Digital Image Processing
Chapter 11:
Image Description and Representation

Image Representation and Description?

Objective:

To represent and describe information embedded in an image in other forms that are more suitable than the image itself.

Benefits:

- Easier to understand
- Require fewer memory, faster to be processed
- More "ready to be used"

What kind of information we can use?

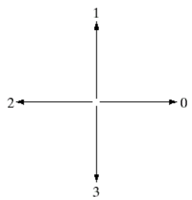
- Boundary, shape
- Region
- Texture
- Relation between regions

Shape Representation by Using Chain Codes

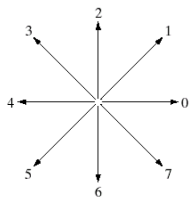
Why we focus on a boundary?

The boundary is a good representation of an object shape and also requires a few memory.

Chain codes: represent an object boundary by a connected sequence of straight line segments of specified length and direction.



4-directional chain code

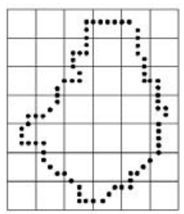


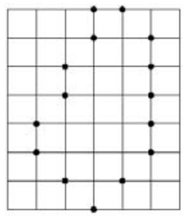
8-directional chain code

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Examples of Chain Codes

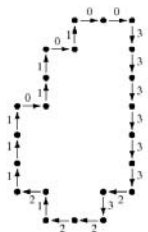
Object boundary (resampling)

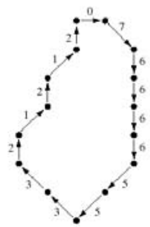




Boundary vertices

4-directional chain code





8-directional chain code

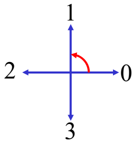
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

The First Difference of a Chain Codes

Problem of a chain code:
a chain code sequence depends on a starting point.
Solution: treat a chain code as a circular sequence and redefine the starting point so that the resulting sequence of numbers forms an integer of minimum magnitude.

The first difference of a chain code: counting the number of direction change (in counterclockwise) between 2 adjacent elements of the code.

Example:



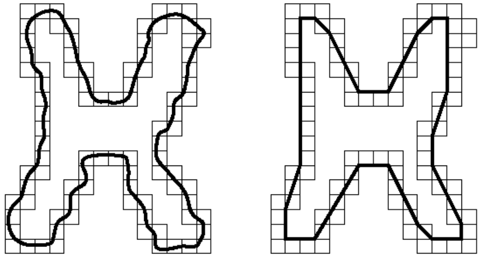
Chain code :	The first difference
0 → 1	1
0 → 2	2
0 → 3	3
2 → 3	1
2 → 0	2
2 → 1	3

Example:
- a chain code: 10103322
- The first difference = 3133030
- Treating a chain code as a circular sequence, we get the first difference = 33133030

The first difference is rotational invariant.

Polygon Approximation

Represent an object boundary by a polygon

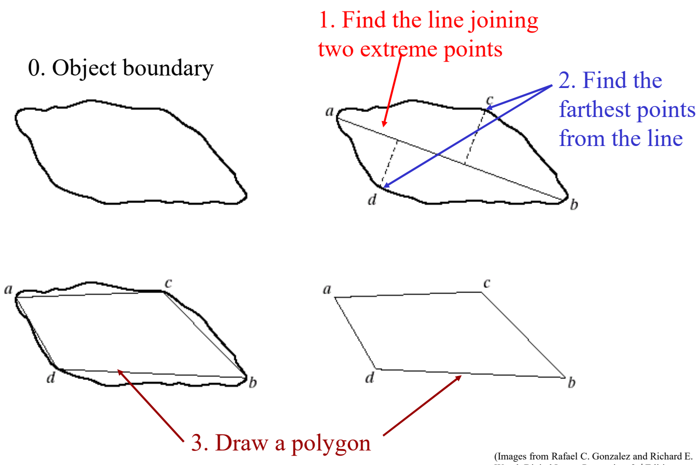


Object boundary

Minimum perimeter polygon

Minimum perimeter polygon consists of line segments that minimize distances between boundary pixels.

