- The *second-order histogram* provides statistics based on pairs of pixels and their corresponding gray levels
- These methods are also referred to as *gray level co-occurrence matrix* or *gray level dependency matrix* methods
- These features are based on two parameters: *distance and angle*

1

- Distance is the pixel distance between the pairs of pixels, and the angle refers to the angle between the pixel pairs
- Four angles are used corresponding to vertical, horizontal, and two diagonals
- Typical pixel distance used is 1 or 2, but depends on the image resolution and the coarseness of the texture of interest
- To make the features rotationally invariant they can be calculated for all angles and then averaged

2

- Numerous features have been derived via these methods, but the following five have been found to be the most useful:

1. Energy (homogeneity/smoothness)
2. Inertia (contrast)
3. Correlation (similarity)
4. Inverse difference (local homogeneity)
5. Entropy (information content measure)

3

- Letting c_{ij} be the elements in the co-occurrence matrix normalized by dividing by the number of pixel pairs in the matrix, with a given distance and angle:

$$Energy = \sum_i \sum_j c_{ij}^2$$

$$Inertia = \sum_i \sum_j (i-j)^2 c_{ij}$$

$$Correlation = \frac{1}{\sigma_x \sigma_y} \sum_i \sum_j (i-\mu_x)(j-\mu_y) c_{ij}$$

$$where: \mu_x = \sum_i i \sum_j c_{ij}$$

$$and: \mu_y = \sum_j j \sum_i c_{ij}$$

$$and: \sigma_x^2 = \sum_i (1-\mu_x)^2 \sum_j c_{ij}$$

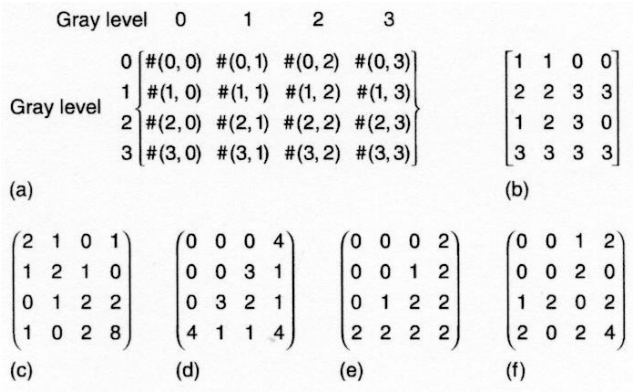
$$and: \sigma_y^2 = \sum_j (1-\mu_y)^2 \sum_i c_{ij},$$

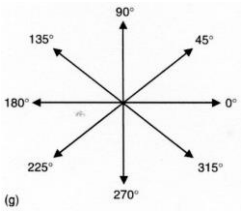
$$InverseDifference = \sum_i \sum_j \frac{c_{ij}}{|i-j|}; for: i \neq j$$

$$Entropy = -\sum_i \sum_j c_{ij} \log_2 c_{ij}$$

4

Figure 6.2-10: Example of Gray Level Co-occurrence Matrices





Given a 4x4 image with 4 possible gray levels, 2-bits per pixel, and using a distance d = 1, we have: a) General form of the matrix, where each entry is the number (#) of occurrences of the pair listed b) An example 4x4 image c) The matrix corresponding to the horizontal direction (0˚ and 180˚) d) The matrix corresponding to the vertical direction (90˚ and 270˚ degrees) e) The matrix corresponding to the left diagonal direction (135˚ and 315˚) f) The matrix corresponding to the right diagonal direction (45˚ and 225˚) g) Angle definitions

Remember it is ***important to normalize*** the values in the co-occurrence matrix by dividing by the number of pixel pairs in the matrix before calculating the texture features

5

- *Laws texture energy masks* are another method for measuring texture
- They work by finding the average gray *Level, Edges, Spots, Ripples* and *Waves* in the image
- They are based on the following 5 vectors:

$$L_5 = (1,4,6,4,1)$$
$$E_5 = (-1,-2,0,2,1)$$
$$S_5 = (-1,0,2,0,-1)$$
$$R_5 = (1,-4,6,-4,1)$$
$$W_5 = (-1,2,0,-2,-1)$$

6

- These are used to generate the Laws 5x5 filter masks by finding the vector outer product of each pair of vectors

  For example, using L_5 and S_5:

$$\begin{bmatrix} -1 & 0 & 2 & 0 & -1 \\ -4 & 0 & 8 & 0 & -4 \\ -6 & 0 & 12 & 0 & -6 \\ -4 & 0 & 8 & 0 & -4 \\ -1 & 0 & 2 & 0 & -1 \end{bmatrix}$$

7

- The first step in application of Laws energy masks is to preprocess the image to remove artifacts caused by uneven lighting
  - Useful as a preprocessing step for all texture measures

