IMAGE PROCESSING

presented to:

DR. Ayman Soliman

Presented by [G10]:

Hend Samir Abdullah

Yassmin Mohamed Gouda

❑Content:

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 memor
  - ➢ Faster to processed

Feature Extraction

After an image has been segmented into regions or their boundaries using methods such as:

- Segmentation by region growing and by region splitting and merging
- Region segmentation using clustering and superpixels
- Region segmentation using gragh cuts
- Segmentation using morphological watersheds

the resulting sets of segmented pixels usually have to be converted into a form suitable for further computer processing. Typically, the step after segmentation **is feature extraction**

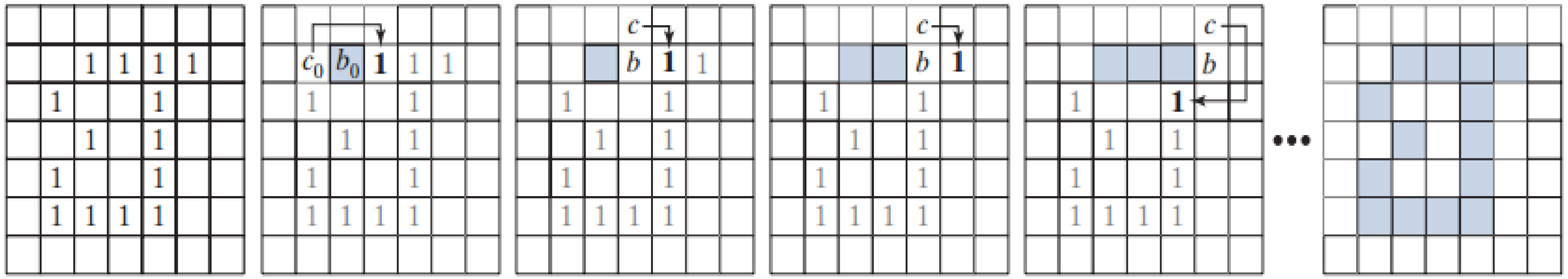
Feature extraction categories

For example, we might detect corners in a region boundary, and describe those corners by their orientation and location, both of which are quantitative attributes.

- ❑ **Feature detection:** refers to finding the features in an image, region, or boundary.

- ❑ **Feature description:** assigns quantitative attributes to the detected features.

Boundary-following algorithm



a b c d e f

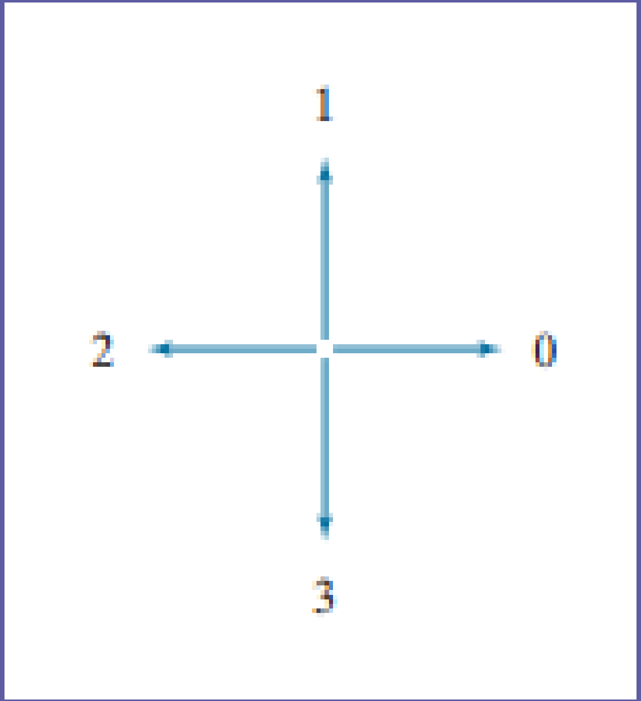
whose output is an *ordered* sequence of points. We assume (1) that we are working with binary images in which object and background points are labeled 1 and 0, respectively; and (2) that images are padded with a border of 0's to eliminate the possibility of an object merging with the image border.

Chain codes

Chain codes are used to represent a boundary by a connected sequence of straightline segments of specified length and direction. We assume in this section that all curves are closed, simple curves (i.e., curves that are closed and not self intersecting).

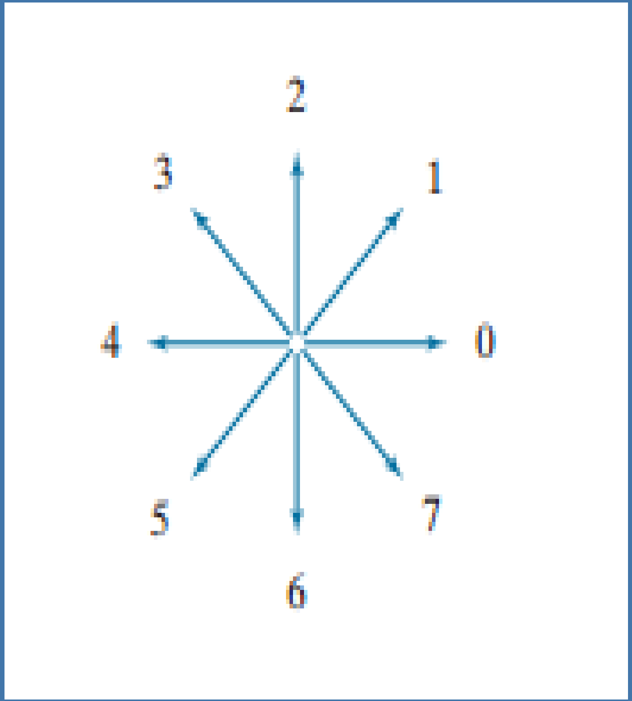
Freeman Chain Codes

4-connectivity



a chain code representation is based on 4- or 8-connectivity of the segments. The direction of each segment is coded by using a numbering scheme, as in the Fig. A boundary code formed as a sequence of such directional numbers is referred to as a Freeman chain code.

8-connectivity



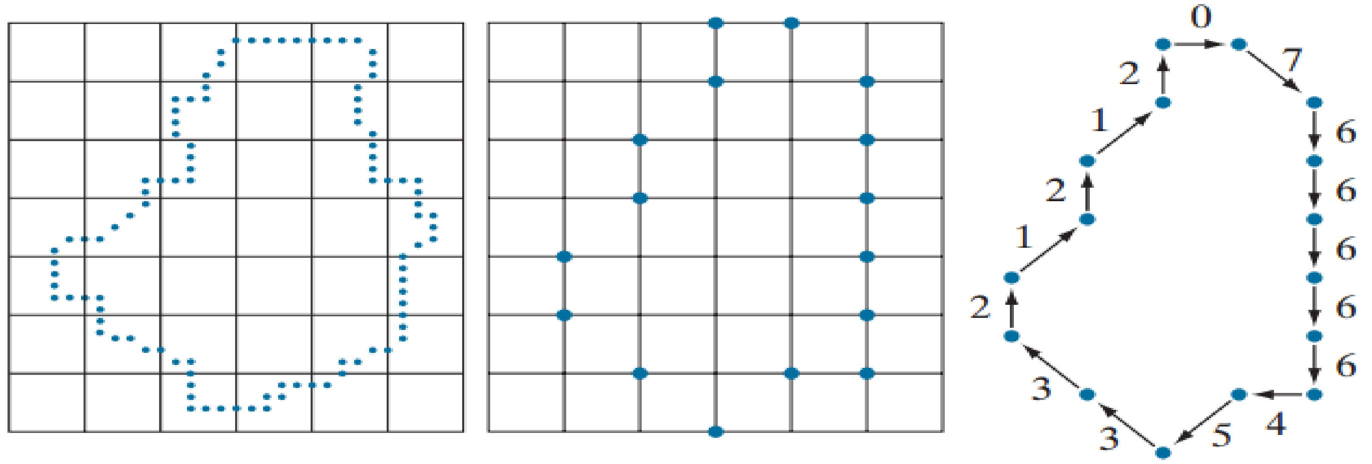
Chain code problems

a chain code could be generated by following a boundary in, say, a clockwise direction and assigning a direction to the segments connecting every pair of pixels. This level of detail generally is not used for two principal reasons:

- ➢ The resulting chain would be quite long
- ➢ any small disturbances along the boundary due to noise or imperfect segmentation would cause changes in the code that may not be related to the principal shape features of the boundary.

Resampling

An approach used to address these problems is to resample the boundary by selecting a larger grid spacing, as in Fig.(a). Then, as the boundary is traversed, a boundary point is assigned to a node of the coarser grid, depending on the proximity of the original boundary point to that node, as in Fig.(b). The resampled boundary obtained in this way can be represented by a 4- or 8-code. Figure(c) shows the coarser boundary points represented by an 8-directional chain code.



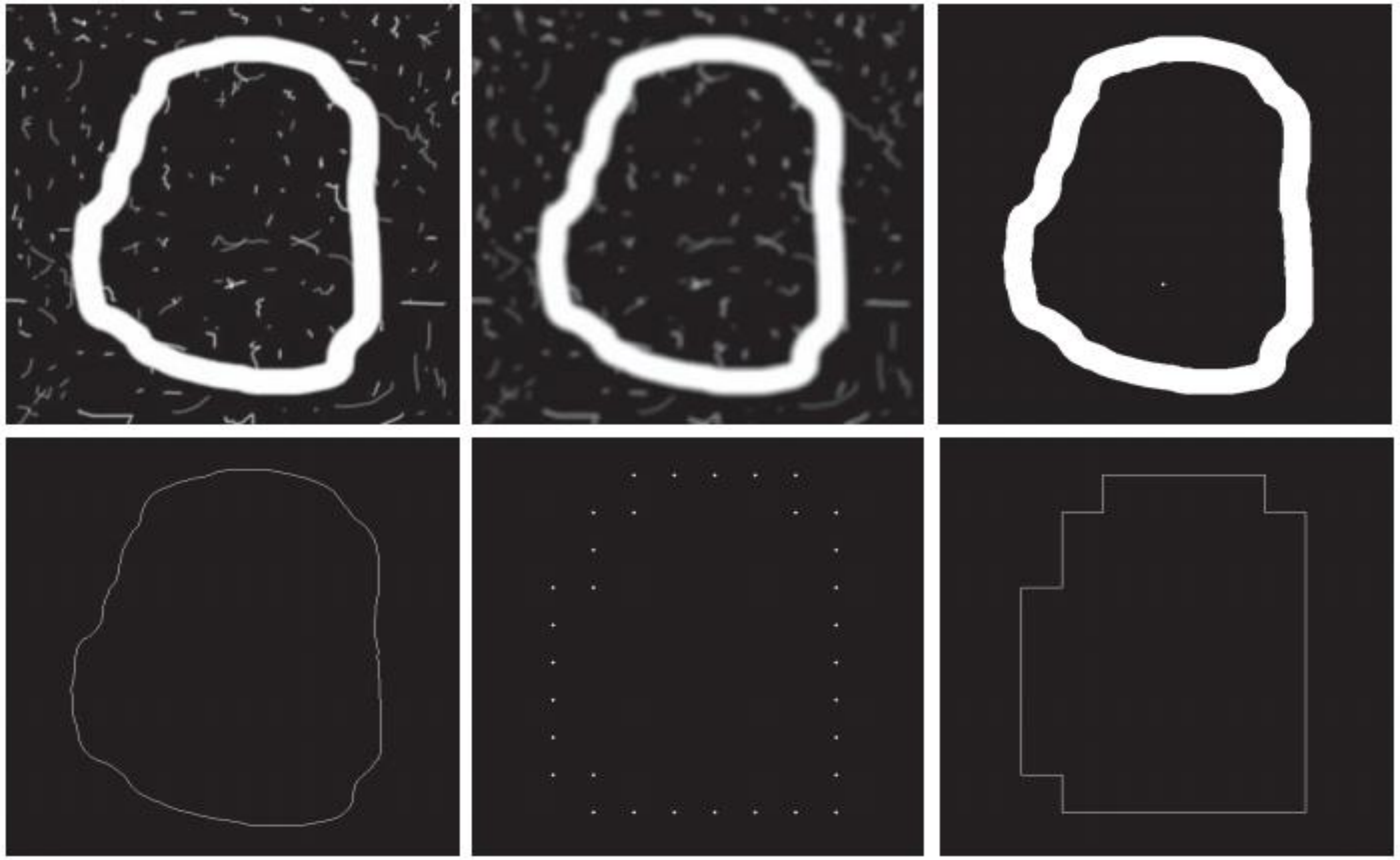
Normalization for chain codes

The numerical value of a chain code depends on the starting point. However, the code can be normalized:

- ➢ with respect to the starting point by a straightforward procedure: We simply treat the chain code as a circular sequence of direction numbers and redefine the starting point so that the resulting sequence of numbers forms an integer of minimum magnitude.
- ➢ We can normalize also for rotation by using the first difference of the chain code instead of the code itself. This difference is obtained by counting the number of direction changes (in a counterclockwise direction).

Example: 101003333222 normalized to 003333222101

EXAMPLE: Freeman chain code and some of its variations



a b c
d e f

(a) Noisy image of size 570 570 · pixels. (b) Image smoothed with a 9*9 · box kernel. (c) Smoothed
image, thresholded using Otsu's method. (d) Longest outer boundary of (c). (e) Subsampled boundary (the points
are shown enlarged for clarity). (f) Connected points from (e)

EXAMPLE: Freeman chain code and some of its variations (cont.)

