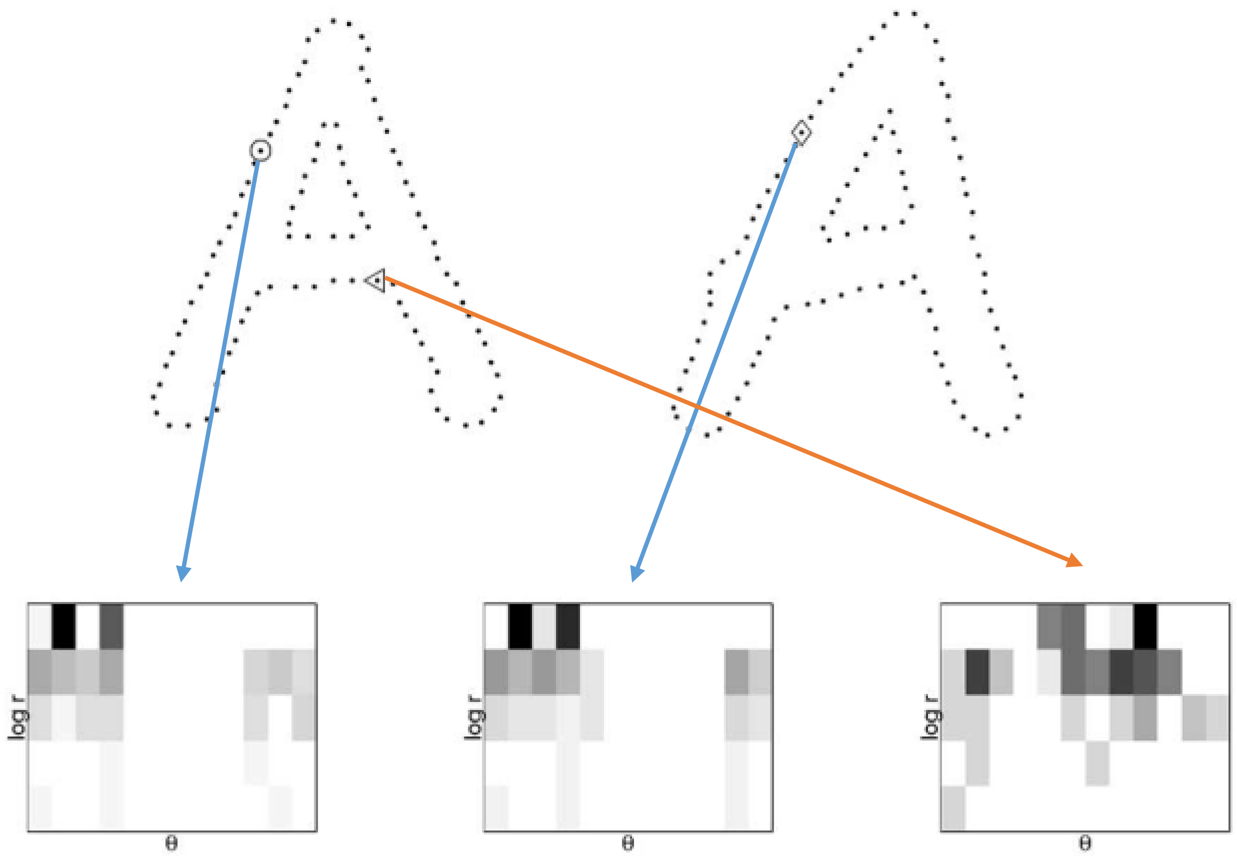


Shape Context

- For each point, shape context is a log-polar histogram of the coordinates of n-1 points measured from it.

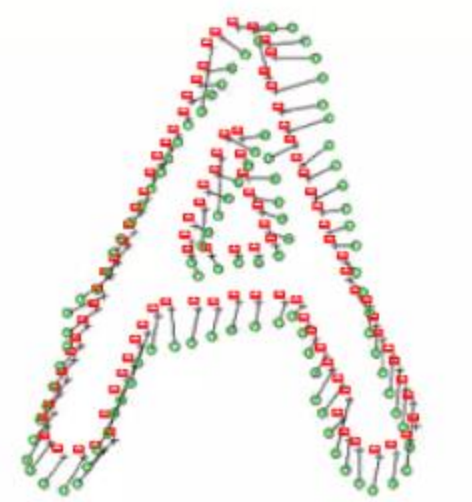


Shape Context



Shape Context

- Dissimilarity between two shapes can be formulated as the cost of their matching.

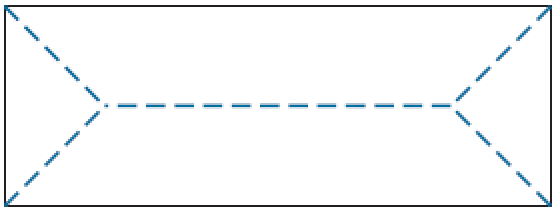


Representing Shape of a Region

Skeletons, Medial Axes, Distance Transforms

Skeletons, Medial Axes, Distance Transforms

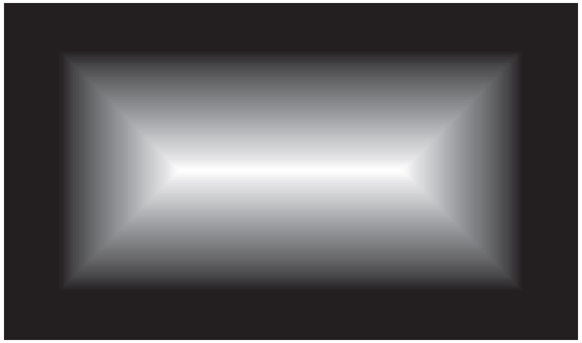
- Skeletons are computed using the points in the entire region including the boundary.
- The idea is to reduce a region to a tree or a graph by computing its skeleton.
- Skeleton of a region is the set of points in the region equidistant from the border of the region.



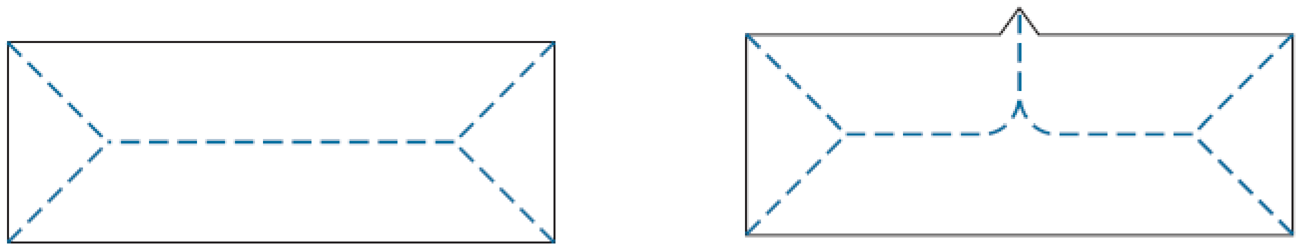
Skeletons, Medial Axes, Distance Transforms

0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0

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



Skeletons, Medial Axes, Distance Transforms



Skeletons, Medial Axes, Distance Transforms

- Tari, Shah, Pien. Extraction of shape skeletons from grayscale images, CVIU 1997.
- Aslan, Tari. An axis-based representation for recognition, ICCV 2005.