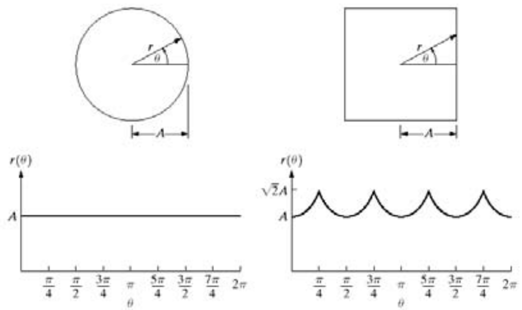
Polygon Approximation:Splitting Techniques



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

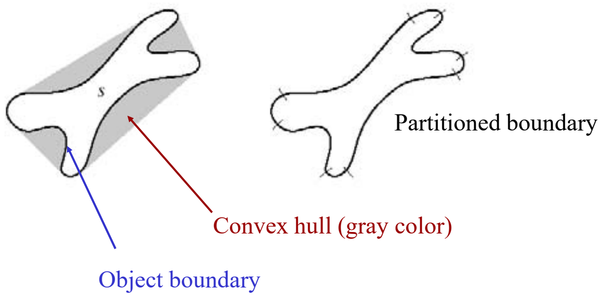
Distance-Versus-Angle *Signatures*

- Represent a 2-D object boundary in term of a 1-D function(*signature*) of radial distance with respect to θ.
- Invariant to translation, but depend on rotation and scaling



Boundary Segments

Concept: Partitioning an object boundary by using vertices of a convex hull.



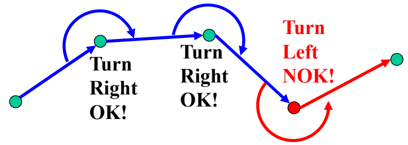
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Convex Hull Algorithm

Input : A set of points on a cornea boundary
Output: A set of points on a boundary of a cor
1. Sort the points by x-coordinate to get a seq

For the upper side of a convex hull

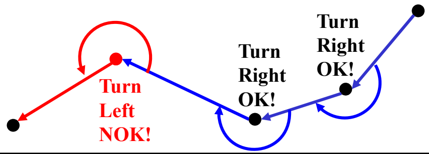
2. **Put** the points p_1 and p_2 in a list L_{upper} with p
3. **For** i = 3 to n
4. **Do** append p_i to L_{upper}
5. **While** L_{upper} contains more than 2 points an points in L_{upper} do not make a right tu
6. **Do delete the middle point** of the l



Convex Hull Algorithm (cont.)

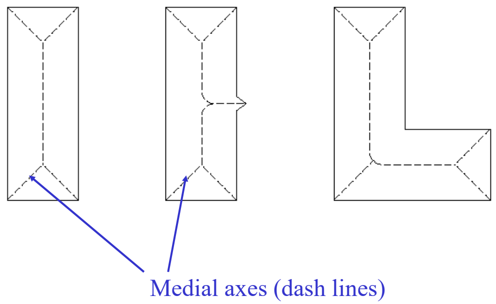
For the lower side of a convex hull

7. **Put** the points p_n and p_{n-1} in a list L_{lower} with
8. **For** i = n-2 down to 1
9. **Do** append p_i to L_{lower}
10. **While** L_{lower} contains more than 2 points ar in L_{lower} do not make a right turn
11. **Do delete the middle point** of the la
12. **Remove** the first and the last points from l
13. **Append** L_{lower} to L_{upper} resulting in the list l
14. **Return** L



Skeletons

Obtained from *thinning* or *skeletonizing* processes



a b c

FIGURE 11.7 Medial axes (dashed) of three simple regions.

Thinning Algorithm

Concept:
1. Do not remove *end points*
2. Do not break *connectivity*
3. Do not cause *excessive erosion*

Apply only to contour pixels: pixels "1" having at least one of its 8 neighbor pixels valued "0"

Notation:

Let

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

= Neighborhood arrangement for the thinning algorithm

Let $N(p_1) = p_2 + p_3 + \dots + p_8 + p_9$

$T(p_1)$ = the number of transition 0-1 in the ordered sequence $p_2, p_3, \dots, p_8, p_9, p_2$.

Example

0	0	1
1	p_1	0
1	0	1

$N(p_1) = 4$
$T(p_1) = 3$

Thinning Algorithm (cont.)

Step 1. Mark pixels for deletion if the following conditions are true.

a) $2 \le N(p_1) \le 6$
b) $T(p_1) = 1$
c) $p_2 \cdot p_4 \cdot p_6 = 0$
d) $p_4 \cdot p_6 \cdot p_8 = 0$

(Apply to all border pixels)

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

Step 2. Delete marked pixels and go to Step 3.