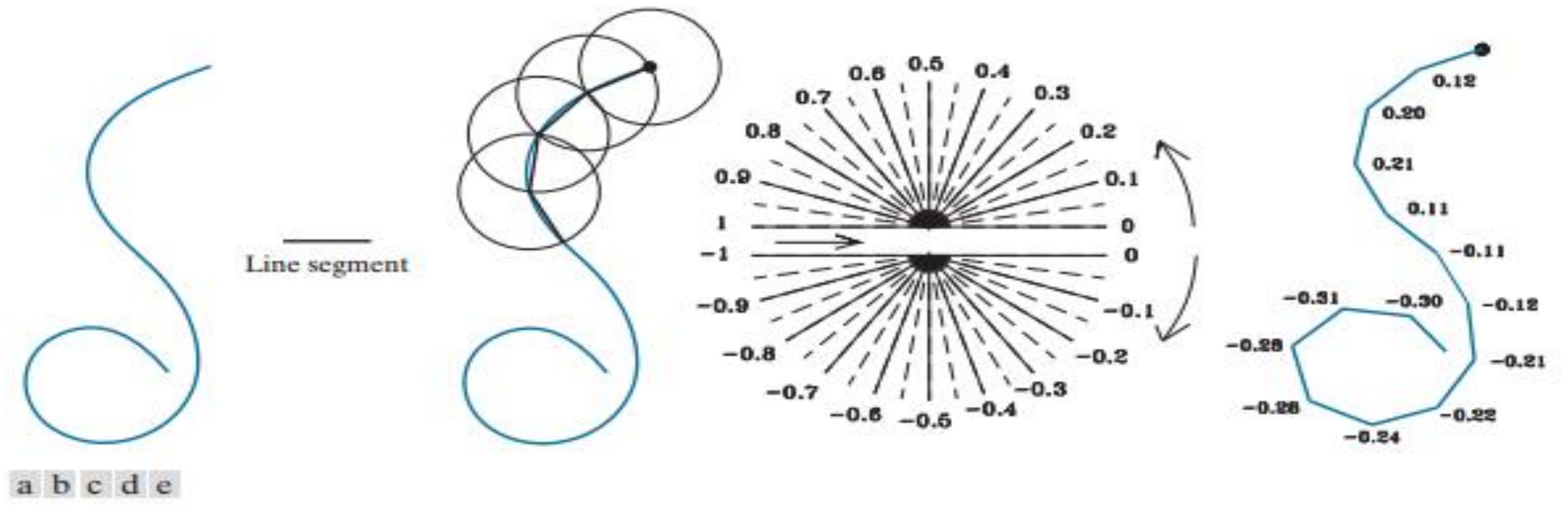
The starting point of the boundary is at coordinates (2, 5) in the subsampled.

- The 8-directional Freeman chain code of the simplified boundary is
  0 0 0 0 6 0 6 6 6 6 6 6 6 2 6 4 4 4 4 4 4 2 4 2 2 2 2 2 0 2 2 0 2

- The integer of minimum magnitude of the code happens in this case to be the same as the chain code:
  0 0 0 0 6 0 6 6 6 6 6 6 6 2 6 4 4 4 4 4 4 2 4 2 2 2 2 2 0 2 2 0 2
- The first difference of the code is
  0 0 0 6 2 6 0 0 0 0 0 0 6 0 0 0 0 0 6 2 6 0 0 0 0 6 2 0 6 2 6

Slope Chain Codes

Using Freeman chain codes generally requires resampling a boundary to smooth
small variations, a process that implies defining a grid and subsequently assigning
all boundary points to their closest neighbors in the grid. An alternative to this
approach is to use *slope chain codes* (SCCs) (Bribiesca [1992, 2013]). The SCC of a
2-D curve is obtained by placing straight-line segments of equal length around the
curve, with the end points of the segments touching the curve.

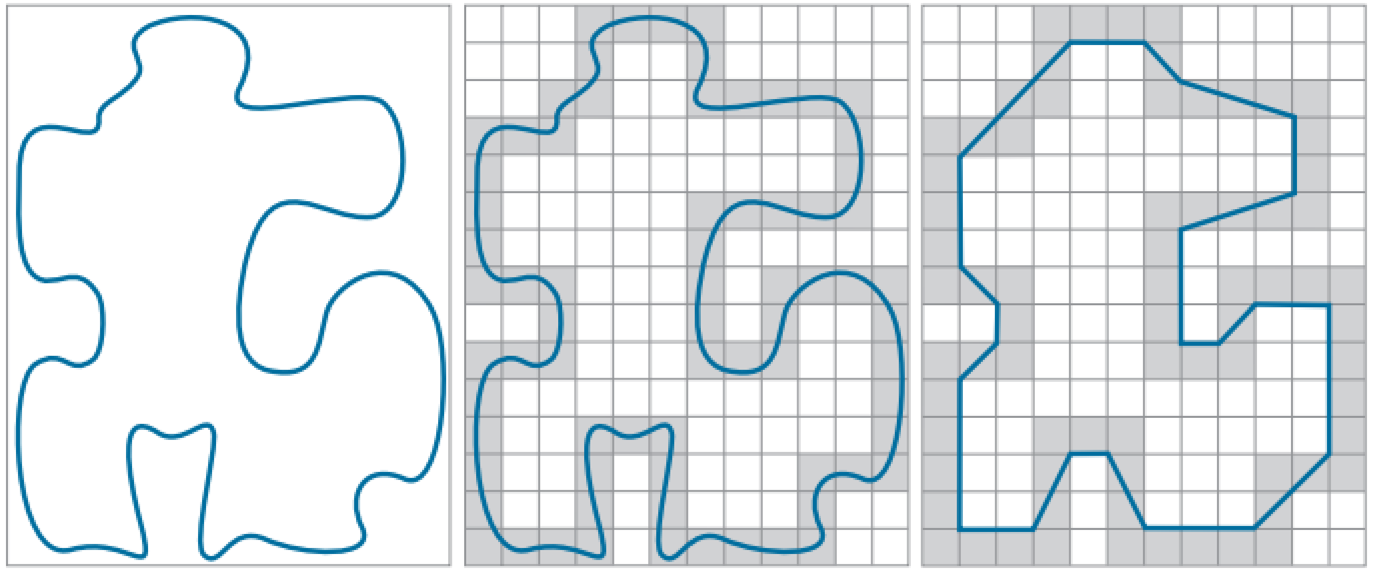


The sequence of slope changes is the chain that defines the SCC approximation to the original curve. For example, the code for the curve in Fig. (e) is 0 12 . , 0 20 . , 0 21 . , 0 11 . , –0 1 . , 1 –0 1 . , 2 –0 2 . , 1 –0 2 . , 2 –0 2 . , 4 –0 2 . , 8 –0 2 . , 8 –0 3 . , 1 –0 3 . . 0 The accuracy of the slope changes defined in Fig. (d) is 10–2, resulting in an "alphabet" of 199 possible symbols (slope changes).The accuracy can be changed, of course. For instance, and accuracy of 10–1 produces an alphabet of 19 symbols

(a) An open curve. (b) A straight-line segment. (c) Traversing the curve using circumferences to determine slope changes; the dot is the origin (starting point). (d) Range of slope changes in the open interval (,) –1 1
(the arrow in the center of the chart indicates direction of travel). There can be ten subintervals between the slope numbers shown.(e) Resulting coded curve showing its corresponding numerical sequence of slope changes

BOUNDARY APPROXIMATIONS USING MINIMUM-PERIMETER POLYGONS

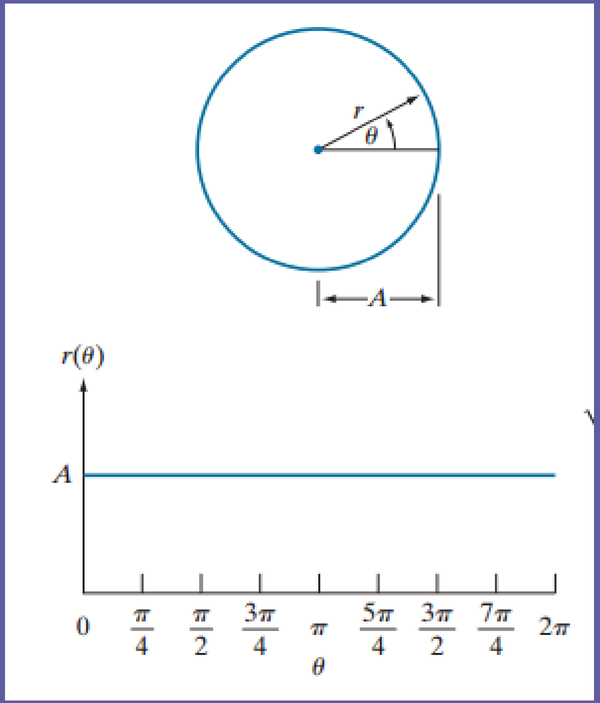
An intuitive approach for computing MPPs is to enclose a boundary [see Fig.(a)] by a set of concatenated cells.



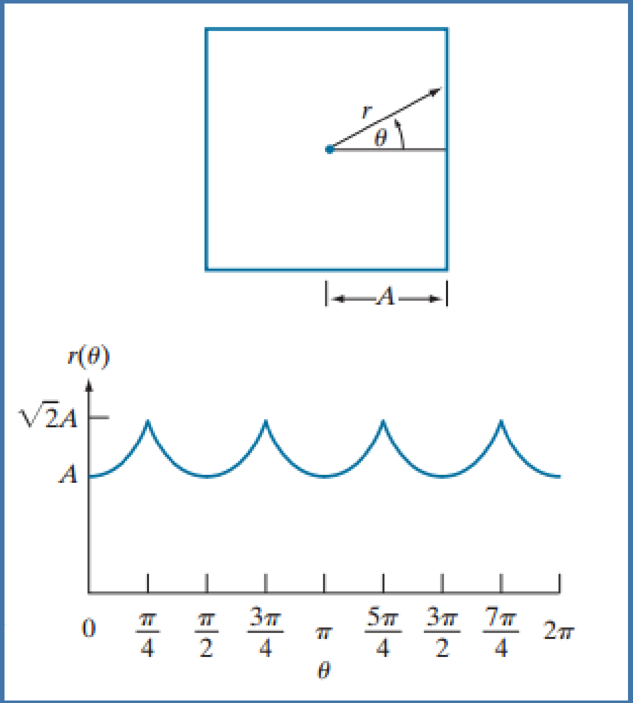
(a) An object boundary. (b) Boundary enclosed by cells (shaded). (c) Minimum-perimeter polygon obtained by allowing the boundary to shrink. The vertices of the polygon are created by the corners of the inner and outer walls of the gray region

SIGNATURES

A signature is a 1-D functional representation of a 2-D boundary and may be generated in various ways. One of the simplest is to plot the distance from the centroid to the boundary as a function of angle, as illustrated in Fig. 11.10. The basic idea of using signatures is to reduce the boundary representation to a 1-D function that presumably is easier to describe than the original 2-D boundary.



Distance-versusangle signatures.
In (a), *r*(θ) is constant. In (b), the signature consists of repetitions of the pattern
r (θ) =A secθ for 0<=θ<=π/4, and
r (θ) = csc for π/4 <=θ<=π/2



SOME BASIC BOUNDARY DESCRIPTORS

The length of a boundary is one of its simplest descriptors. The number of pixels along a boundary is an approximation of its length. For a chain-coded curve with unit spacing in both directions, the number of vertical and horizontal components plus 2 multiplied by the number of diagonal components gives its exact length. If the boundary is represented by a polygonal curve, the length is equal to the sum of the lengths of the polygonal segments.

$$diameter(B) = \max_{i,j}\left[D\left(p_i, p_j\right)\right]$$

$$length_m = \left[(x_2 - x_1)^2 + (y_2 - y_1)^2\right]^{1/2}$$

$$angle_m = \tan^{-1}\left[\frac{y_2 - y_1}{x_2 - x_1}\right]$$

Curvature of a boundary

The curvature of a boundary is defined as the rate of change of slope. In general, obtaining reliable measures of curvature at a point of a raw digital boundary is difficult because these boundaries tend to be locally “ragged.” Smoothing can help, but a more rugged measure of curvature is to use the difference between the slopes of adjacent boundary segments that have been represented as straight lines.