Representing Shape of a Region

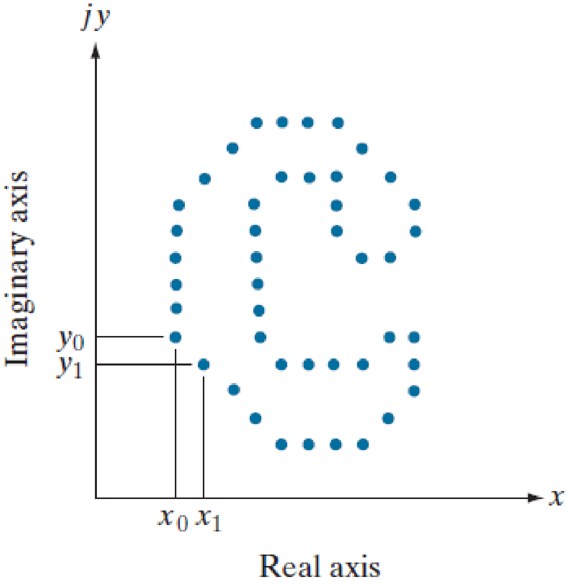
Fourier Descriptors

Fourier Descriptors

- Starting at an arbitrary point (x_0, y_0), coordinate pairs $(x_0, y_0), (x_1, y_1), (x_2, y_2), \ldots, (x_{k-1}, y_{k-1})$ are encountered in traversing the boundary in counterclockwise direction.

$$s(k) = x(k) + jy(k) \quad \text{for } k = 0, 1, 2, \ldots, K-1$$

FIGURE 11.18
A digital boundary and its representation as sequence of complex numbers. The points (x_0, y_0) and (x_1, y_1) are (arbitrarily) the first two points in the sequence.



Fourier Descriptors

discrete Fourier transform (DFT) of $s(k)$ is

$$a(u) = \sum_{k=0}^{K-1} s(k) e^{-j2\pi uk/K} \quad \text{for } u = 0, 1, 2, \ldots, K-1$$

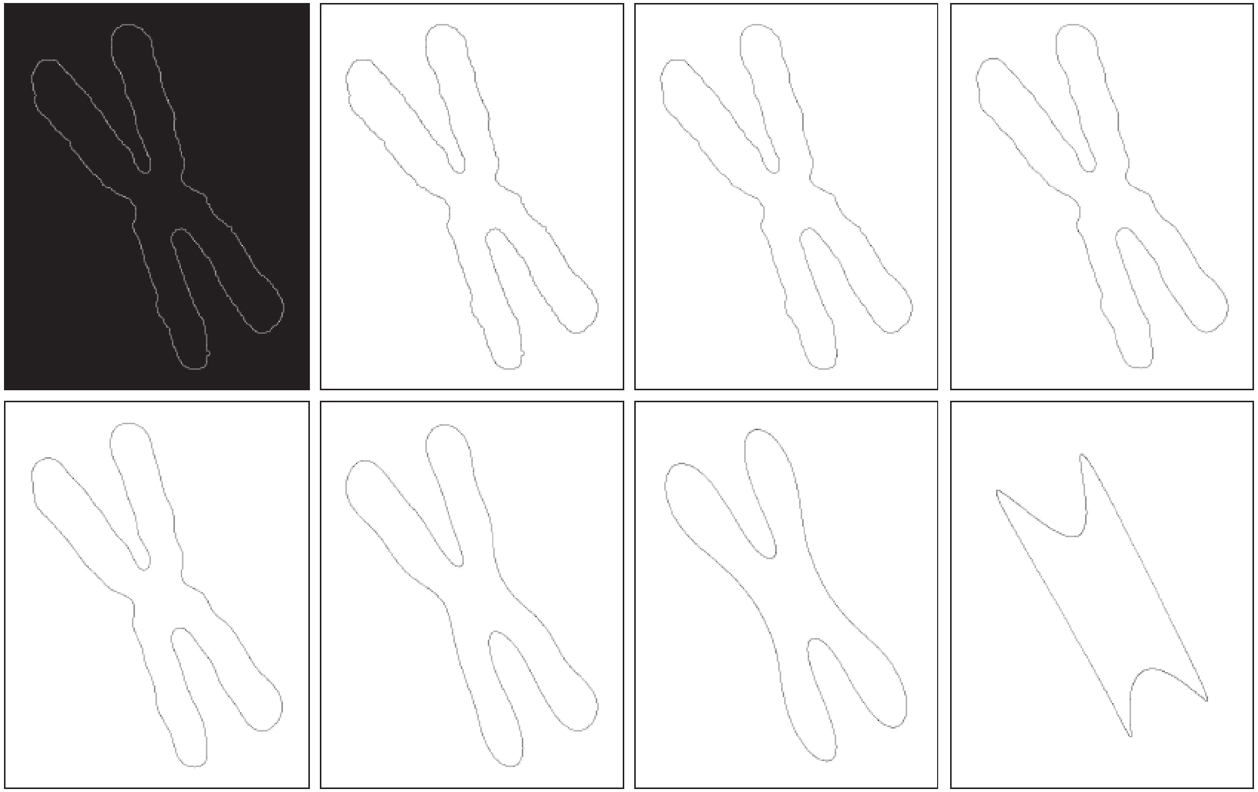
inverse Fourier transform of these coefficients restores $s(k)$

$$s(k) = \frac{1}{K} \sum_{u=0}^{K-1} a(u) e^{j2\pi uk/K} \quad \text{for } k = 0, 1, 2, \ldots, K-1$$

Instead of all Fourier coefficients, use only the first P coefficients

$$\hat{s}(k) = \frac{1}{K} \sum_{u=0}^{P-1} a(u) e^{j2\pi uk/K} \quad \text{for } k = 0, 1, 2, \ldots, K-1$$

Fourier Descriptors



a b c d
e f g h

FIGURE 11.19 (a) Boundary of a human chromosome (2868 points). (b)–(h) Boundaries reconstructed using 1434, 286, 144, 72, 36, 18, and 8 Fourier descriptors, respectively. These numbers are approximately 50%, 10%, 5%, 2.5%, 1.25%, 0.63%, and 0.28% of 2868, respectively. Images (b)–(h) are shown as negatives to make the boundaries easier to see.

Fourier Descriptors

TABLE 11.1
Some basic properties of Fourier descriptors.

Transformation	Boundary	Fourier Descriptor
Identity	$s(k)$	$a(u)$
Rotation	$s_r(k) = s(k)e^{j\theta}$	$a_r(u) = a(u)e^{j\theta}$
Translation	$s_t(k) = s(k) + \Delta_{xy}$	$a_t(u) = a(u) + \Delta_{xy}\delta(u)$
Scaling	$s_s(k) = \alpha s(k)$	$a_s(u) = \alpha a(u)$
Starting point	$s_p(k) = s(k - k_0)$	$a_p(u) = a(u)e^{-j2\pi k_0 u/K}$

Representing Texture

Texture

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

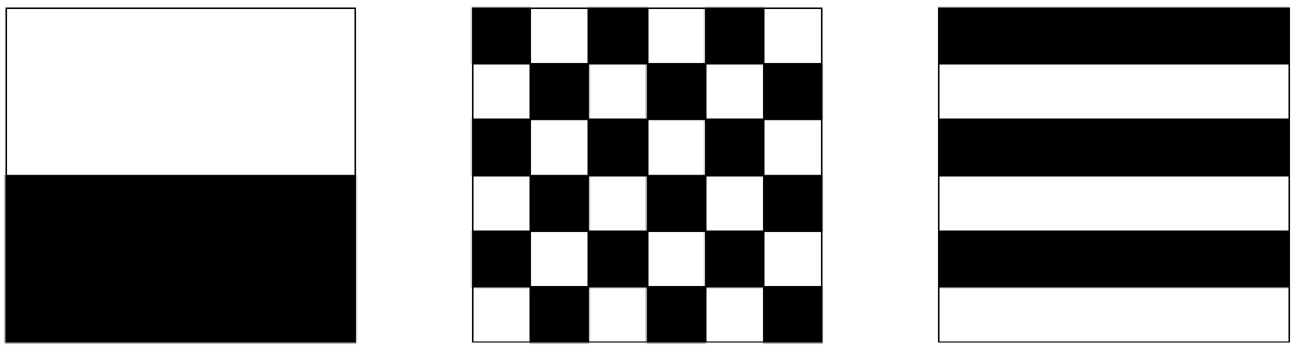


Figure 7.2: Three different textures with the same distribution of black and white.