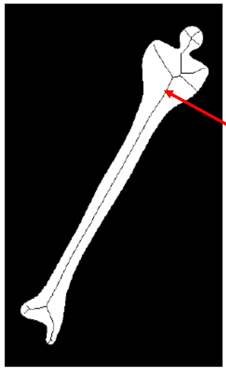
Step 3. Mark pixels for deletion if the following conditions are true.

a) $2 \le N(p_1) \le 6$
b) $T(p_1) = 1$
c) $p_2 \cdot p_4 \cdot p_8 = 0$
d) $p_2 \cdot p_6 \cdot p_8 = 0$

(Apply to all border pixels)

Step 4. Delete marked pixels and repeat Step 1 until no change occurs.

Example: Skeletons Obtained from the Thinning Alg.



Skeleton

FIGURE 11.10 Human leg bone and skeleton of the region shown superimposed.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Boundary Descriptors

1. Simple boundary descriptors:

we can use

- Length of the boundary
- The size of smallest circle or box that can totally enclosing the object

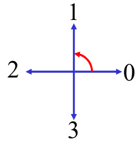
2. Shape number

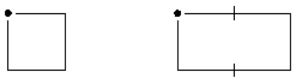
3. Fourier descriptor

4. Statistical moments

Shape Number

Shape number of the boundary definition:
the first difference of smallest magnitude
The order n of the shape number:
the number of digits in the sequence





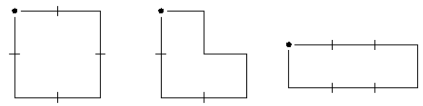
	Order 4	Order 6
Chain code:	0 3 2 1	0 0 3 2 2 1
Difference:	3 3 3 3	3 0 3 3 0 3
Shape no.:	3 3 3 3	0 3 3 0 3 3

Shape Number (cont.)

	Order 4	Order 6
Chain code:	0 3 2 1	0 0 3 2 2 1
Difference:	3 3 3 3	3 0 3 3 0 3
Shape no.:	3 3 3 3	0 3 3 0 3 3

Shape numbers of order 4, 6 and 8

Order 8

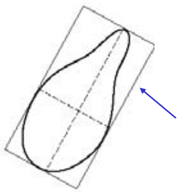


Chain code:	0 0 3 3 2 2 1 1	0 3 0 3 2 2 1 1	0 0 0 3 2 2 2 1
Difference:	3 0 3 0 3 0 3 0	3 3 1 3 3 0 3 0	3 0 0 3 3 0 0 3
Shape no.:	0 3 0 3 0 3 0 3	0 3 0 3 3 1 3 3	0 0 3 3 0 0 3 3

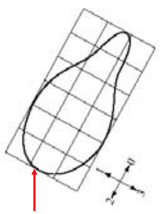
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Example: Shape Number

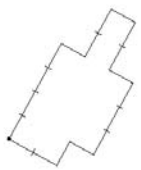
1. Original boundary



2. Find the smallest rectangle that fits the shape



3. Create grid



4. Find the nearest Grid.

Chain code:
0 0 0 0 3 0 0 3 2 2 3 2 2 2 1 2 1 1

First difference:
3 0 0 0 3 1 0 3 3 0 1 3 0 0 3 1 3 0

Shape No.
0 0 3 1 0 3 3 0 1 3 0 0 3 1 3 0 3

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Fourier Descriptor

Fourier descriptor: view a coordinate (x,y) as a complex number (x = *real part* and y = *imaginary part*) then apply the Fourier transform to a sequence of boundary points.

Let s(k) be a coordinate of a boundary point k :

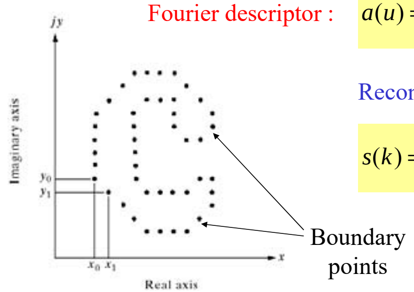
$$s(k) = x(k) + jy(k)$$

Fourier descriptor :

$$a(u) = \frac{1}{K}\sum_{k=0}^{K-1} s(k)e^{-2\pi uk/K}$$

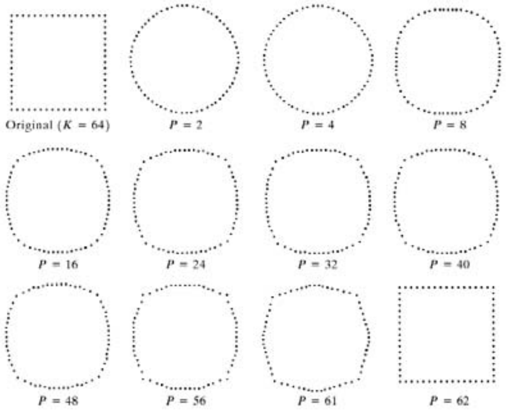
Reconstruction formula

$$s(k) = \frac{1}{K}\sum_{k=0}^{K-1} a(u)e^{2\pi uk/K}$$



Example: Fourier Descriptor

Examples of reconstruction from Fourier descriptors



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

P is the number of Fourier coefficients used to reconstruct the boundary

Fourier Descriptor Properties

Some 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	$s_p(k) = s(k - k_0)$	$a_p(u) = a(u)e^{-j2\pi k_0 u/K}$