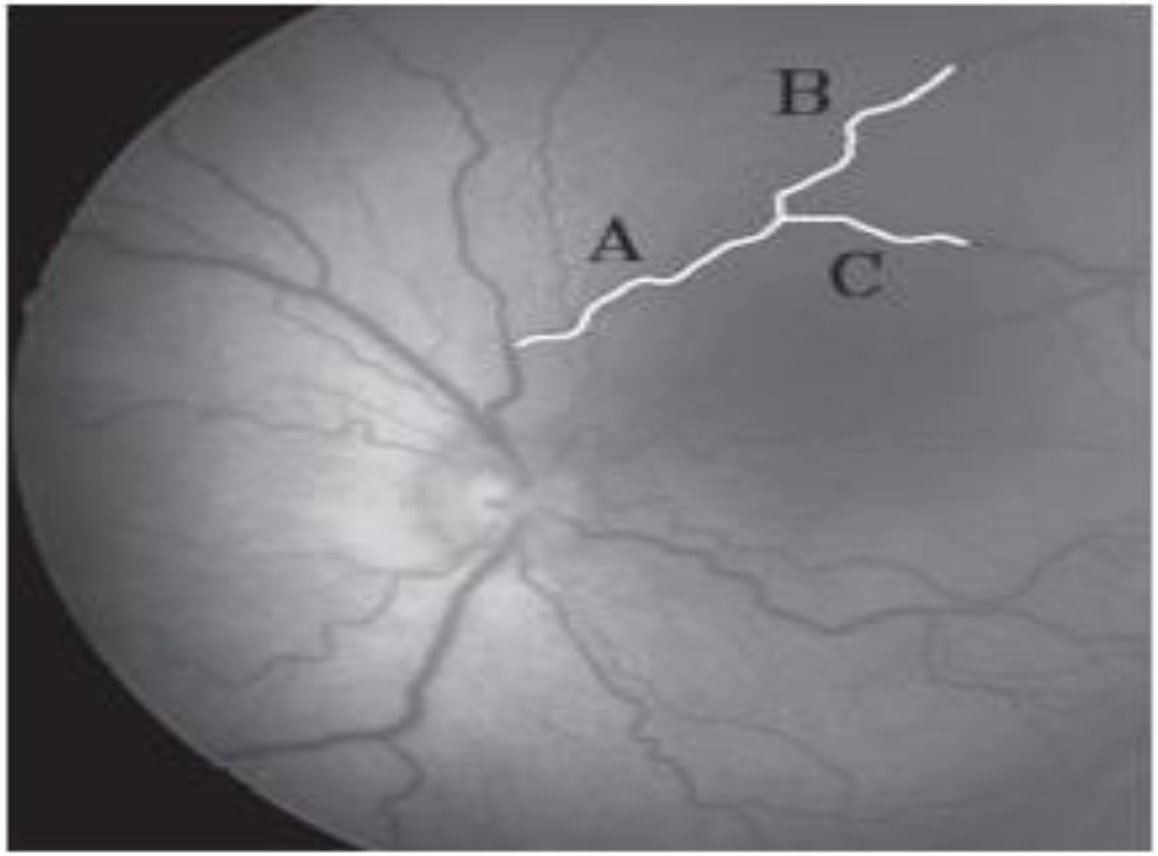
EXAMPLE: Using slope chain codes to describe tortuosity

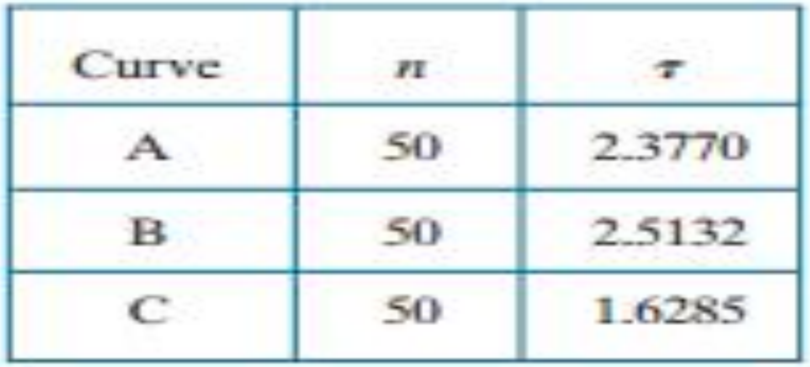
An important measures of blood vessel morphology is its tortuosity. This metric can assist in the computeraided diagnosis of Retinopathy of Prematurity (ROP), an eye disease that affects babies born prematurely (Bribiesca [2013]). ROP causes abnormal blood vessels to grow in the retina .This growth can cause the retina to detach from the back of the eye, potentially leading to blindness.

Figure (a) shows an image of the retina (called a *fundus* image) from a newborn baby. Ophthalmologists diagnose and make decisions about the initial treatment of ROP based on the appearance of retinal blood vessels. Dilatation and increased tortuosity of the retinal vessels are signs of highly probable ROP. Blood vessels denoted A, B, and C in the Fig were selected to demonstrate the discriminative potential of SCCs for quantifying tortuosity (each vessel shown is a long, thin *region*, not a line segment).

The border of each vessel was extracted and its length (number of pixels), P, was calculated. To make SCC comparisons meaningful, the three boundaries were normalized so that each would have the same number, m, of straight-line segments. The length, L, of the line segment was then computed as $L\ m\ P =$. It follows that the number of elements of each SCC is $m -$ 1. The tortuosity, t, of a curve represented by an SCC is defined as the sum of the absolute values of the chain elements, as noted in Eq.

The table in Fig. (b) shows values of t for vessels A, B, and C based on 51 straight-line segments (as noted above, $n\ m = -$ 1). The values of tortuosity are in agreement with our visual analysis of the three vessels, showing B as being slightly “busier” than A, and C as having the fewest twists and turns.





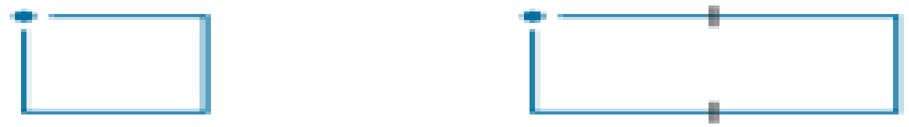
Curve	n	τ
A	50	2.3770
B	50	2.5132
C	50	1.6285

(a) Fundus image from a prematurely born baby with ROP.
(b) Tortuosity of vessels A, B, and C

Shape numbers

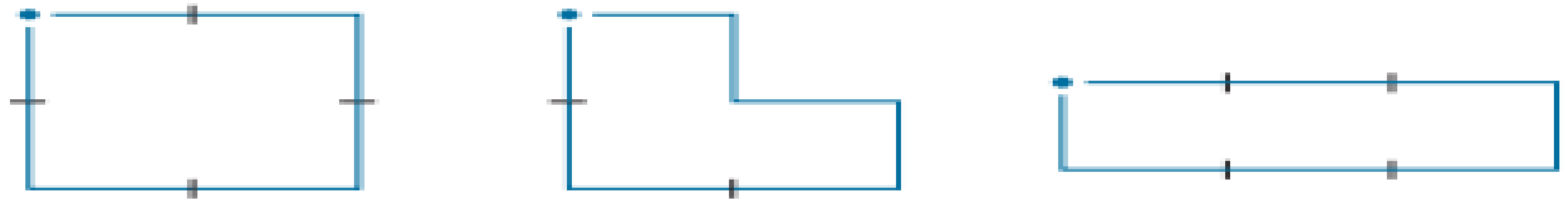
The shape number of a Freeman chain-coded boundary, based on the 4-directional code of Fig. 11.3(a), is defined as the first difference of smallest magnitude. The *order*, *n*, of a shape number is defined as the number of digits in its representation. Moreover, *n* is even for a closed boundary, and its value limits the number of possible different shapes

Order 4 | Order 6



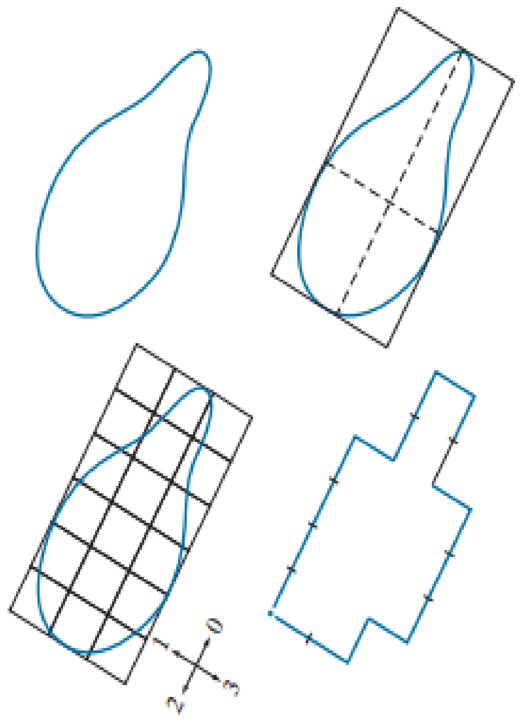
	Order 4	Order 6
Chain code:	0 3 2 1	0 0 3 2 2 1
Difference:	3 3 3 3	3 0 3 3 0 3
Shape no.:	3 3 3 3	0 3 3 0 3 3

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

EXAMPLE : Computing shape numbers



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

Difference: 3 0 0 0 3 1 0 3 3 0 1 3 0 0 3 1 3 0

Shape no.: 0 0 0 3 1 0 3 3 0 1 3 0 0 3 1 3 0 3

Suppose that $n = 18$ is specified for the boundary in Fig.(a). To obtain a shape number of this order we follow the steps just discussed. First, we find the basic rectangle, as shown in Fig.(b). Next we find the closest rectangle of order 18. It is a 3 6 · rectangle, requiring the subdivision of the basic rectangle shown in Fig.(c). The chain-code directions are aligned with the resulting grid. The final step is to obtain the chain code and use its first difference to compute the shape number, as shown in Fig.(d)

Fourier descriptors

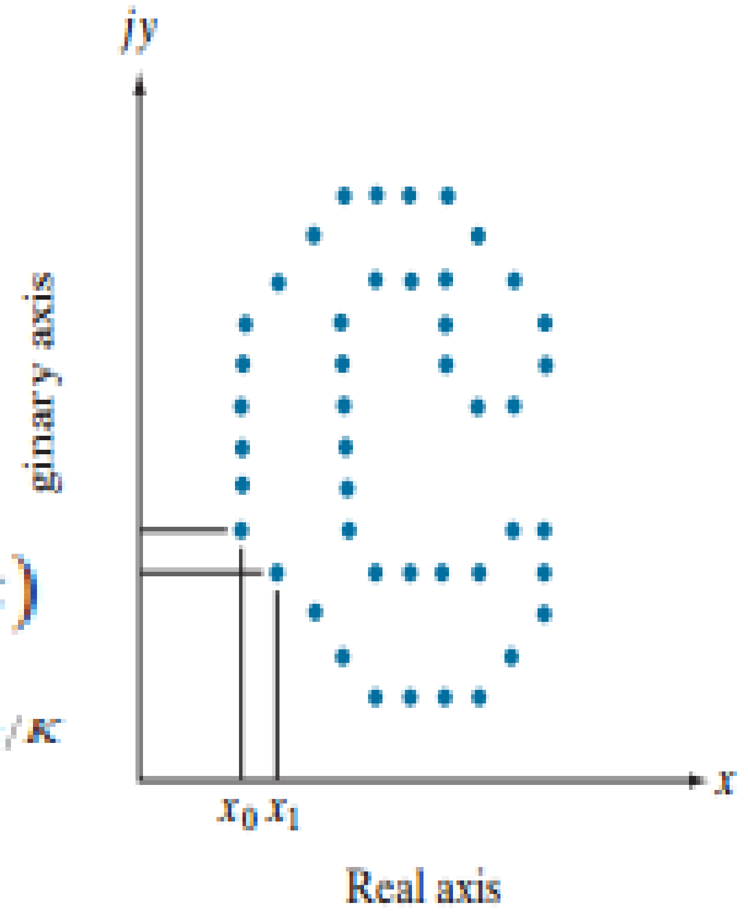
The figure shows a digital boundary in the *xy*-plane, consisting of *K* points. Starting at an arbitrary point (x_0, y_0) ,coordinate pairs (x_0, y_0), (x_1, y_1),....... are encountered in traversing the boundary, say, in the counterclockwise direction. These coordinates can be expressed in the form of x(k)=x_kUsing this notation, the boundary itself can be represented as the sequence of coordinates s(k)=[x(k),y(k)] for k=0,1,2,….,k-1Moreover, each coordinate pair can be treated as a complex number so that :