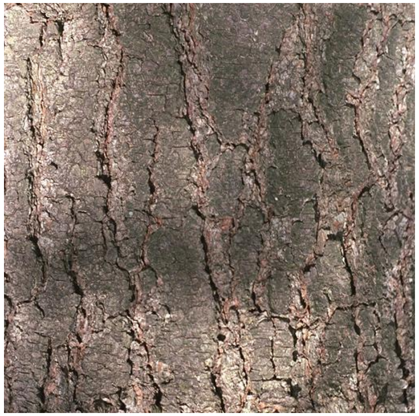
Slide credit: S. Aksoy

Texture

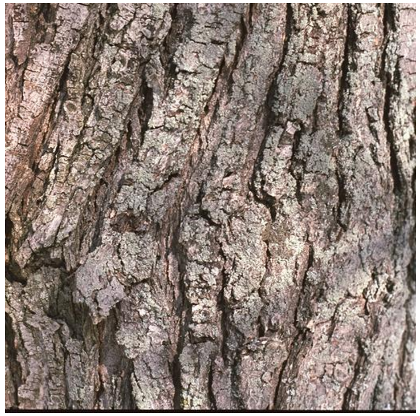
- Although no formal definition of texture exists, intuitively, it can be defined as uniformity, density, coarseness, roughness, regularity and directionality of discrete tonal features and their spatial relationships.
- Texture is commonly found in natural scenes, particularly, in outdoor scenes containing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

Slide credit: S. Aksoy

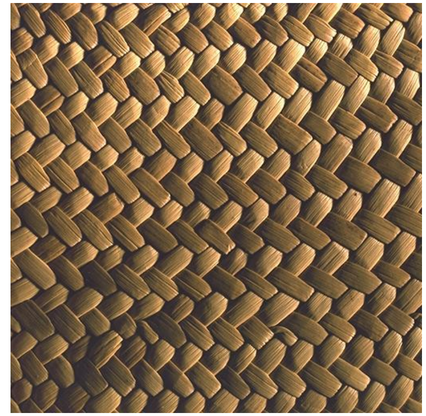
Texture



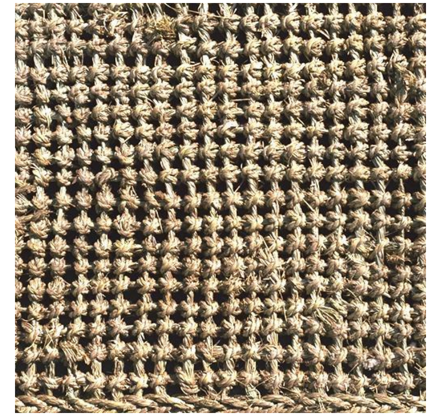
Bark



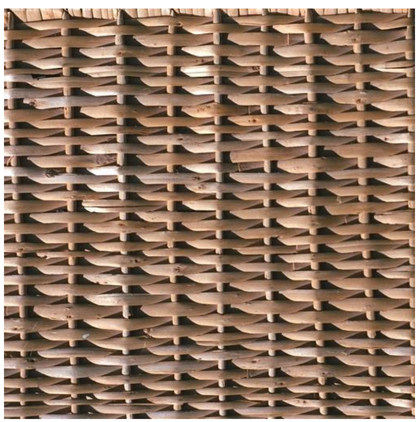
Bark



Fabric



Fabric



Fabric

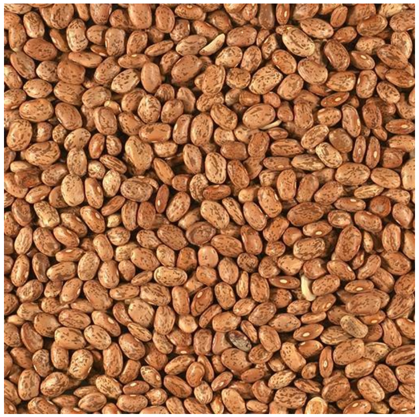
Flowers

Flowers

Flowers

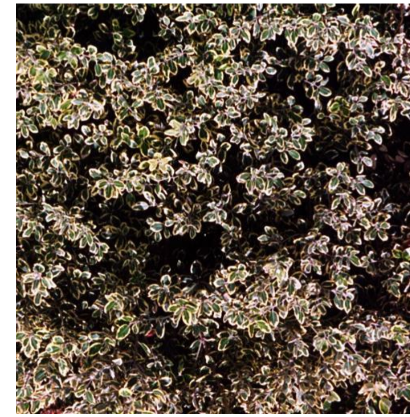
Slide credit: S. Aksoy

Texture


Food


Food

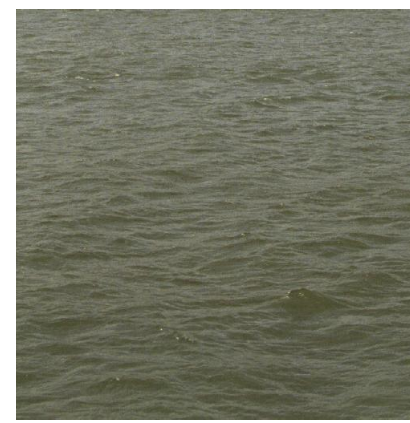

Leaves


Leaves


Leaves


Leaves

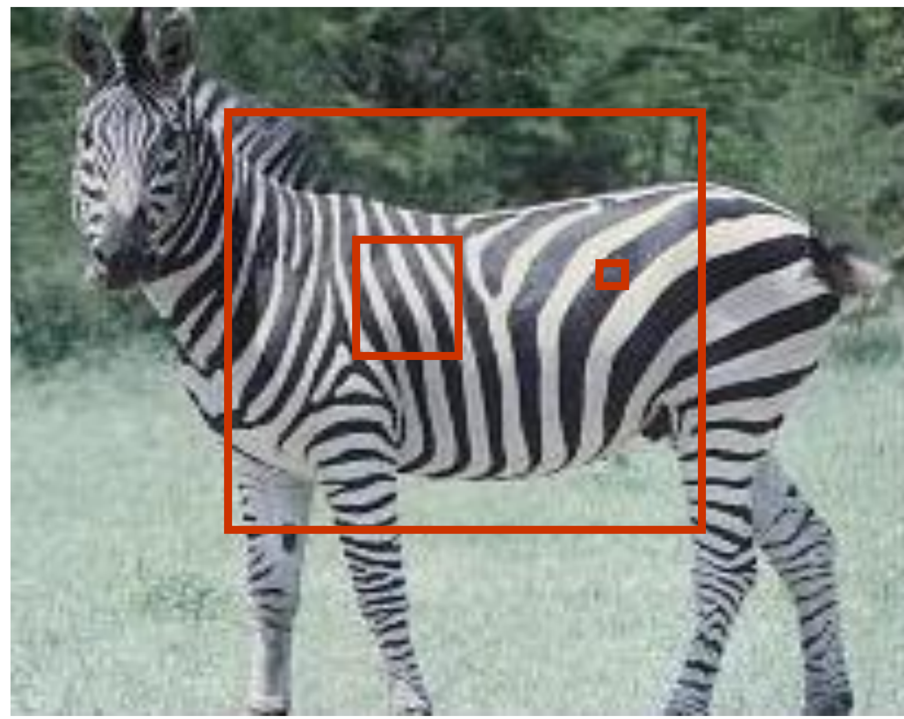

Water


Water

Slide credit: S. Aksoy

Texture

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



Slide credit: S. Aksoy

Texture

- The approaches for characterizing and measuring texture can be grouped as:
  - **structural approaches** use the idea that textures are made of primitives appearing in a near-regular repetitive arrangement,
  - **statistical approaches** yield a quantitative measure of the arrangement of intensities.
- 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.