Statistical Moments

Definition: the n[th] moment

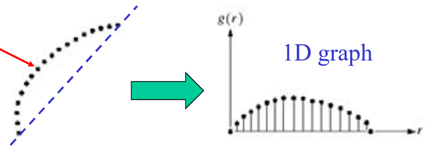
$$\mu_n(r) = \sum_{i=0}^{K-1}(r_i - m)^n g(r_i)$$

where

$$m = \sum_{i=0}^{K-1} r_i g(r_i)$$

The first moment = *mean*
The second moment = *variance*
Other *high order moments*

Boundary segment



1D graph

1. Convert a boundary segment into 1D graph
2. View a 1D graph as a PDF function
3. Compute the n[th] order moment of the graph

Regional Descriptors

Purpose: to describe regions or "areas"

1. Some simple regional descriptors

- area of the region
- length of the boundary (perimeter) of the region
- Compactness

$$C = \frac{A(R)}{P^2(R)}$$

where $A(R)$ and $P(R)$ = area and perimeter of region R

Example: a circle is the most compact shape with C = 1/4π

2. Topological Descriptors
3. Texture
4. Moments of 2D Functions

Example: Regional Descriptors

Infrared image of America at night

White pixels represent
"light of the cities"

Region no.	% of white pixels compared to the total white pixels
1	20.4%
2	64.0%
3	4.9%
4	10.7%

Topological Descriptors

Use to describe holes and connected components of the region

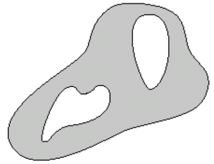


FIGURE 11.17 A region with two holes.

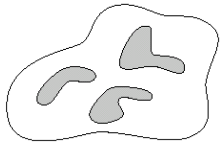


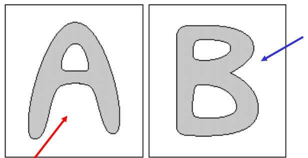
FIGURE 11.18 A region with three connected components.

Euler number (E):

$$E = C - H$$

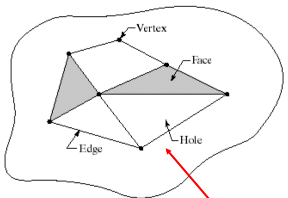
C = the number of connected components
H = the number of holes

Topological Descriptors (cont.)



E = -1

E = 0



E = -2

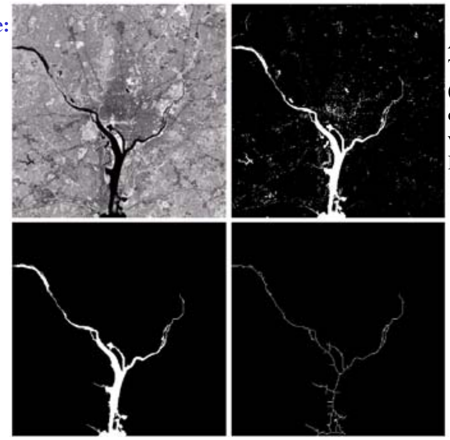
Euler Formula

$$V - Q + F = C - H = E$$

V = the number of vertices
Q = the number of edges
F = the number of faces

Example: Topological Descriptors

Original image:
Infrared image
Of Washington
D.C. area



After intensity
Thresholding
(1591 connected
components
with 39 holes)
Euler no. = 1552

The largest
connected
area
(8479 Pixels)
(Hudson river)

After thinning

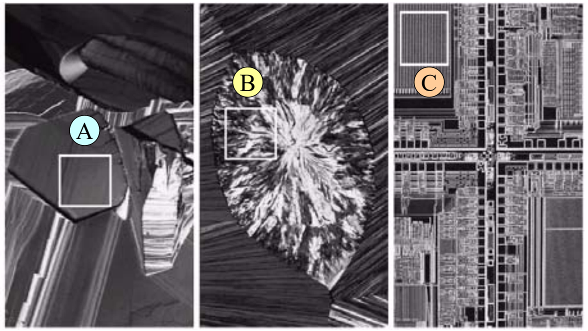
Texture Descriptors

- Texture: measures of properties such as smoothness, coarseness, and regularity
  1. *Statistical method*
     - Characterizes texture as smooth, coarse, grainy, etc.
  2. *Structural method*
     - Description based on regularly spaced parallel lines
  3. *Spectral method*
     - Fourier spectrum, detect global periodicity in an image by identifying high-energy, narrow peaks in the spectrum

Texture Descriptors

Purpose: to describe "texture" of the region.