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

$$a(u) = \sum_{k=0}^{K-1} s(k) e^{-j2\pi uk/K}$$

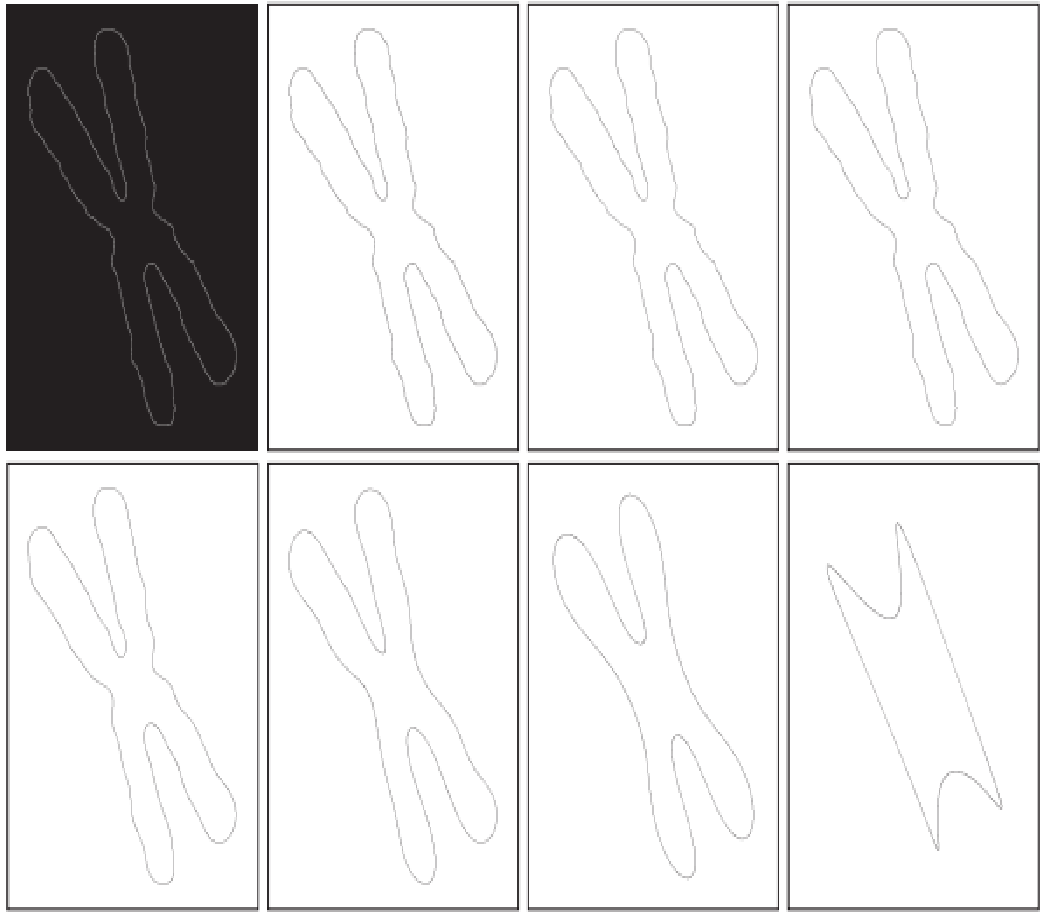
$$s(k) = \frac{1}{K} \sum_{u=0}^{K-1} a(u) e^{j2\pi uk/K}$$

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



EXAMPLE: Using Fourier descriptors

Figure (a) shows the boundary of a human chromosome, consisting of 2868 points. The corresponding 2868 Fourier descriptors were obtained using Eq. The objective of this example is to examine the effects of reconstructing the boundary using fewer Fourier descriptors. Figure (b) shows the boundary reconstructed using one-half of the 2868 descriptors in Eq. Observe that there is no perceptible difference between this boundary and the original. Figures(c) through (h) show the boundaries reconstructed with the number of Fourier descriptors being 10%, 5%, 2.5%, 1.25%, 0.63% and 0.28% of 2868, respectively. When rounded to the nearest even integer, these percentages are equal to 286, 144, 72, 36, 18, and 8 descriptors, respectively. The important point is that 18 descriptors, a mere six-tenths of one percent of the original 2868 descriptors, were sufficient to retain the principal shape features of the original boundary: four long protrusions and two deep bays. Figure (h), obtained with 8 descriptors, is unacceptable because the principal features are lost. Further reductions to 4 and 2 descriptors would result in an ellipse and a circle, respectively.



a b c d
e f g h

What is a Feature?

a **feature** is a piece of information about the content of an image;

typically about whether a certain region of the image has certain prope

Features include things like, points, edges, blobs, and corners.

For example, suppose you saw this feature?

With just two features, you were able to identify this object.

Computers follow a similar process when you run a feature detection algorithm to perform object recognition.

Region Descriptors :-

The area of a region is defined as the number of pixels in the region.

The perimeter of a region is the length of its boundary. When area and perimeter are used as descriptors, they generally make sense only when they are normalized .

❖ TOPOLOGICAL DESCRIPTORS:-

Topology is the study of properties of a figure that are unaffected by any deformation, provided that there is no tearing or joining of the figure (sometimes these are called rubber-sheet distortions)

For example, Fig. 1.1(a) shows a region with two holes. Obviously, a topological descriptor defined as the number of holes in the region will not be affected by a stretching or rotation transformation. However, the number of holes can change if t

a b

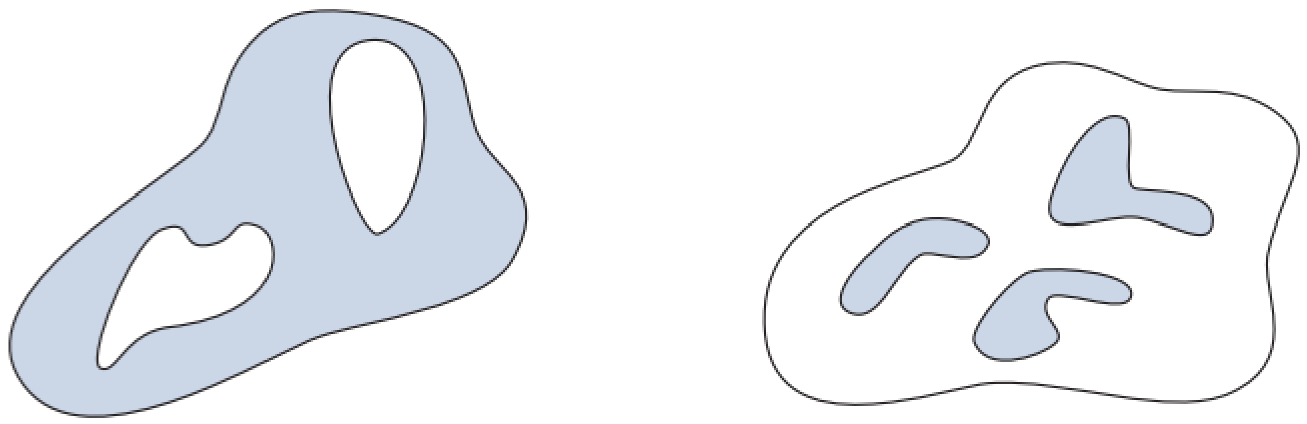


Figure 1.1

(a) A region with two holes.
(b) A region with three connected components.

Figure 1.2 shows a polygonal network. Classifying interior regions of such a network into faces and holes is often important. Denoting the number of vertices by V, the number of edges by Q, and the number of faces by F gives the following relationship, called the Euler formula:

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

The network in Fig. 1.2 has seven vertices, eleven edges, two faces, one connected region, and three holes; thus the Euler number is −2 . (i.e., $7 - 11 + 2 = 1 - 3 = -2$).

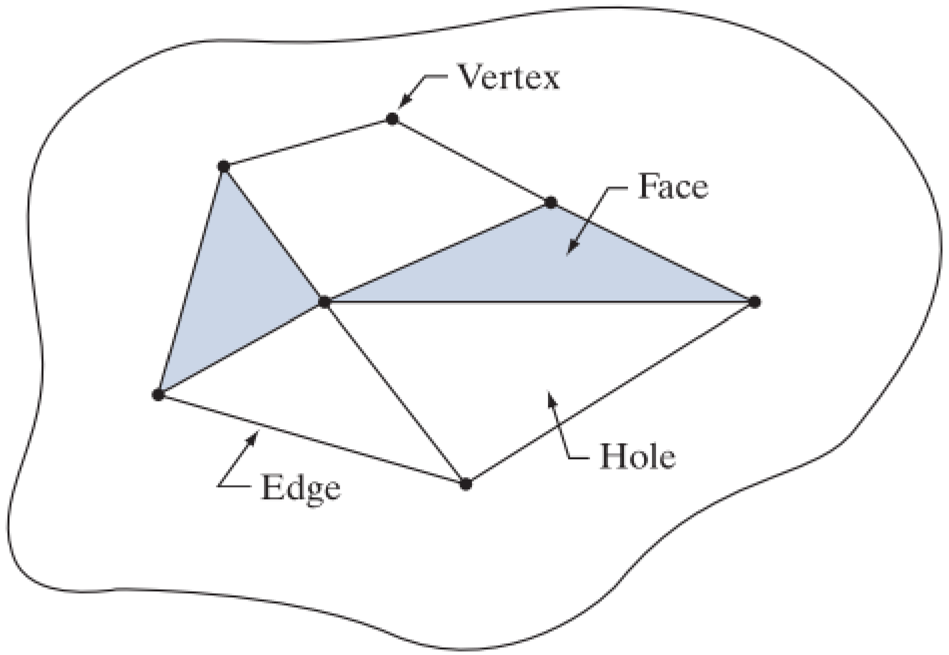
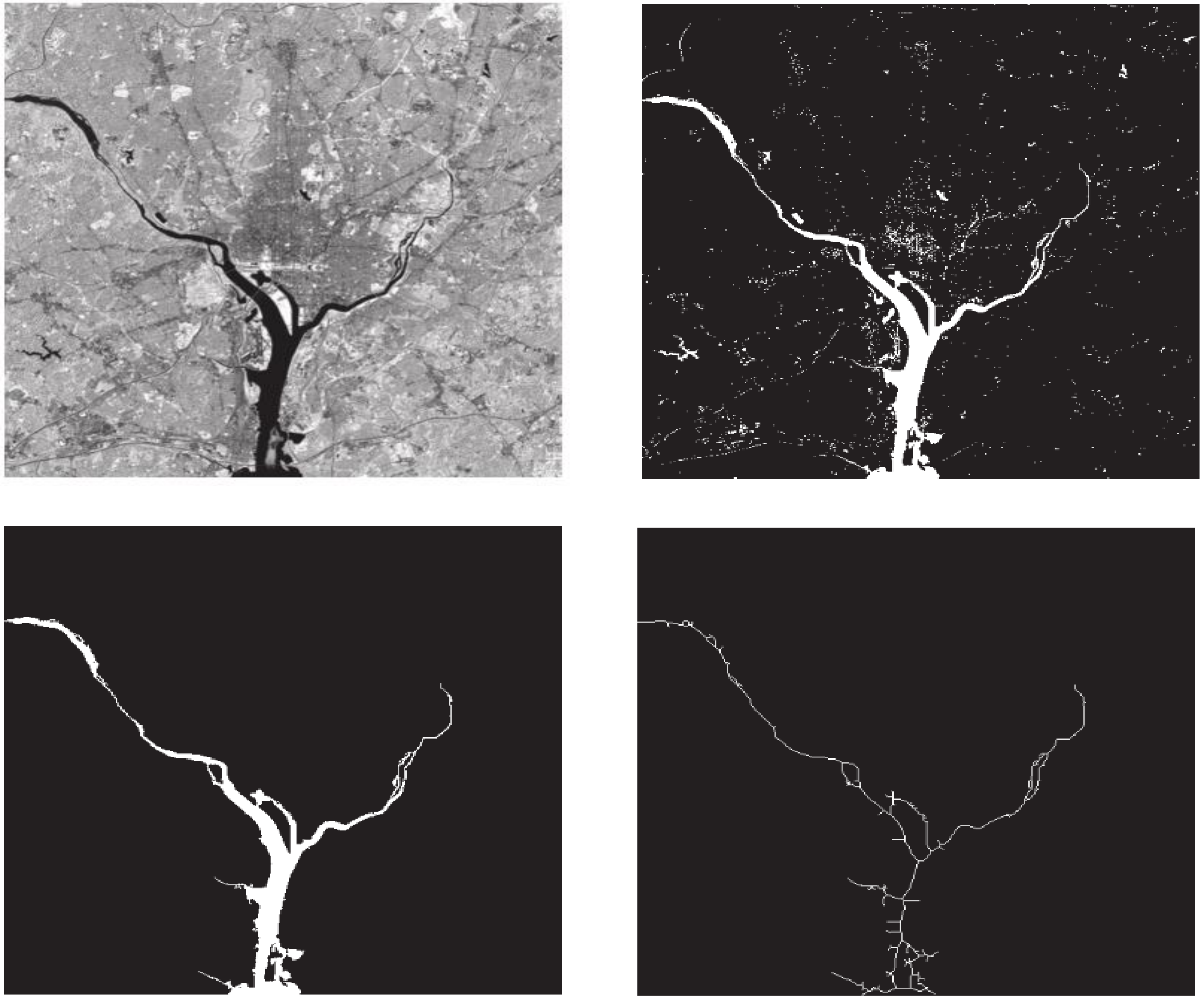


figure 1.2

A region containing a polygonal network.

- Figure 1.3(a) shows a 512 x 512 , 8-bit image of Washington, D.C. taken by a NASA LANDSAT satellite. This image is in the near infrared band. Suppose that we want to segment the river using only this image (as opposed to using several multispectral images, which would simplify the task, as you will see later in this chapter). Because the river is a dark, uniform region relative to the rest of the image, thresholding is an obvious approach to try. The result of thresholding the image with the highest possible threshold value before the river became a disconnected region is shown in Fig1.3(b).

- The threshold was selected manually to illustrate the point that it would be impossible in this case to segment the river by itself without other regions of the image also appearing in the thresholded result. The image in Fig. 1.3(b) has 1591 connected components (obtained using 8-connectivity) and its Euler number is 1552, from which we deduce that the number of holes is 39. Figure 1.3(c) shows the connected component with the largest number of pixels (8479). This is the desired result, which we already know cannot be segmented by itself from the image using a threshold. Note how clean this result is. The number of holes in the region defined by the connected component just found would give us the number of land masses within the river. If we wanted to perform measurements, like the length of each branch of the river, we could use the skeleton of the connected component [Fig. 1.3(d)] to do so.