Slide credit: S. Aksoy

Texture

- 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

Slide credit: S. Aksoy

Representing Texture

Co-occurrence matrices

Co-occurrence matrices

There are 8 different intensity levels → G is 8x8
The position operator is defined as → one pixel immediately to the right

FIGURE 11.30 How to construct a co-occurrence matrix.

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**

Co-occurrence matrices

$$p_{ij} = \frac{g_{ij}}{n} \xrightarrow{\hspace{2em}} \text{sum of elements in G}$$

$$m_r = \sum_{i=1}^{K} i \sum_{j=1}^{K} p_{ij} \qquad \sigma_r^2 = \sum_{i=1}^{K} (i - m_r)^2 \sum_{j=1}^{K} p_{ij}$$

$$m_c = \sum_{j=1}^{K} j \sum_{i=1}^{K} p_{ij} \qquad \sigma_c^2 = \sum_{j=1}^{K} (j - m_c)^2 \sum_{i=1}^{K} p_{ij}$$

m_r and m_c are means computed along rows and columns of the normalized G.

σ_r^2 and σ_c^2 are variances computed along rows and columns of the normalized G.

Co-occurrence matrices

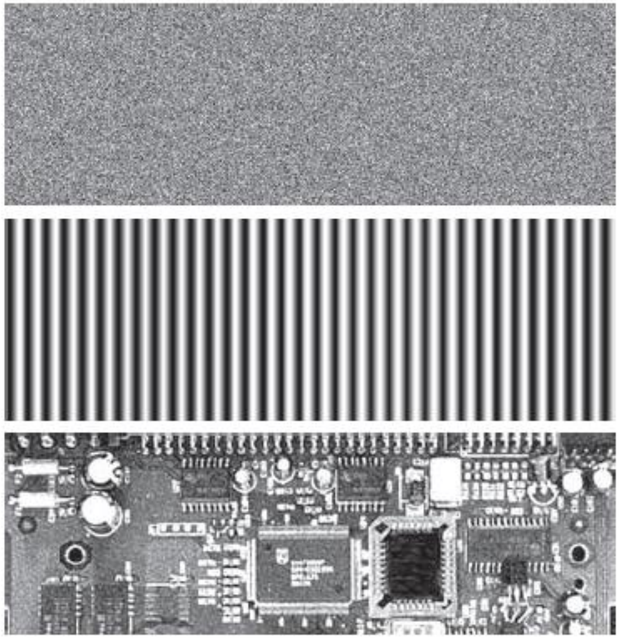
TABLE 11.3
Descriptors used for characterizing co-occurrence matrices of size $K \times K$. The term p_{ij} is the ij-th term of **G** divided by the sum of the elements of **G**.

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. The 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 to the diagonal of the distribution of elements in **G**. The range of values is [0, 1], with the maximum being achieved when **G** is a diagonal matrix.	$\sum_{i=1}^{K}\sum_{j=1}^{K}\frac{p_{ij}}{1+\lvert i-j\rvert}$
Entropy	Measures the randomness of the elements of **G**. The entropy is 0 when all p_{ij}'s are 0, and is maximum when the p_{ij}'s are uniformly distributed. The maximum value is thus $2\log_2 K$.	$-\sum_{i=1}^{K}\sum_{j=1}^{K}p_{ij}\log_2 p_{ij}$

Co-occurrence matrices

a
b
c

FIGURE 11.31 Images whose pixels have (a) random, (b) periodic, and (c) mixed texture patterns. Each image is of size 263×800 pixels.



a b c

FIGURE 11.32 256×256 co-occurrence matrices $\mathbf{G}_1$, $\mathbf{G}_2$, and $\mathbf{G}_3$, corresponding from left to right to the images in Fig. 11.31.

Co-occurrence matrices

a
b
c

FIGURE 11.31 Images whose pixels have (a) random, (b) periodic, and (c) mixed texture patterns. Each image is of size 263×800 pixels.

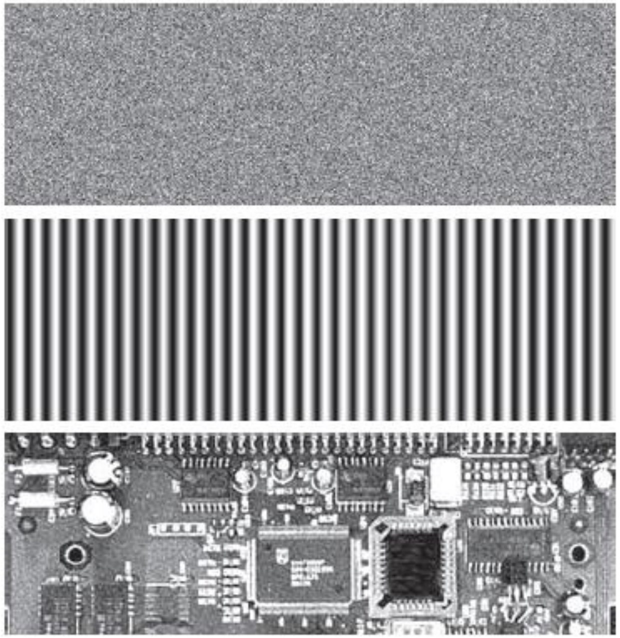


TABLE 11.4
Descriptors evaluated using the co-occurrence matrices displayed as images in Fig. 11.32.

Normalized Co-occurrence Matrix	Maximum Probability	Correlation	Contrast	Uniformity	Homogeneity	Entropy
$\mathbf{G}_1/n_1$	0.00006	−0.0005	10838	0.00002	0.0366	15.75
$\mathbf{G}_2/n_2$	0.01500	0.9650	00570	0.01230	0.0824	06.43
$\mathbf{G}_3/n_3$	0.06860	0.8798	01356	0.00480	0.2048	13.58

Representing Texture

Gabor filters

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 $\sigma_v = 1/2\pi\sigma_y$