Examples: optical microscope images:



Superconductor (smooth texture) | Cholesterol (coarse texture) | Microprocessor (regular texture)

Statistical Approaches for Texture Descriptors

We can use statistical moments computed from an image histogram:

$$\mu_n(z) = \sum_{i=0}^{K-1} (z_i - m)^n p(z_i)$$

z = intensity

$p(z)$ = PDF or histogram of z

where

$$m = \sum_{i=0}^{K-1} z_i p(z_i)$$

Example: The 2nd moment = variance → measure "smoothness"
The 3rd moment → measure "skewness"
The 4th moment → measure "uniformity" (flatness)

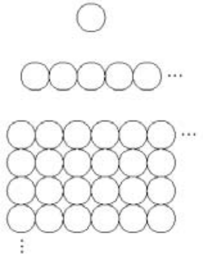
	Texture	Mean	Standard deviation	R (normalized)	Third moment	Uniformity	Entropy
A	Smooth	82.64	11.79	0.002	−0.105	0.026	5.434
B	Coarse	143.56	74.63	0.079	−0.151	0.005	7.783
C	Regular	99.72	33.73	0.017	0.750	0.013	6.674

Structural Approach for Texture Descriptor

- A rule of the form S → *aS* , "*circles to the right*"
- Add new rules : S → *b*A, A → *c*A, A → *c*, A → *b*S, S → *a*
- *b* means "*circle down*" and *c* means "*circle to the left*."
- We can now generate a string of the form *aabccbaa* that corresponds to a 3 X 3 **matrix of circles.**

a
b
c

FIGURE 11.23
(a) Texture primitive.
(b) Pattern generated by the rule $S \rightarrow aS$.
(c) 2-D texture pattern generated by this and other rules.



Fourier Approach for Texture Descriptor

- Fourier spectrum is ideally suited for describing the *directionality* of periodic or almost periodic 2-D patterns in an image.

- These *global texture patterns*, although easily distinguishable as concentrations of high-energy bursts in the spectrum,
    - generally are quite difficult to detect with *spatial* methods because of the local nature of these techniques.

    1. Prominent peaks in the spectrum give the *principal direction* of the texture patterns.
    2. The location of the peaks in the frequency plane gives the fundamental *spatial period* of the patterns
    3. Eliminating any periodic components via filtering leaves *nonperiodic image elements*, which can then be described by statistical techniques.

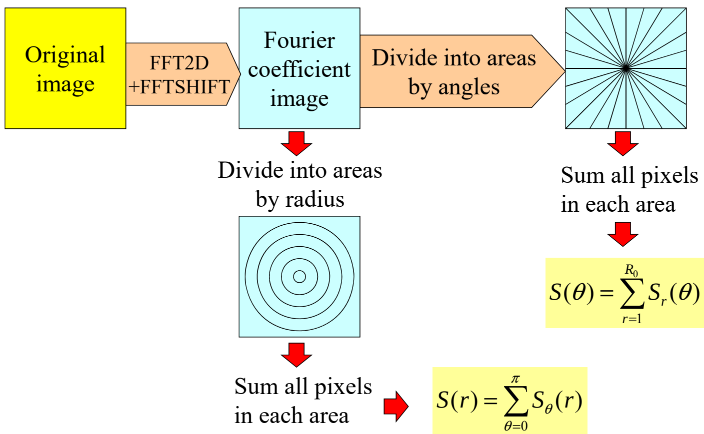
Fourier Approach for Texture Descriptor

- Detection and interpretation of the spectrum features just mentioned are simplified by expressing the spectrum in
    - polar coordinates to yield a function $S(r, \theta)$, where S is the spectrum function and r and θ are the variables in this coordinate system.

- For each direction θ,
    - $S(r, \theta)$ may be considered a 1-D function $S_\theta(r)$.
    - Similarly, for each frequency r, $S_r(\theta)$ is a 1-D function.
        - Analyzing $S_\theta(r)$ for a fixed value of θ yields the behavior of the spectrum (such as the presence of peaks) along a radial direction from the origin, whereas
        - analyzing $S_r(\theta)$ for a fixed value of r yields the behavior along a circle centered on the origin.