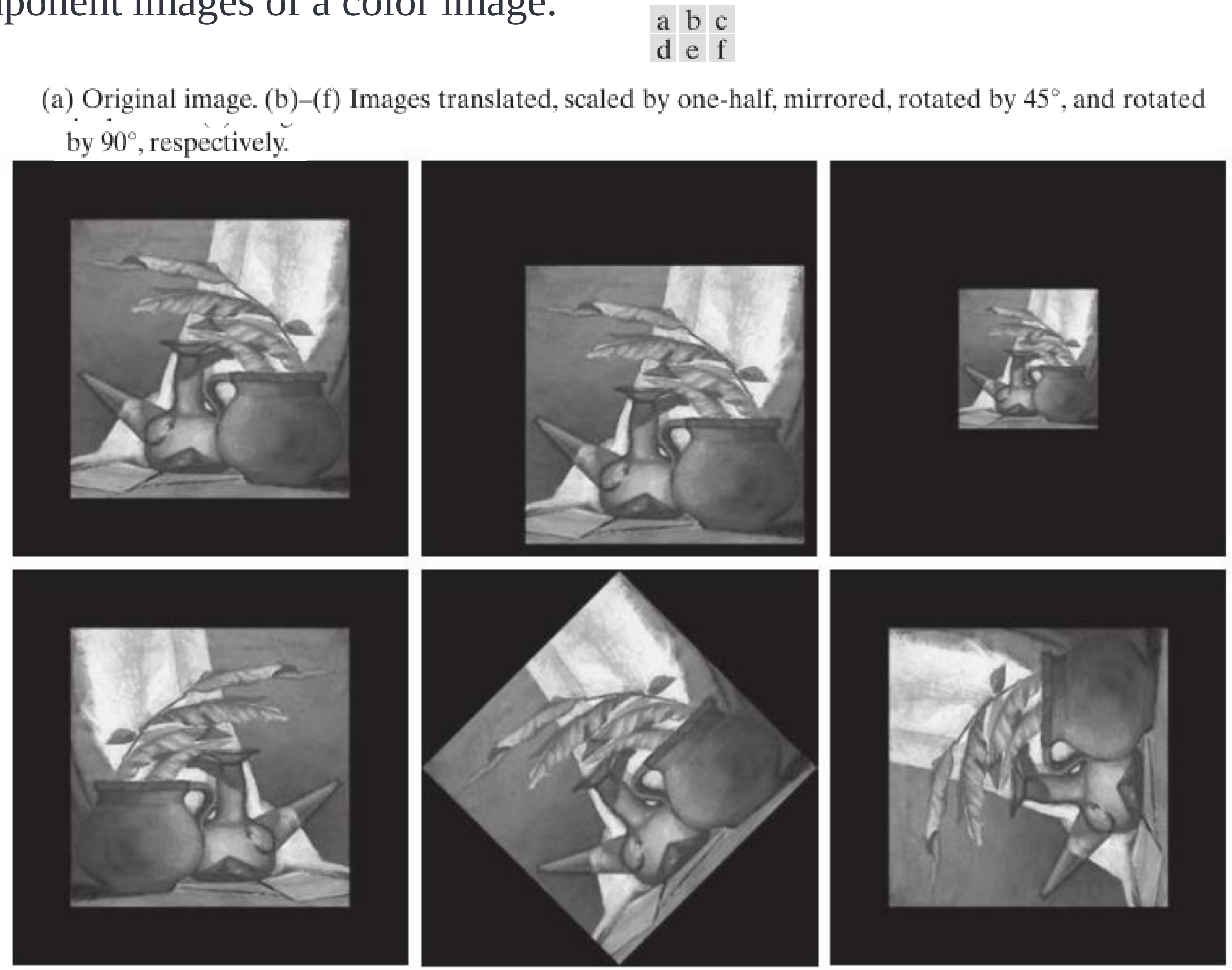
a b
c d

(a) Infrared image of the Washington, D.C. area. (b) Thresholded image. (c) The largest connected component of (b). (d) Skeleton of (c). (Original image courtesy of NASA.)

Principle component as feature descriptors :-

Suppose that we are given the three component images of a color image.

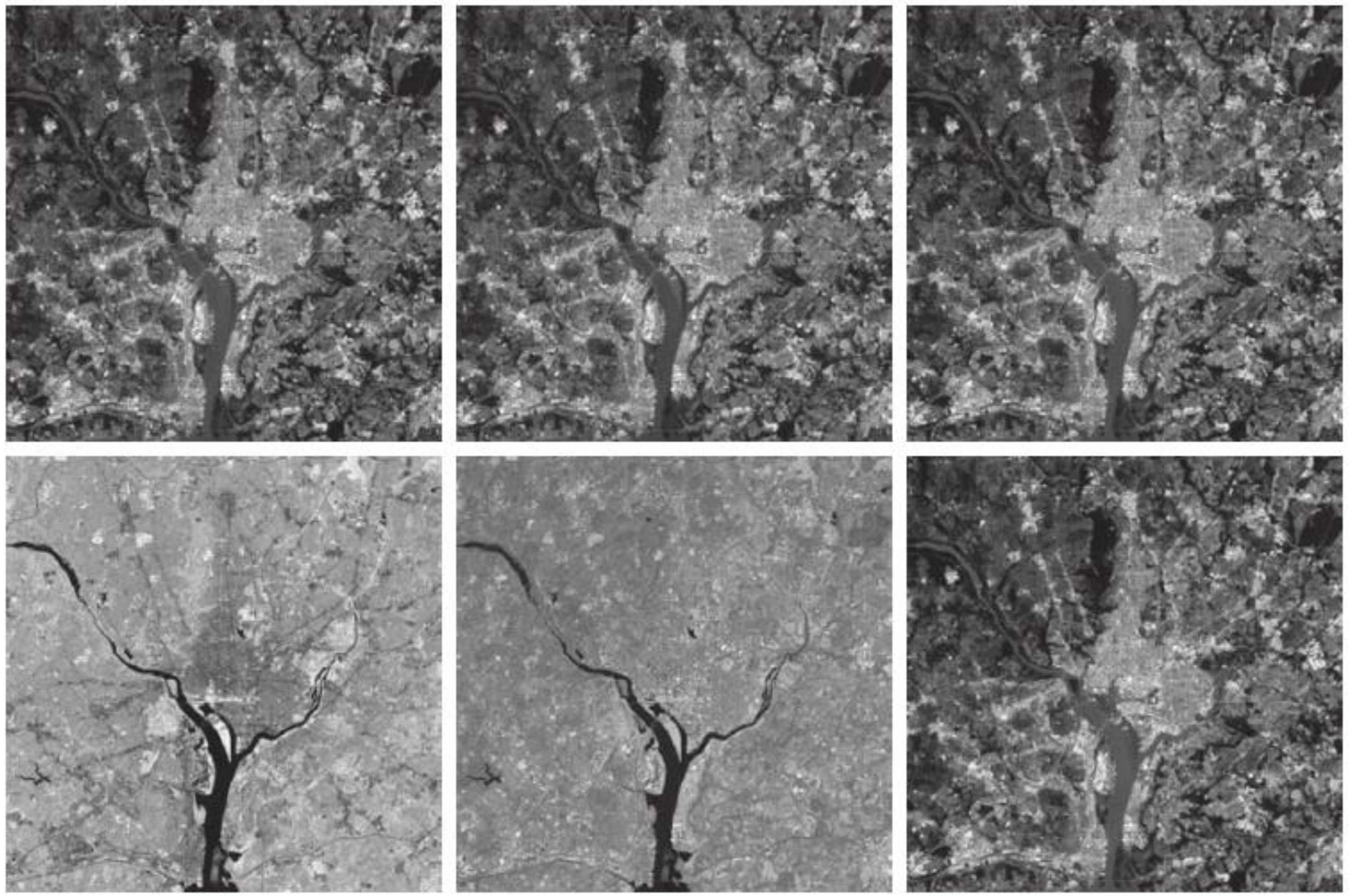
The three images can be treated as a unit by expressing each group of three corresponding pixels as a vector



(a) Original image. (b)–(f) Images translated, scaled by one-half, mirrored, rotated by 45°, and rotated by 90°, respectively.

Using principal components for image description:-

Figure shows six multispectral satellite images corresponding to six spectral bands: visible blue (450–520 nm), visible green (520–600 nm), visible red (630–690 nm), near infrared (760–900 nm), middle infrared (1550–1,750 nm), and thermal infrared (10,400–12,500 nm). The objective of this example is to illustrate how to use principal components as image features.



a b c
d e f

Multispectral images in the (a)visible blue, (b) visible green, (c) visible red, (d) near infrared, (e) middle infrared, and (f) thermal infrared bands. (Images courtesy of NASA.)

Organizing the images as in Fig.1.5 leads to the formation of a six-element vector x from each set of corresponding pixels in the images, as discussed earlier in this section.
The images in this example are of size 564 x 564pixels, so the population consisted of $(564)^2 = 318{,}096$,vectors from which the mean vector, covariance matrix,and corresponding eigenvalues and eigenvectors were computed.

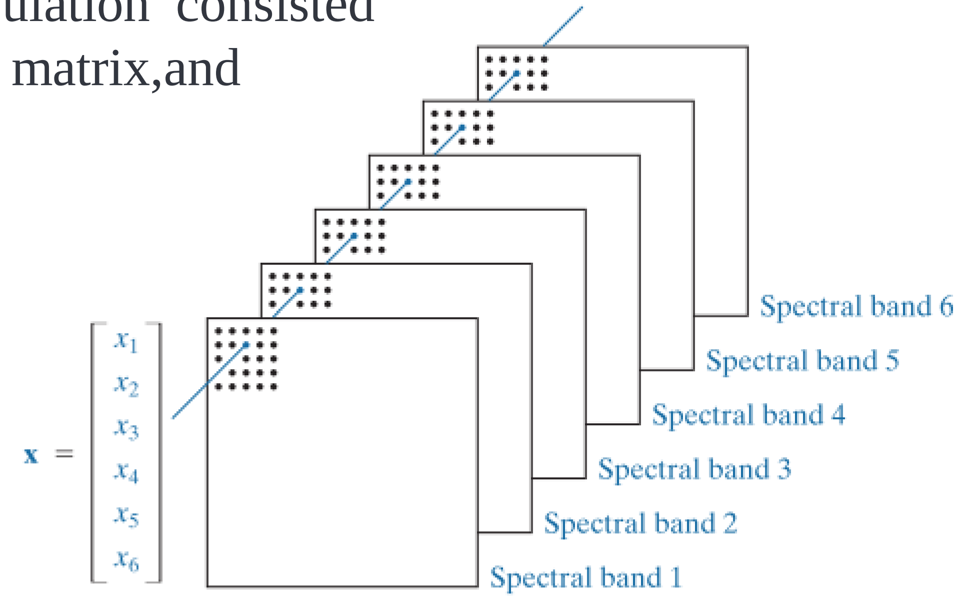
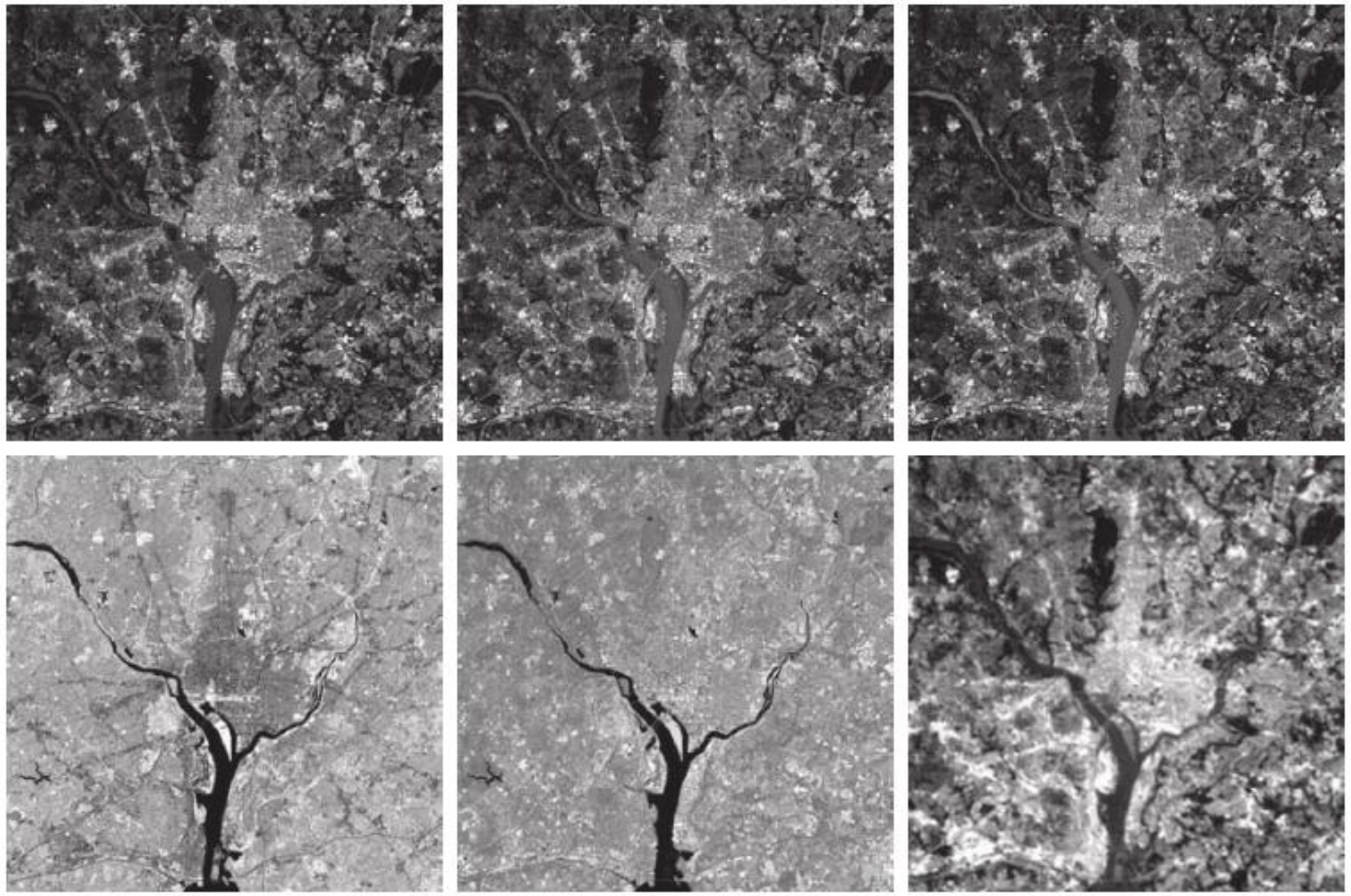


Figure 1.5
Forming of a feature vector from corresponding pixels in six images.