Slide credit: S. Aksoy

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

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

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

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

Gabor filters

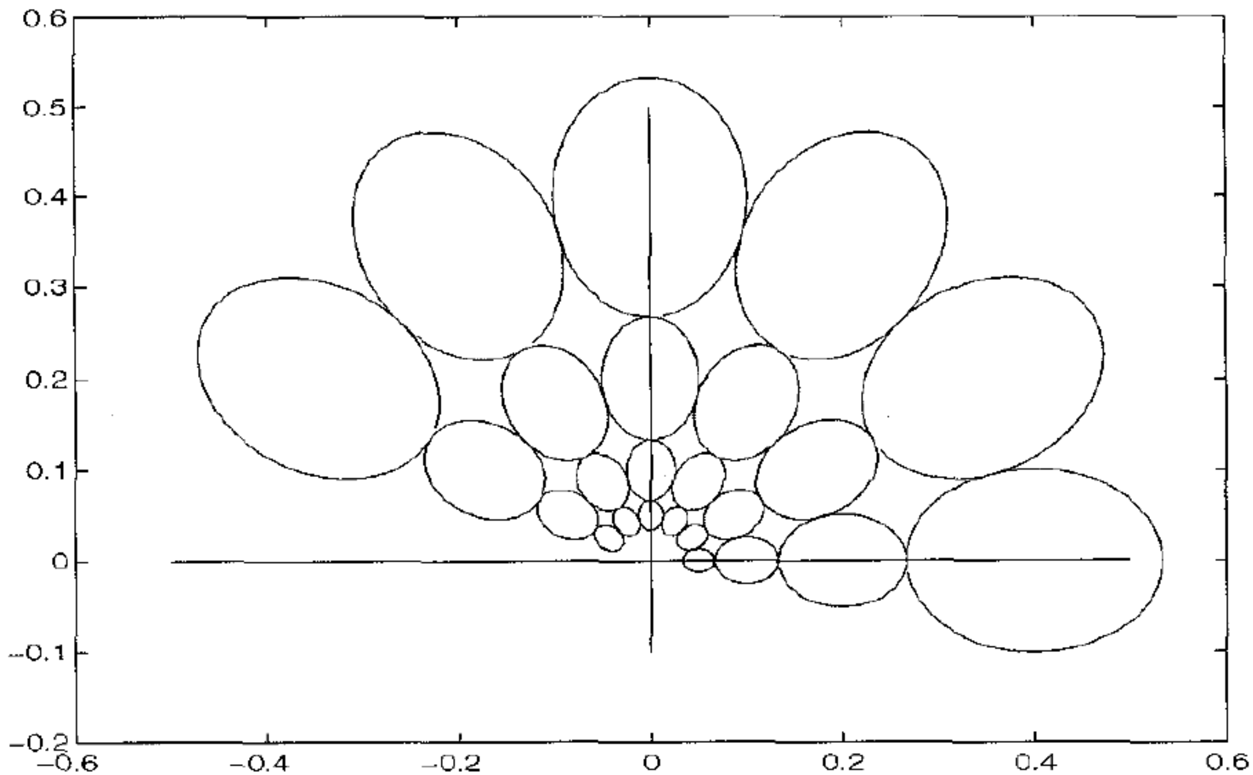
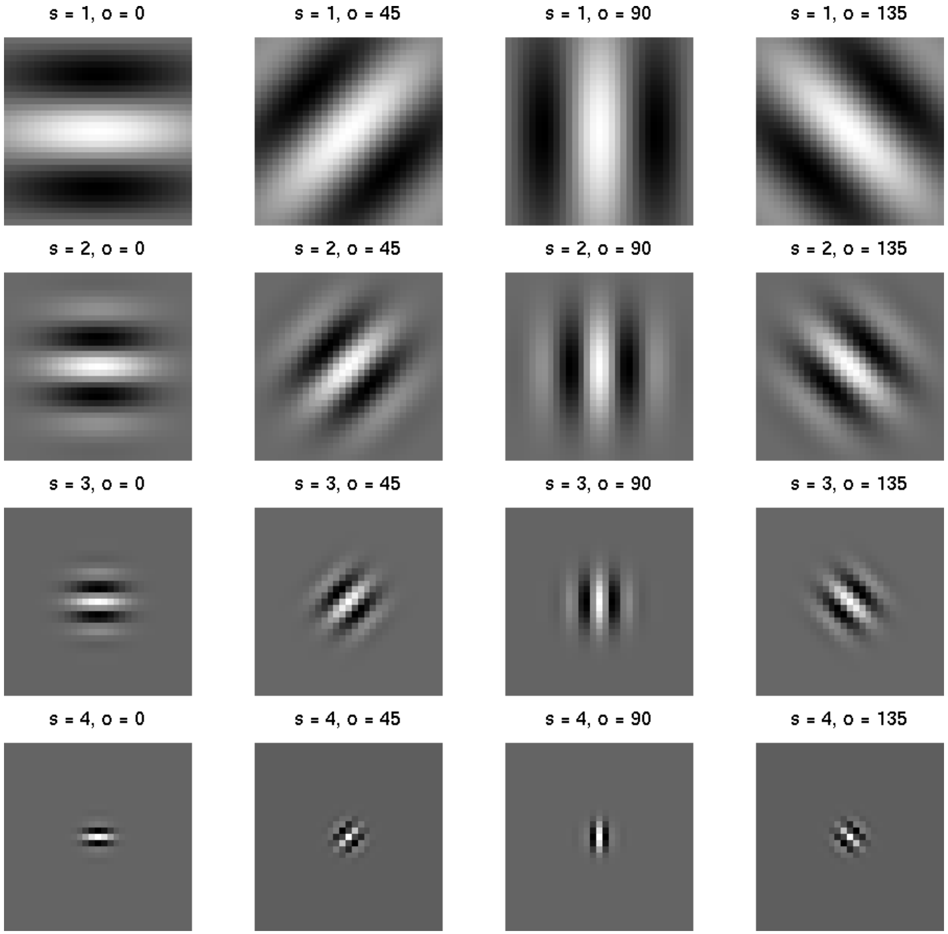


Fig. 1. The contours indicate the half-peak magnitude of the filter responses in the Gabor filter dictionary. The filter parameters used are $U_h = 0.4$, $U_l = 0.05$, $K = 6$, and $S = 4$.

Slide credit: S. Aksoy

Gabor filters

Filters at multiple scales and orientations.



Slide credit: S. Aksoy

Whole Image Features

Scale Invariant Feature Transform (SIFT)

Scale Invariant Feature Transform (SIFT)