Fourier Approach for Texture Descriptor

Concept: convert 2D spectrum into 1D graphs



Fourier Approach for Texture Descriptor

- By varying these coordinates, we can generate two 1-D functions, $S(r)$ and $S(\theta)$, that constitute a spectral-energy description of texture for an entire image or region under consideration.
- Furthermore, descriptors of these functions themselves can be computed in order to characterize their behavior quantitatively.
- Descriptors typically used for this purpose are
    - the location of the highest value,
    - the mean and variance of both the amplitude and axial variations,
    - and the distance between the mean and the highest value of the function.

Fourier Approach for Texture Descriptor

Original image

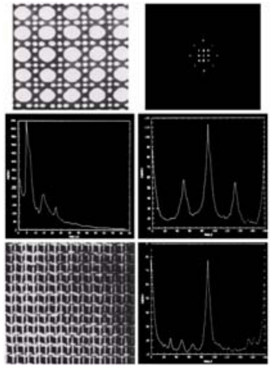
2D Spectrum (Fourier Tr.)

$S(r)$

$S(\theta)$

Another image

Another $S(\theta)$



Discriminating between the two texture patterns by analyzing their corresponding $S(\theta)$ waveforms would be straightforward

Moments of Two-D Functions

The moment of order $p+q$

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y) \qquad \bar{x} = \frac{m_{10}}{m_{00}} \qquad y = \frac{m_{01}}{m_{00}}$$

The central moments of order $p+q$

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y)$$

$$\mu_{00} = m_{00} \qquad \mu_{01} = \mu_{10} = 0$$

$$\mu_{11} = m_{11} - \bar{x}m_{01} = m_{11} - \bar{y}m_{10}$$

$$\mu_{20} = m_{20} - \bar{x}m_{10} \qquad \mu_{02} = m_{02} - \bar{y}m_{01}$$

$$\mu_{21} = m_{21} - 2\bar{x}m_{11} - \bar{y}m_{20} + 2\bar{x}^2 m_{01} \qquad \mu_{30} = m_{30} - 3\bar{x}m_{20} + 2\bar{x}^2 m_{10}$$

$$\mu_{12} = m_{12} - 2\bar{y}m_{11} - \bar{x}m_{02} + 2\bar{y}^2 m_{10} \qquad \mu_{03} = m_{03} - 3\bar{y}m_{02} + 2\bar{y}^2 m_{01}$$

Invariant Moments of Two-D Functions

The normalized central moments of order $p+q$

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \qquad \text{where} \quad \gamma = \frac{p+q}{2} + 1$$

Invariant moments: independent of ***rotation***, ***translation***, ***scaling***, and ***reflection***

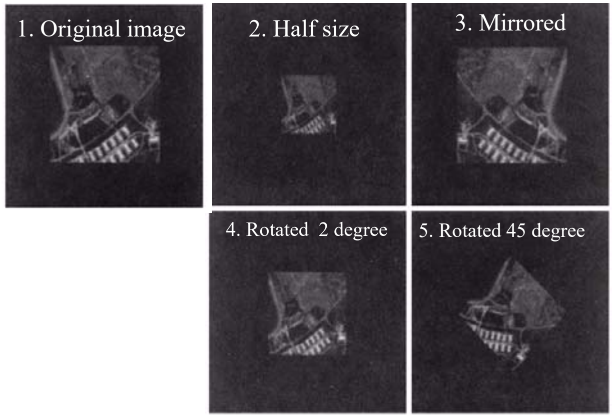
$$\phi_1 = \eta_{20} + \eta_{02} \qquad \phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \qquad \phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]$$

$$\phi_6 = (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

Example: Invariant Moments of Two-D Functions



Example: Invariant Moments of Two-D Functions

Invariant moments of images in the previous slide

Invariant (Log)	Original	Half Size	Mirrored	Rotated 2°	Rotated 45°
ϕ_1	6.249	6.226	6.919	6.253	6.318
ϕ_2	17.180	16.954	19.955	17.270	16.803
ϕ_3	22.655	23.531	26.689	22.836	19.724
ϕ_4	22.919	24.236	26.901	23.130	20.437
ϕ_5	45.749	48.349	53.724	46.136	40.525
ϕ_6	31.830	32.916	37.134	32.068	29.315
ϕ_7	45.589	48.343	53.590	46.017	40.470

Invariant moments are independent of rotation, translation, scaling, and reflection

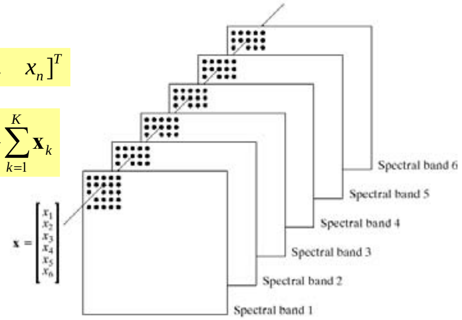
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Principal Components for Description

Purpose: to reduce dimensionality of a vector image while maintaining information as much as possible.