Using principal components for normalizing for variations in size, translation, and rotation:-

feature descriptors should be as independent as possible of variations in size, translation, and rotation. Principal components provide a convenient way to normalize boundaries and/or regions for variations in these three variables.

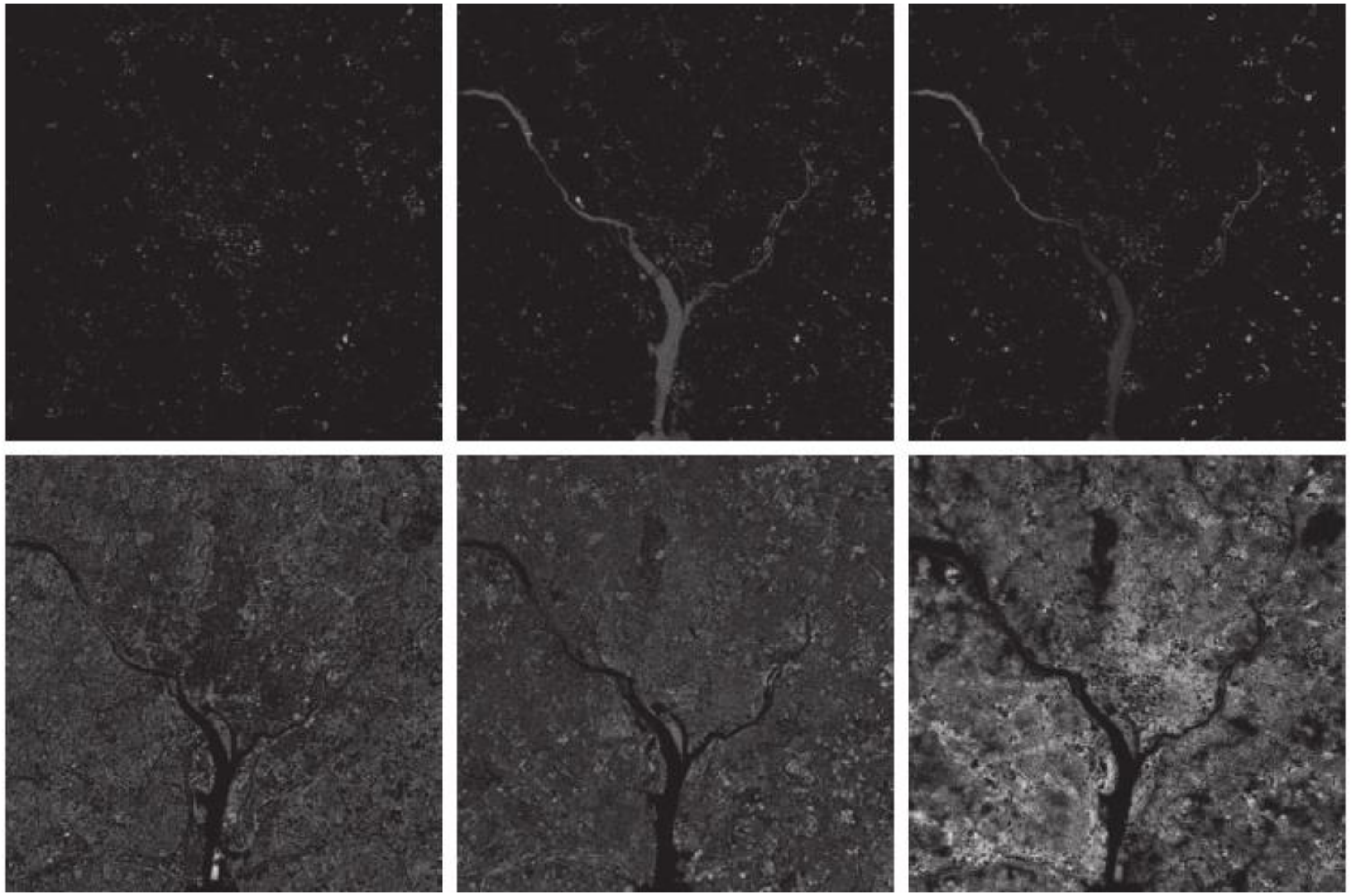


a b c
d e f

Multispectral images reconstructed using only the two principal component images corresponding to the two principal component vectors with the largest eigenvalues.

a b c
d e f

Differences between the original and reconstructed images. All images were enhanced by scaling them to the full [0, 255] range to facilitate visual analysis.



Whole-Image Features :-

The state of the art in image processing is such that as the complexity of the task increases, the number of techniques suitable for addressing those tasks decreases. This is particularly true when dealing with feature descriptors applicable to entire images that are members of a large family of images. In this section, we discuss two of the principal feature detection methods currently being used for this purpose. One is based on detecting corners, and the other works with entire regions in an image.

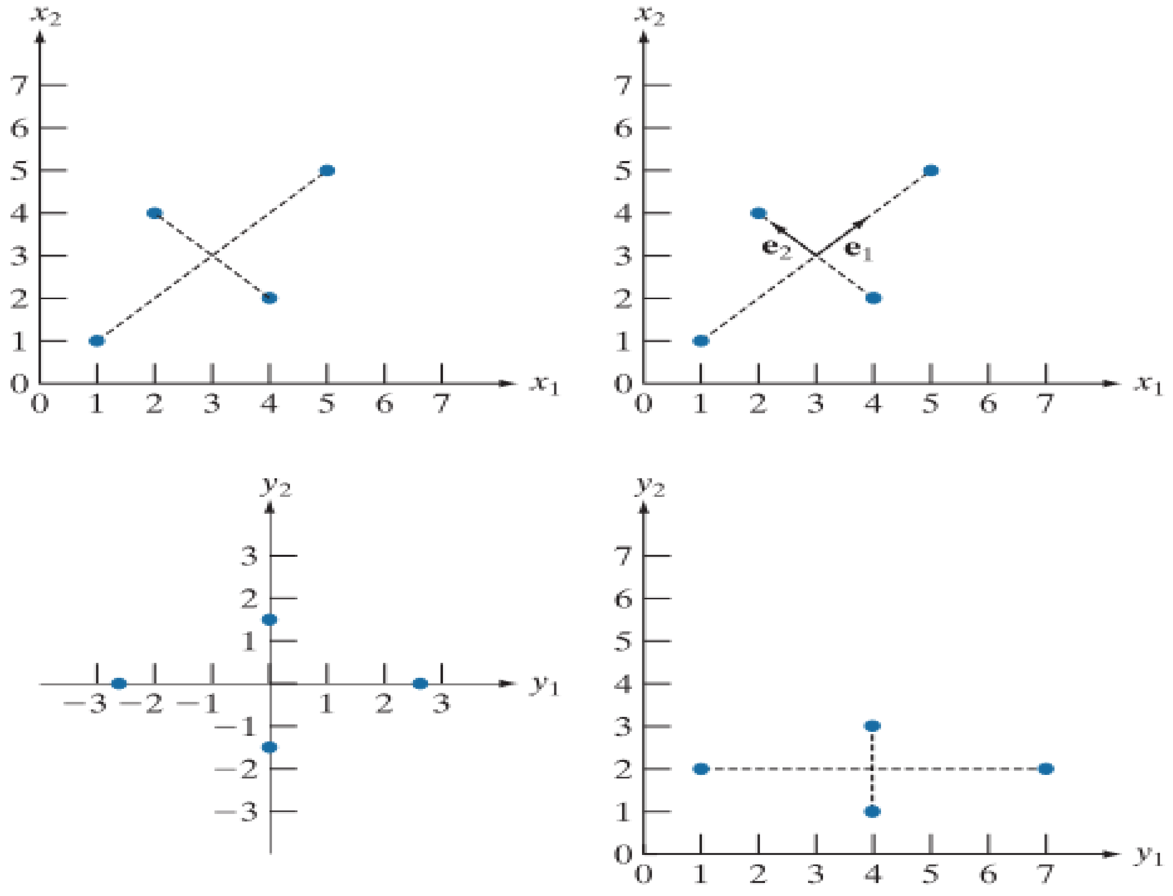
A manual example:

(a) Original points.

(b) Eigenvectors of the covariance matrix of the points in (a).

(c) Transformed points

(d) Points from (c), rounded and translated so that all coordinate values are integers greater than 0. The dashed lines are included to facilitate viewing. They are not part of the data.



THE HARRIS-STEPHENS CORNER DETECTOR:-

The Harris corner detector is a corner detection operator that is commonly used in computer vision algorithms to extract corners and infer features of an image.

Compared to the previous one, Harris' corner detector takes the differential of the corner score into account with reference to direction directly, instead of using shifting patches for every 45 degree angles, and has been proved to be more accurate in distinguishing between edges and corners.

Process of Harris corner detection algorithm:-

- Commonly, Harris corner detector algorithm can be divided into five steps.

1. Color to grayscale
2. Spatial derivative calculation
3. Structure tensor setup
4. Harris response calculation
5. Non-maximum suppression

Color to grayscale:

If we use Harris corner detector in a color image, the first step is to convert it into a grayscale image, which will enhance the processing speed.

The value of the gray scale pixel can be computed as a weighted sums of the values R, B and G of the color image,

$$\sum_{C \in \{R,G,B\}} w_C \cdot C,$$

where, e.g.,

$$w_R = 0.299,\ w_G = 0.587,\ w_B = 1 - (w_R + w_G) = 0.114.$$

Spatial derivative calculation:

Next, we are going to compute $I_x(x,y)$ and $I_y(x,y)$.

Structure tensor setup:

With $I_x(x,y)$, $I_y(x,y)$, we can construct the structure tensor M.

Harris response calculation:

For $x \ll y$, one has $\frac{x \cdot y}{x+y} = x \frac{1}{1+x/y} \approx x$. In this step, we compute the smallest eigenvalue of the structure tensor using that approximation:

$$\lambda_{\min} \approx \frac{\lambda_1 \lambda_2}{(\lambda_1 + \lambda_2)} = \frac{\det(M)}{\operatorname{tr}(M)}$$

with the trace $\operatorname{tr}(M) = m_{11} + m_{22}$.

Another commonly used Harris response calculation is shown as below,

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 = \det(M) - k \operatorname{tr}(M)^2$$

where k is an empirically determined constant; $k \in [0.04, 0.06]$.

Non-maximum suppression

In order to pick up the optimal values to indicate corners, we find the local maxima as corners within the window which is a 3 by 3 filter.