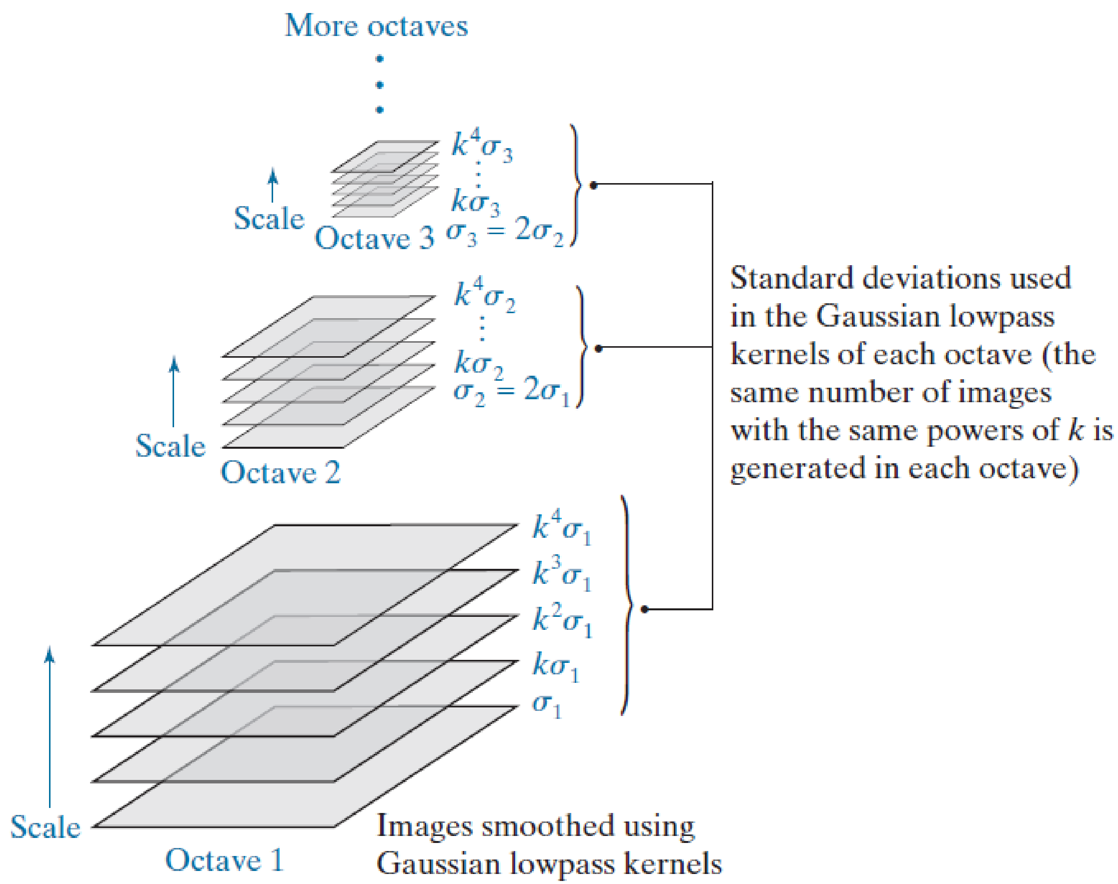
- SIFT is an algorithm developed by Lowe (2004) for extracting invariant features from an image.
- It is called transform because it transforms image data into scale-invariant coordinates relative to local image features.
- Scale changes, rotation, changes in illumination, changes in viewpoint → Use methods like SIFT
- SIFT features (keypoints) are invariant to image scale and rotation and robust across a range of affine distortions, changes in 3D viewpoint, noise and changes of illumination.
- The input to SIFT is an image and its output is an n-dimensional feature vector whose elements are the invariant feature descriptors.

Scale Invariant Feature Transform (SIFT)

1. **Construct scale space:** The parameters are σ, s, (k is computed from s), and the number of octaves. Suggested values are $\sigma = 1.6$, s = 2, and three octaves. → Invariance to scale
2. **Obtain the initial keypoints:** Compute the difference of Gaussians from the smoothed images in the scale space. Find extrema in each difference image as the initial keypoints.
3. **Improve the accuracy of the location of the keypoints:** Interpolate the values of difference images via a Taylor expansion.
4. **Delete unsuitable keypoints:** Eliminate keypoints that have low contrast and/or are poorly localized.
5. **Compute keypoint orientations:** Compute the magnitude and the orientation of each keypoint using the histogram based procedure. → Invariance to rotation
6. **Compute keypoint descriptors:** Compute a feature vector for each keypoint. If a region of size 16 x 16 around each keypoint is used, the result will be 128-dimensional feature vector.

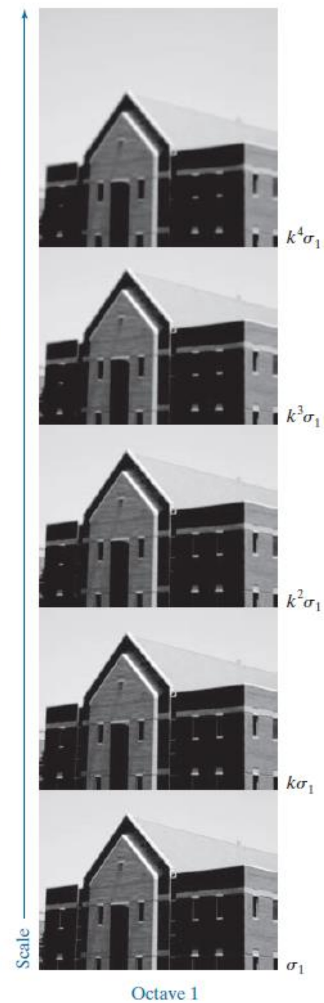
Scale Invariant Feature Transform (SIFT): **Construct scale space**

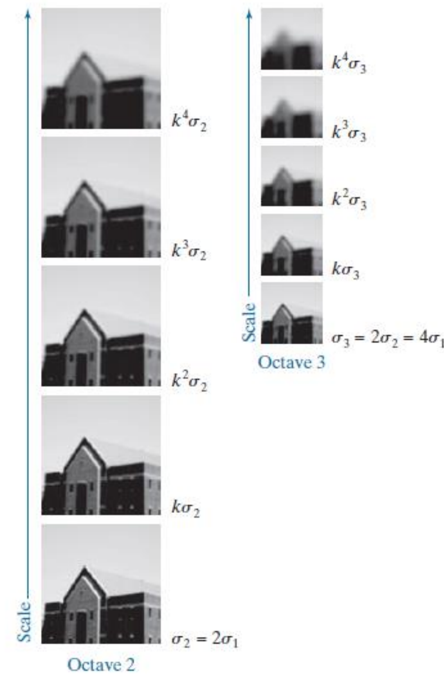
FIGURE 11.56
Scale space, showing three octaves. Because $s = 2$ in this case, each octave has five smoothed images. A Gaussian kernel was used for smoothing, so the space parameter is σ.



Scale Invariant Feature Transform (SIFT): **Construct scale space**

FIGURE 11.57 Illustration using images of the first three octaves of scale space in SIFT. The entries in the table are values of standard deviation used at each scale of each octave. For example the standard deviation used in scale 2 of octave 1 is $k\sigma_1$, which is equal to 1.0. (The images of octave 1 are shown slightly overlapped to fit in the figure space.)





$\sigma_1 = \sqrt{2}/2 = 0.707 \quad k = \sqrt{2} = 1.414$

Octave	Scale				
	1	2	3	4	5
1	0.707	1.000	1.414	2.000	2.828
2	1.414	2.000	2.828	4.000	5.657
3	2.828	4.000	5.657	8.000	11.314