Let $\mathbf{x} = [x_1 \quad x_2 \quad \dots \quad x_n]^T$

Mean: $\mathbf{m}_x = E\{\mathbf{x}\} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k$



Covariance matrix $\mathbf{C}_x = E\{(\mathbf{x}-\mathbf{m}_x)(\mathbf{x}-\mathbf{m}_x)^T\} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k\mathbf{x}_k^T - \mathbf{m}_x\mathbf{m}_x^T$

Covariance matrix

Covariance matrix

$$\mathbf{C}_x = E\{(\mathbf{x}-\mathbf{m}_x)(\mathbf{x}-\mathbf{m}_x)^T\} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k\mathbf{x}_k^T - \mathbf{m}_x\mathbf{m}_x^T$$

- Element c_{ii} of $\mathbf{C}_x$: the variance of x_i .
- Element c_{ij} of $\mathbf{C}_x$, is the covariance between elements x_i and x_j .
- $\mathbf{C}_x$ is real and symmetric → can find orthonormal eigenvectors.
- If elements x_i and x_j arc uncorrelated, their covariance is zero, or $c_{ij} = c_{ji} = 0$.

Hotelling transformation

Let $\mathbf{y} = \mathbf{A}(\mathbf{x}-\mathbf{m}_x)$ ← *Hotelling transform*

Where $\mathbf{A}$ is created from eigenvectors of $\mathbf{C}_x$ as follows
- Row 1 contain the 1st *eigenvector* with the largest *eigenvalue*.
- Row 2 contain the 2nd *eigenvector* with the 2nd largest *eigenvalue*.

….

Then we get

$\mathbf{m}_y = E\{\mathbf{y}\} = 0$

$\mathbf{C}_y = \mathbf{A}\mathbf{C}_x\mathbf{A}^T$

and

$$\mathbf{C}_y = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & \lambda_1 \end{bmatrix}$$

Then elements of $\mathbf{y} = \mathbf{A}(\mathbf{x}-\mathbf{m}_x)$ are uncorrelated.
The component of $\mathbf{y}$ with the largest λ is called the ***principal component***.

Principal Components Transform

- $\mathbf{C}_x$ and $\mathbf{C}_y$ have the same eigenvalues and eigenvectors
- $\mathbf{x}$ can be reconstructed from $\mathbf{y}$:

$$\mathbf{x} = \mathbf{A}^T\mathbf{y} + \mathbf{m}_x$$

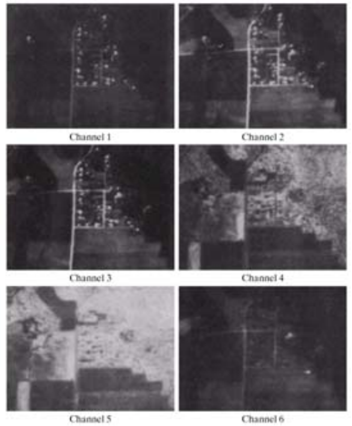
- Instead of using all the eigenvectors of $\mathbf{C}_x$, we form matrix $\mathbf{A}$, from the k eigenvectors corresponding to the k largest eigenvalues, yielding a transformation matrix of order k x n.
- The $\mathbf{y}$ vectors would then be k dimension and the reconstruction will be

$$\hat{\mathbf{x}} = \mathbf{A}_k^T\mathbf{y} + \mathbf{m}_x$$

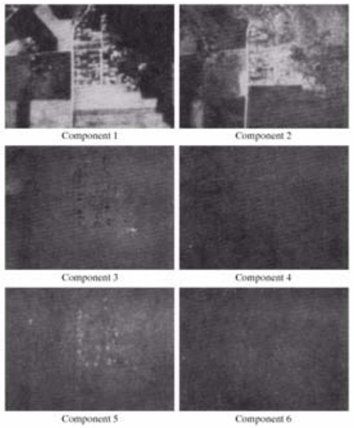
Example: Principal Components

6 spectral images $\mathbf{x}$ from an airborne multispectral scanner.