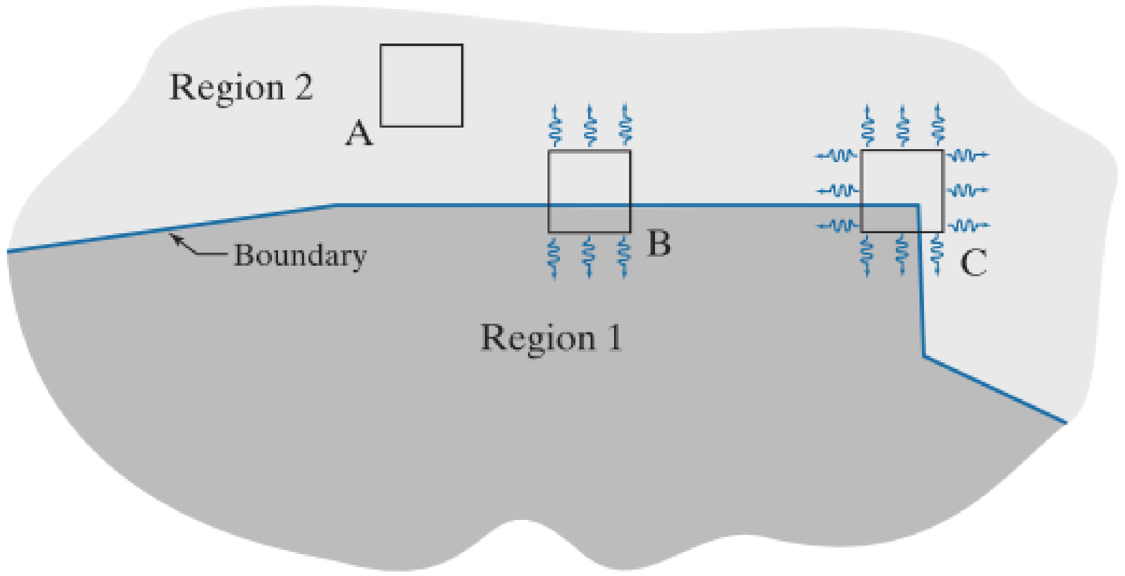
Improvement:-

1. Harris-Laplace Corner Detector
2. Differential Morphological Decomposition Based Corner Detector
3. Multi-scale Bilateral Structure Tensor Based Corner Detector

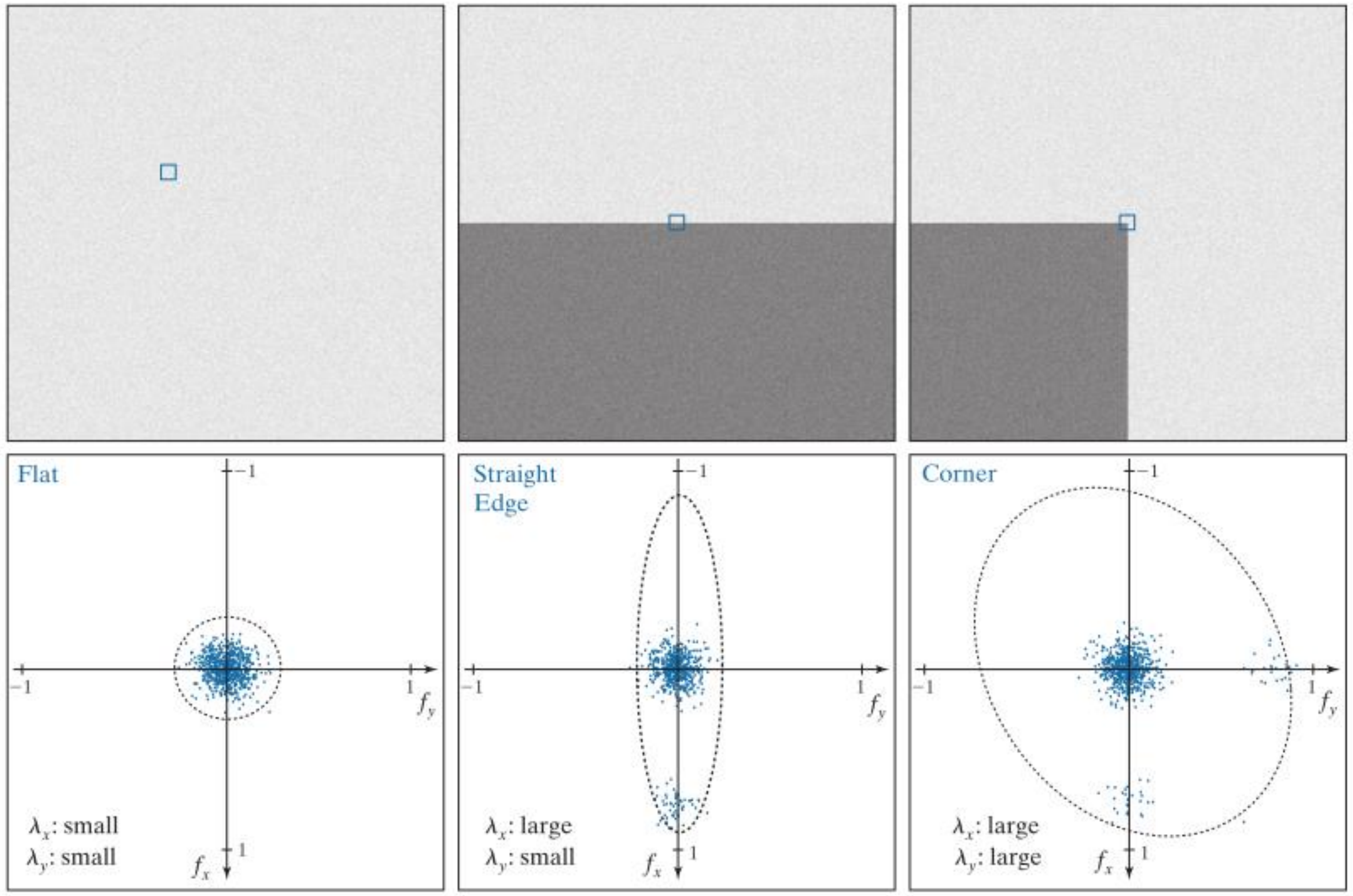
Applications:-

1. Image Alignment, Stitching and Registration
2. 2D Mosaics Creation
3. 3D Scene Modeling and Reconstruction
4. Motion Detection
5. Object Recognition
6. Image Indexing and Content-based Retrieval
7. Video Tracking.

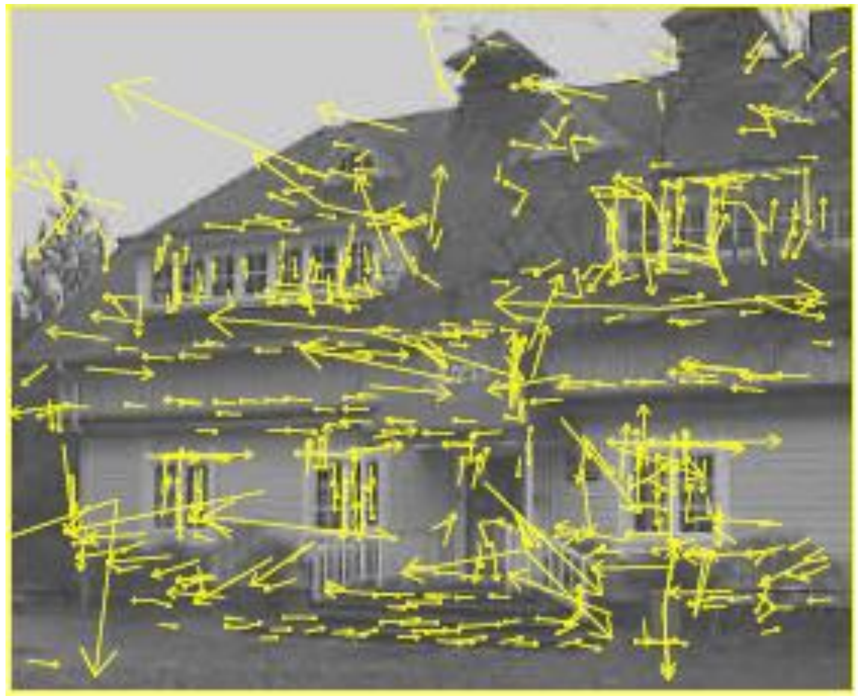
Figure Illustration of how the Harris-Stephens corner detector operates in the three types of subregions indicated by A (flat), B (edge), and C (corner). The wiggly arrows indicate graphically a directional response in the detector as it moves in the three areas shown.

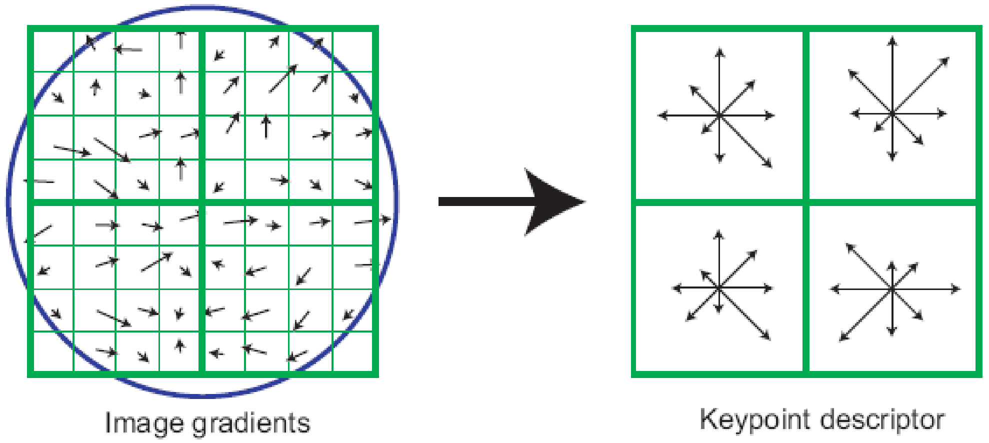


(a)–(c) Noisy images and image patches (small squares) encompassing image regions similar in content. (d)–(f) Plots of value pairs (fx,fy) useful for detecting the presence of a corner in an image patch.



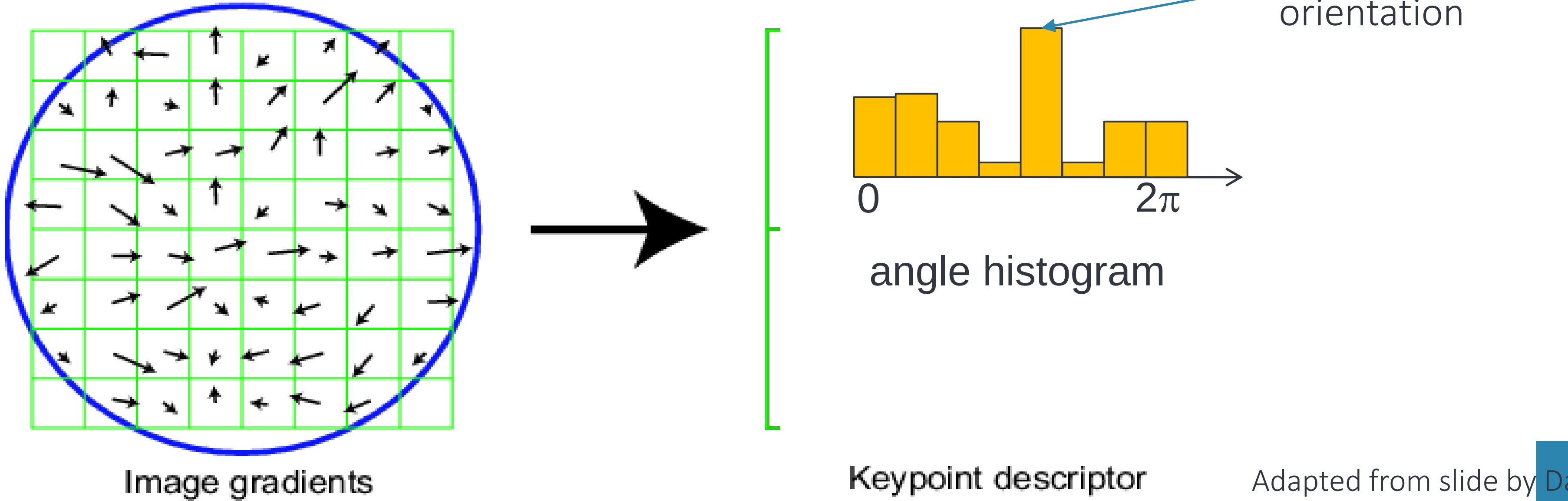
SCALE-INVARIANT FEATURE TRANSFORM (SIFT):-





Scale Invariant Feature Transform

- DoG for scale-space feature detection
- Take 16x16 square window around detected feature at appropriate scale
  - Compute gradient orientation for each pixel
  - Throw out weak edges (threshold gradient magnitude)
  - Create histogram of surviving edge orientations: note: each pixel contributes vote proportional to gradient magnitude
  - Find mode of histogram and rotate patch so that mode is 0