Scale Invariant Feature Transform (SIFT): **Obtain the initial keypoints**

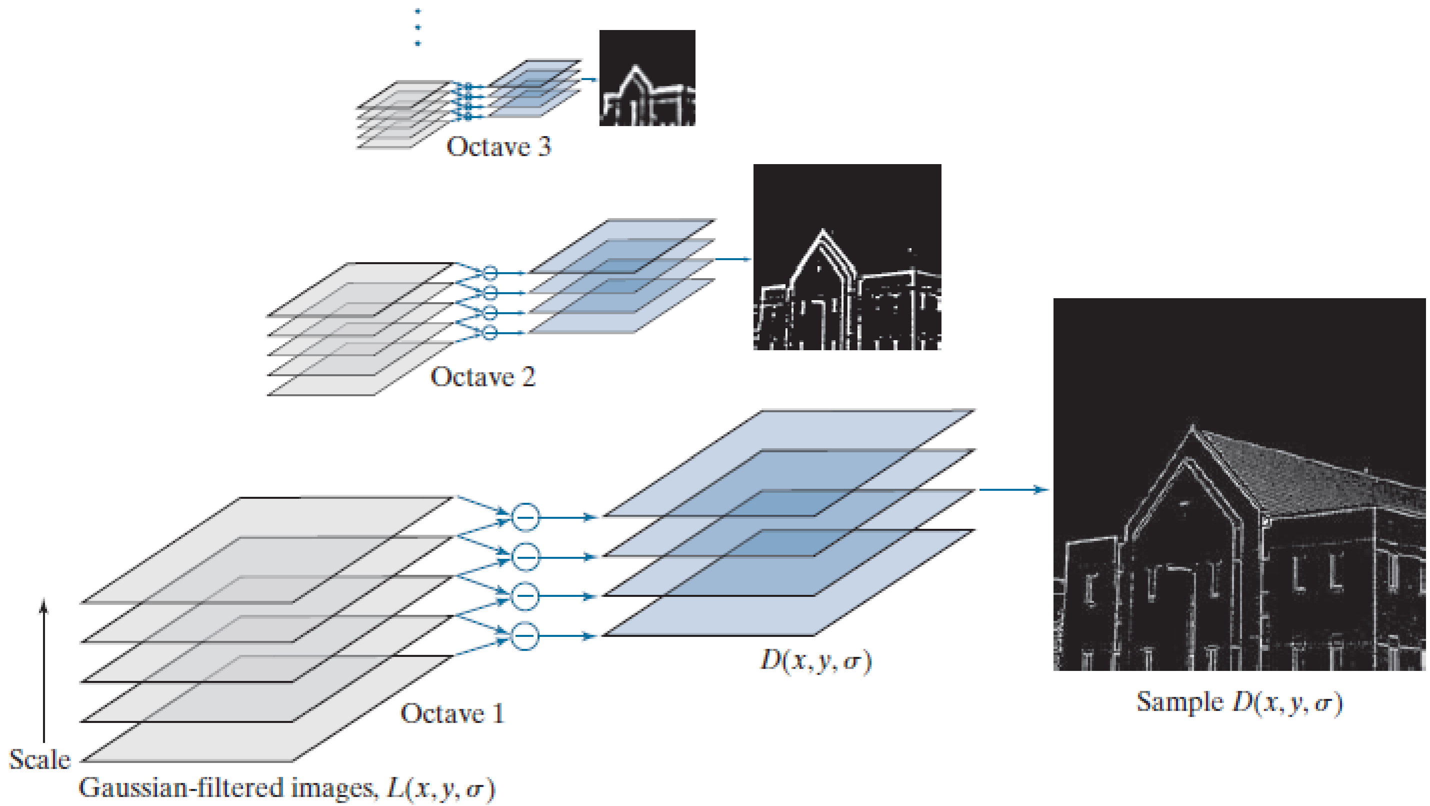
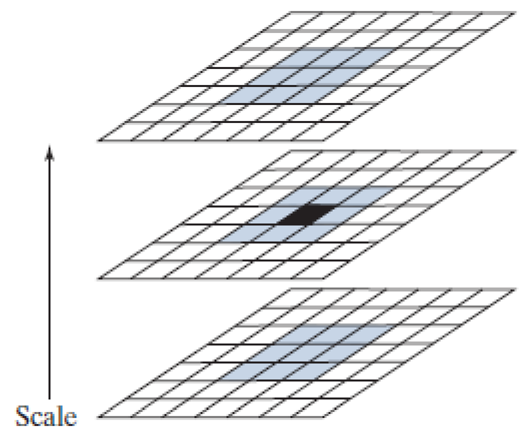


FIGURE 11.58 How Eq. (11-69) is implemented in scale space. There are $s+3$ $L(x, y, \sigma)$ images and $s+2$ corresponding $D(x, y, \sigma)$ images in each octave.

Scale Invariant Feature Transform (SIFT): **Obtain the initial keypoints**

FIGURE 11.59 Extrema (maxima or minima) of the $D(x, y, \sigma)$ images in an octave are detected by comparing a pixel (shown in black) to its 26 neighbors (shown shaded) in 3×3 regions at the current and adjacent scale images.



Scale Invariant Feature Transform (SIFT): **Improve accuracy of location of keypoints**

- Fit an interpolating function at each extremum point and look for an improved extremum location in the interpolated function.

Scale Invariant Feature Transform (SIFT): **Delete unsuitable keypoints**

- Eliminate keypoints at which the difference is less than a threshold such as 0.03.
- Eliminate keypoints associated with edges.

FIGURE 11.60 SIFT keypoints detected in the building image. The points were enlarged slightly to make them easier to see.

Scale Invariant Feature Transform (SIFT): **Compute keypoint orientations**

A histogram of orientations is formed from the gradient orientations of sample points in a neighborhood of each keypoint. The histogram has 36 bins covering the 360° range of orientations on the image plane. Each sample added to the histogram is weighed by its gradient magnitude, and by a circular Gaussian function with a standard deviation 1.5 times the scale of the keypoint.

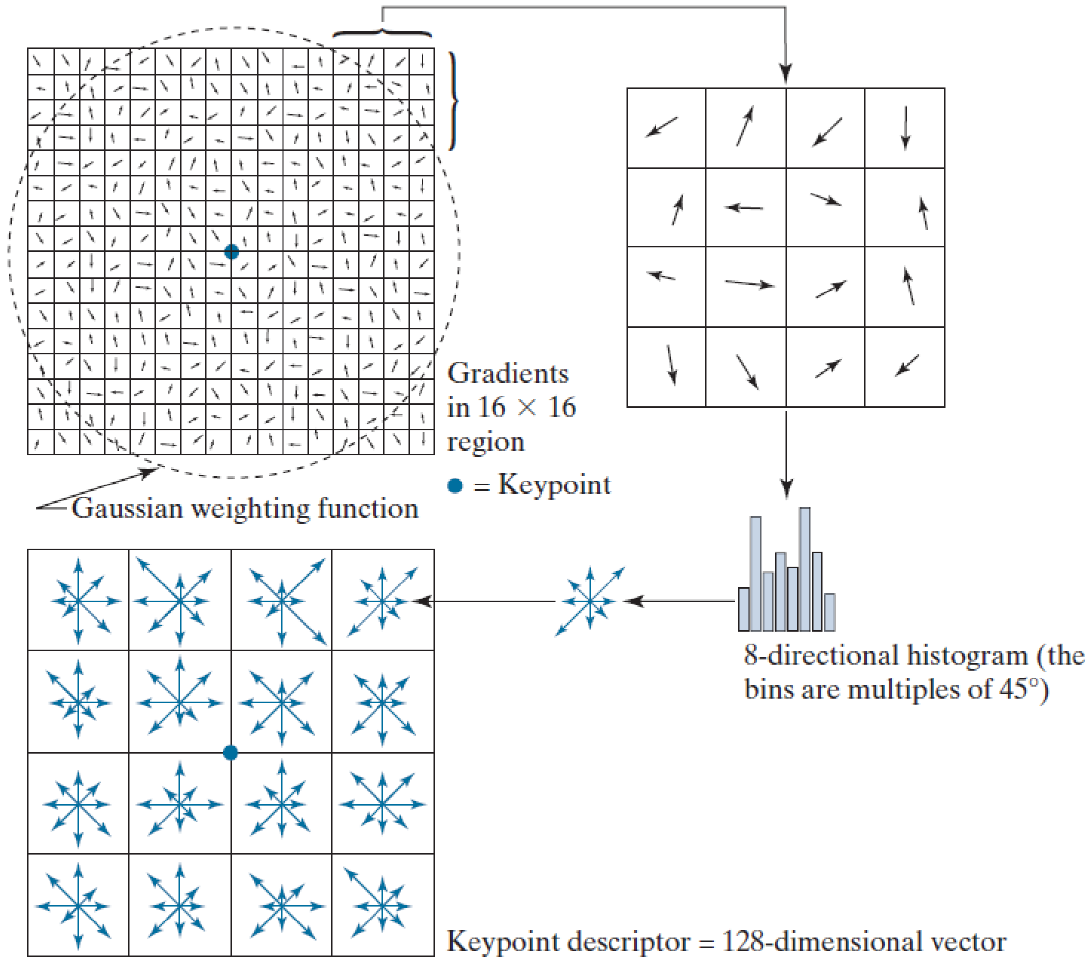
Peaks in the histogram correspond to dominant local directions of local gradients. The highest peak in the histogram is detected and any other local peak that is within 80% of the highest peak is used also to create another keypoint with that orientation.