Channel	Wavelength band (microns)
1	0.40–0.44
2	0.62–0.66
3	0.66–0.72
4	0.80–1.00
5	1.00–1.40
6	2.00–2.60



Example: Principal Components (cont.)

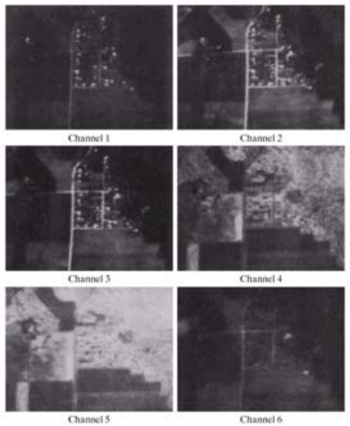


$$\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{m}_x)$$

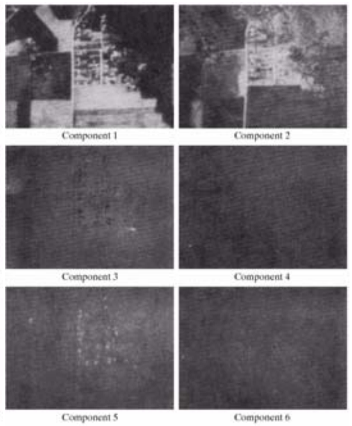
- Component 1 & 2 account for 94% of the total variance
- The other four principal-component images have low contrast
- Instead of storing all, only 1 & 2 are stored.

Component	λ
1	3210
2	931.4
3	118.5
4	83.88
5	64.00
6	13.40

Comparison: Principal Components (cont.)



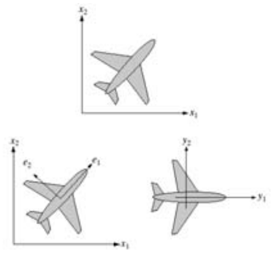
Original image



After Hotelling transform

Principal Components for Describing Boundaries and Regions

- In a single image, each pixel is a 2-D vector of $\mathbf{x}=(a, b)^T$.
- $\mathbf{y}= \mathbf{A}(\mathbf{x}-\mathbf{m}_x)$ establishes a new coordinate system
  - whose origin is at the centroid of the population (mean vector) and
  - whose axes are in the direction of the eigenvectors of $\mathbf{C}_x$.
  - A rotation transformation that aligns the data with the eigenvectors



a
b c

FIGURE 11.29 (a) An object. (b) Eigenvectors. (c) Object rotated by using Eq. (11.4-6). The net effect is to align the object along its eigen axes.

Aligiling a 2-D object with its principal eigenvectors

- Description should be as independent as possible to
  1. *variations* in size,
  2. *translation* and
  3. *Rotation*

- Removing the effects of rotation :
  - by *aligning* the object with its principal axes.

- The *eigenvalues* are the variances along the eigen axes and
  - can be used for size normalization.

- The effects of translation are accounted for
  - by centering the object about its *mean*.

Relational Descriptors

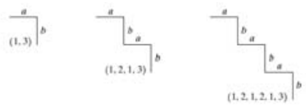
a b

FIGURE 11.30
(a) A simple staircase structure.
(b) Coded structure.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Relational Descriptors

FIGURE 11.31
Sample derivations for the rules $S \rightarrow aA$, $A \rightarrow bS$, and $A \rightarrow b$.



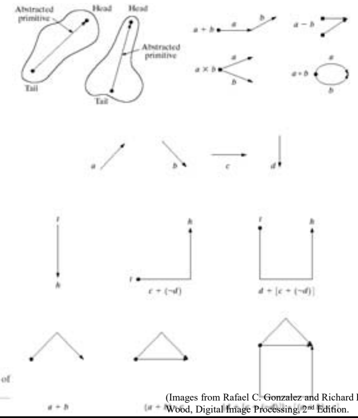
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Relational Descriptors

FIGURE 11.32 Coding a region boundary with directed line segments.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Relational Descriptors



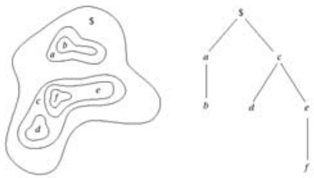
a b
c
d

FIGURE 11.33 (a) Abstracted primitives. (b) Operations among primitives. (c) A set of specific primitives. (d) Steps in building a structure.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Relational Descriptors

FIGURE 11.34 A simple tree with root $ and frontier *xy*.



a b

FIGURE 11.35 (a) A simple composite region. (b) Tree representation obtained by using the relationship "inside of."

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.