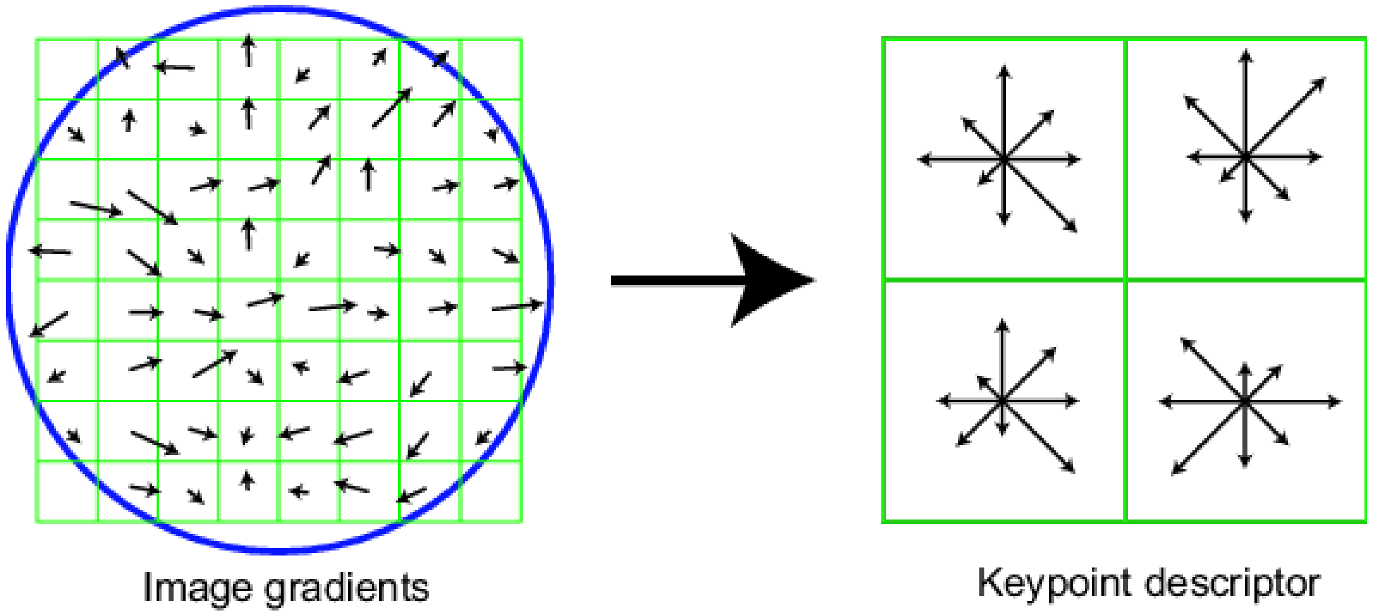


Adapted from slide by David Lowe

SIFT descriptor

Create histogram

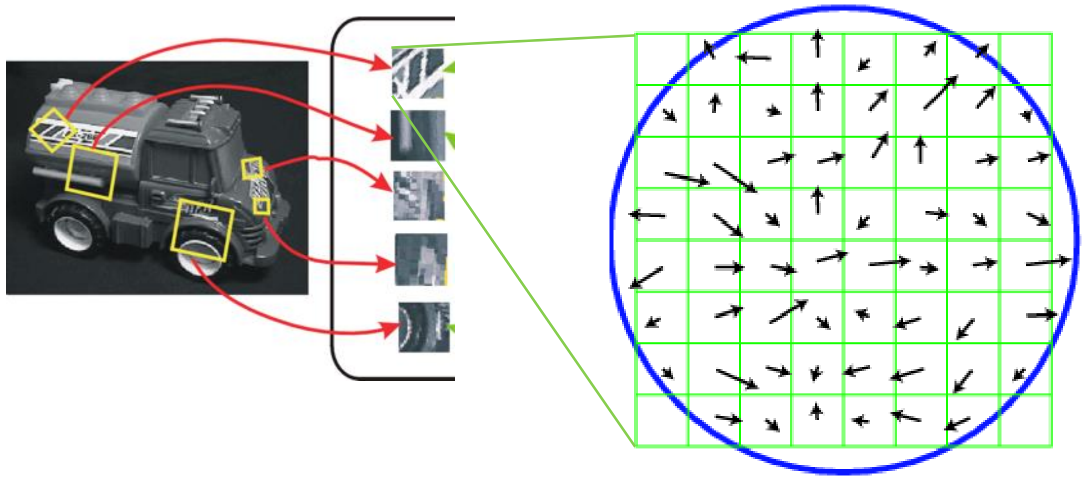
- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor



Adapted from slide by David Lowe

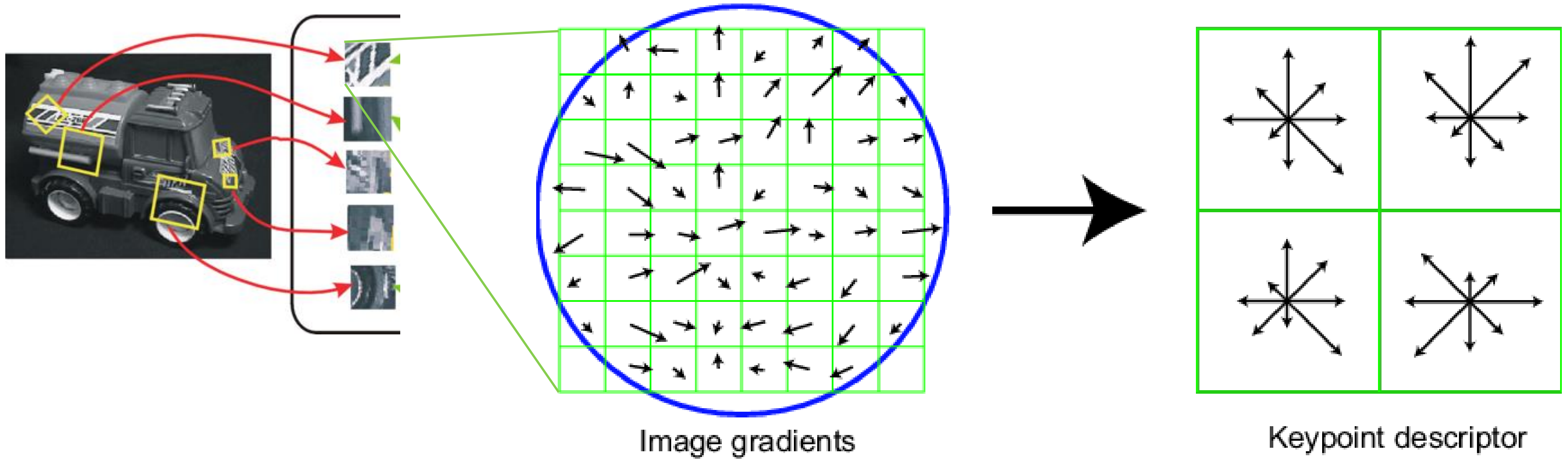
SIFT vector formation

- Computed on rotated and scaled version of window according to computed orientation & scale
  - resample the window



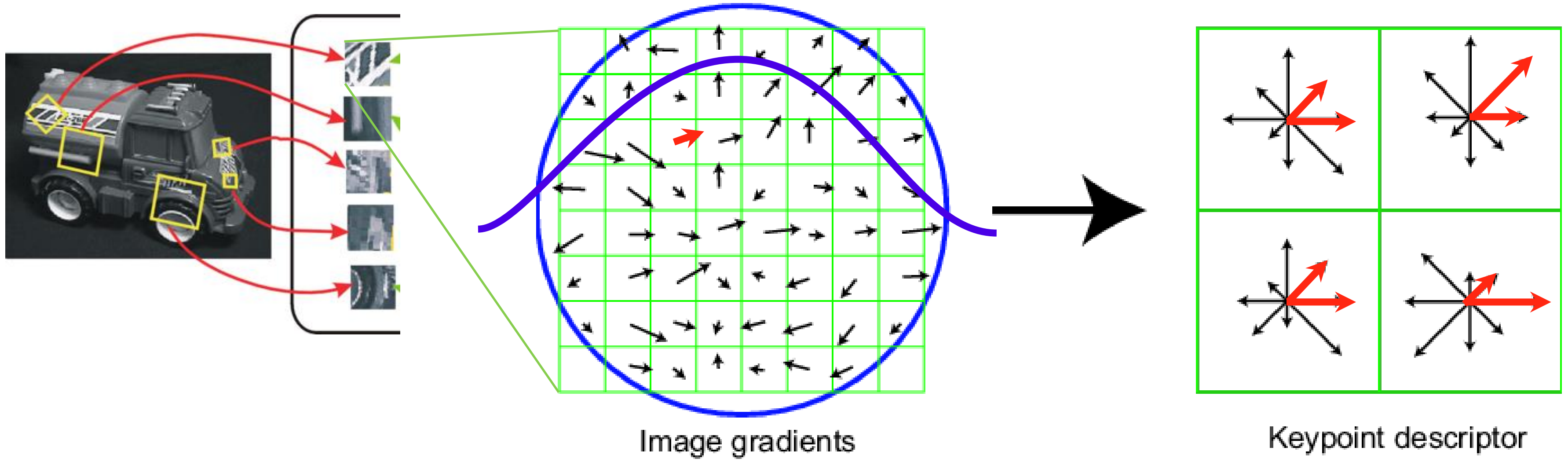
Reduce effect of illumination

- 128-dim vector normalized to 1: invariance to contrast changes
- Threshold gradient magnitudes to avoid excessive influence of high gradients
  - after normalization, clamp gradients >0.2
  - renormalize



Other tips and tricks

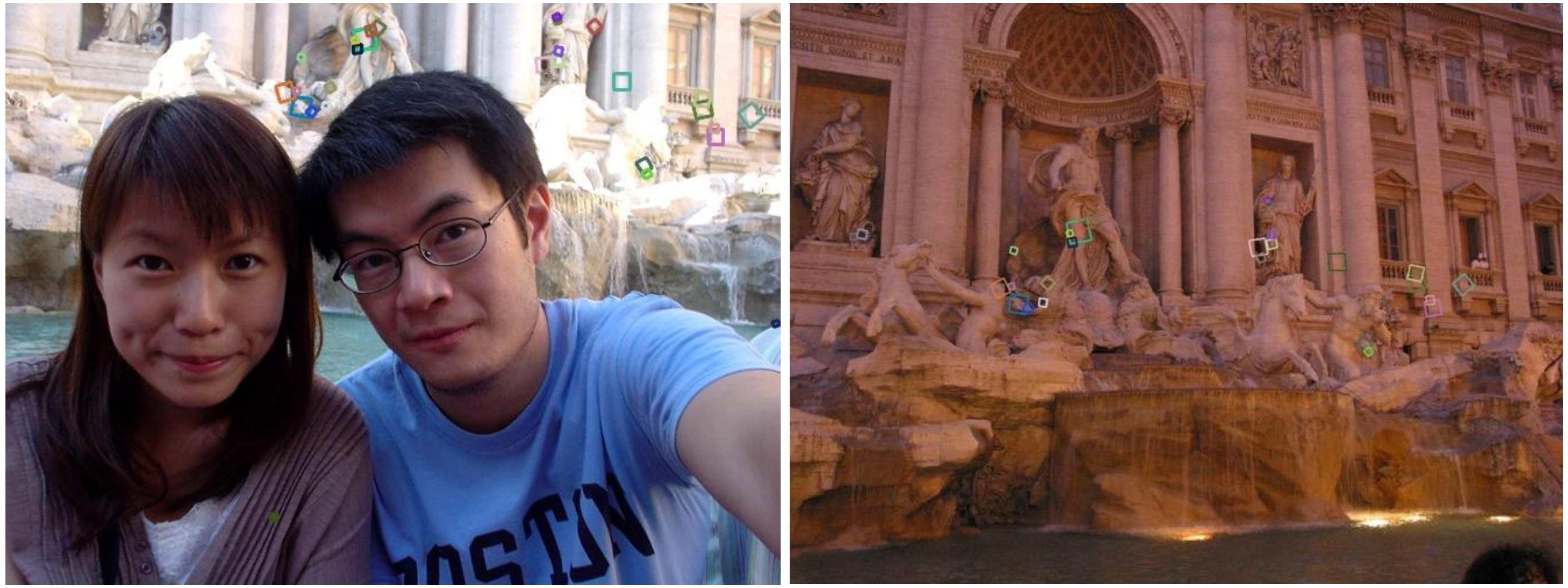
- When identifying dominant orientation, if multiple modes, create multiple keypoints
- Weigh pixels in center of patch more highly (Gaussian weights)
- Trilinear interpolation
    - a given gradient contributes to 8 bins:
      4 in space times 2 in orientation



Properties of SIFT

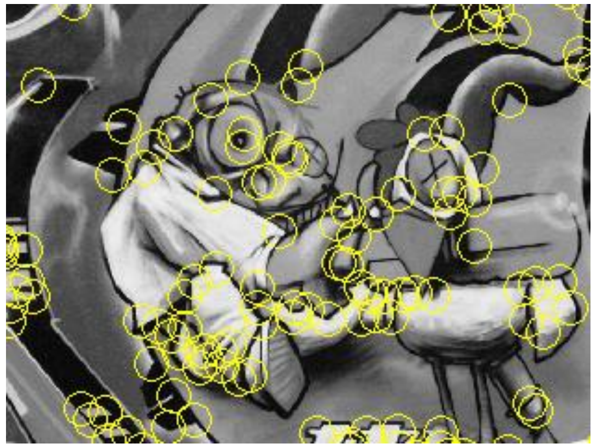
Extraordinarily robust matching technique

- Can handle changes in viewpoint
  - Up to about 60 degree out of plane rotation
- Can handle significant changes in illumination
  - Sometimes even day vs. night (below)
- Fast and efficient—can run in real time

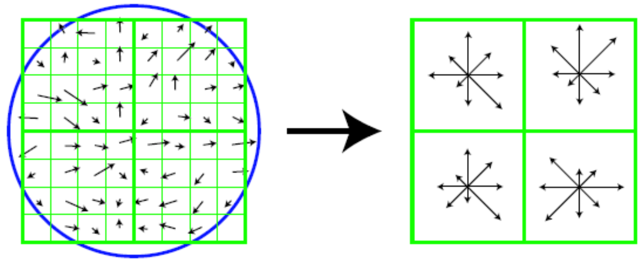


Summary

- Keypoint detection: repeatable and distinctive
  - Corners, blobs, stable regions
  - Harris, DoG



- Descriptors: invariant and discriminative
  - spatial histograms of orientation
- Next up: using correspondences for reconstruction



ANY
questions?