Scale Invariant Feature Transform (SIFT): **Compute keypoint orientations**

FIGURE 11.61 The keypoints from Fig. 11.60 superimposed on the original image. The arrows indicate keypoint orientations.

Scale Invariant Feature Transform (SIFT): **Compute keypoint descriptors**

FIGURE 11.62 Approach used to compute a keypoint descriptor.



Scale Invariant Feature Transform (SIFT)

a b

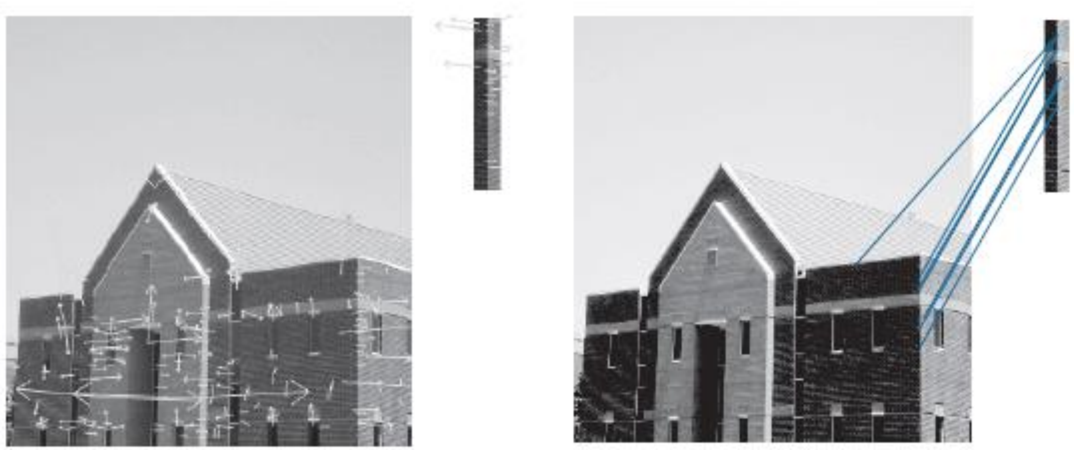
FIGURE 11.63 (a) Keypoints and their directions (shown as gray arrows) for the building image and for a section of the right corner of the building. The subimage is a separate image and was processed as such. (b) Corresponding key points between the building and the subimage (the straight lines shown connect pairs of matching points). Only three of the 36 matches found are incorrect.

Scale Invariant Feature Transform (SIFT)

a b

FIGURE 11.64 (a) Keypoints for the rotated (by 5°) building image and for a section of the right corner of the building. The subimage is a separate image and was processed as such. (b) Corresponding keypoints between the corner and the building. Of the 26 matches found, only two are in error.

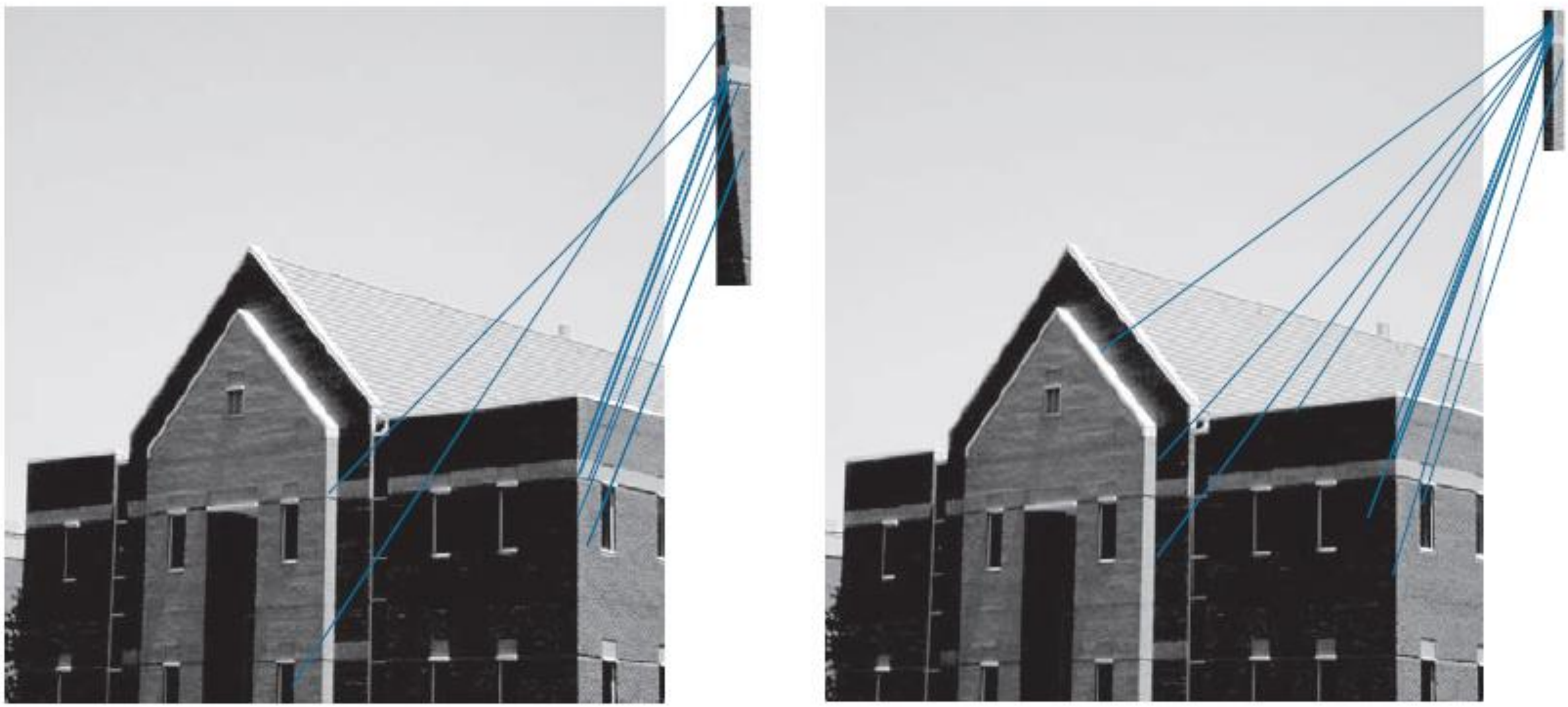
Scale Invariant Feature Transform (SIFT)



a b

FIGURE 11.65 (a) Keypoints for the half-sized building and a section of the right corner. (b) Corresponding keypoints between the corner and the building. Of the seven matches found, only one is in error.

Scale Invariant Feature Transform (SIFT)



a b

FIGURE 11.66 (a) Matches between the original building image and a rotated version of a segment of its right corner. Ten matches were found, of which two are incorrect. (b) Matches between the original image and a half-scaled version of a segment of its right corner. Here, 11 matches were found, of which four were incorrect.