  1. Simple method is to find the local average for every pixel,15x15 window
  2. Then subtract this average from the current pixel in the center of the window

8

- The next step is to convolve the masks with the image to produce the texture filtered images, $F_k(r,c)$ for the k^{th} filter mask
- These texture filtered images are used to produce a *texture energy map*, E_k for the k^{th} filter:

$$E_k(r,c) = \sum_{j=c-7}^{c+7} \sum_{i=r-7}^{r+7} |F_k(i,j)|$$

- These energy maps are then used to generate a texture feature vector for each pixel, which can be used for texture classification

9

✓**Feature Extraction with CVIPtools**

- CVIPtools allows the user to extract features from objects within the image
- This is done by using the original image and a segmented or mask image to define the location of the object
- CVIPtools automatically labels the image, as required

10

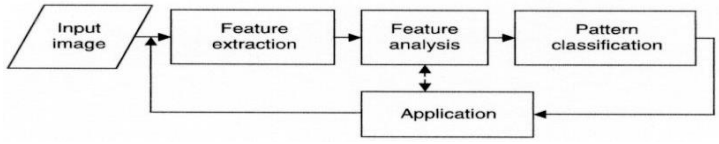
Table 6.2 Feature Extraction with CVIPtools

Feature category	How the features are extracted
Binary object	The labeled image is used by selecting the object corresponding to the row and column coordinates and treating the object as a binary image with the object = '1' and the background = '0'.
RST-invariant moment-based	The labeled image is used by selecting the object corresponding to the row and column coordinates and treating the object as a binary image with the object = '1' and the background = '0'.
Histogram	The labeled image is used by selecting the object corresponding to the row and column coordinates, and then this binary object is used as a mask on the original image to extract features. This is done by only including in the calculations pixels that are part of the object.
Texture	The labeled image is used by selecting the object corresponding to the row and column coordinates, and then this binary object is used as a mask on the original image to extract features. This is done by only including in the calculations pixels that are part of the object.
Spectral	The labeled image is used by selecting the object corresponding to the row and column coordinates, and then this binary object is used as a mask on the original image to extract features. This is done by creating a black image (all zeros) with dimensions a power of 2, imbedding the object from the original image within the black image, and then calculating the Fourier transform on this image.

11

➢Feature Analysis

- ✓After feature extraction, feature analysis is important to aid in the feature selection process
- ✓The features are carefully examined for utility and put back through the application feedback loop in the development process



12

- ✓ To understand the feature analysis process, we need to define these tools:
  1. Feature vectors and feature spaces
  2. Distance and similarity measures to compare feature vectors
  3. Methods to preprocess the feature data for pattern classification algorithms
  - The feature analysis process begins with selection of the tools and methods that will be used for specific imaging problem

13

✓**Feature Vectors and Feature Spaces**

- A feature vector is one method to represent an image, or part of an image (an object), by finding measurements on a set of features
- The *feature vector* is an *n*-dimensional vector that contains these measurements, where *n* is the number of features
- The measurements may be *symbolic, numerical,* or *both*

14

- The feature vector can be used to classify an object, or provide us with condensed higher-level image information
- A *feature space* is a mathematical abstraction, which is also *n*-dimensional and allows for visualization of feature vectors, and relationships between them

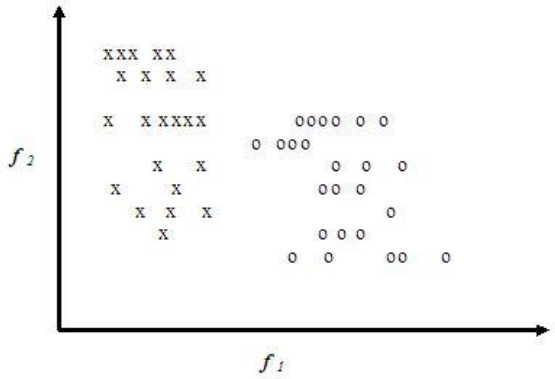
15

- With two and three-dimensional feature vectors the feature space is modeled as a geometric construct with perpendicular axes
- For *n*-dimensional feature vectors it is an abstract mathematical construction called a *hyperspace*
- The feature space allows us to define distance and similarity measures which are used to compare feature vectors and aid in the classification of unknown samples

16

Figure 6.3-1: A Two-Dimensional Feature Space

This shows a two-dimensional feature space defined by feature vectors, $\mathbf{F} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$, and two classes represented by *x* and *o*. Each *x* and *o* represents one sample in the feature space defined by its values for f_1 and f_2. One of the goals of feature analysis and pattern classification is to find clusters in the feature space which correspond to different classes



17

- **EXAMPLE 6.3.1:**

- We are working on a robotic control problem – to control a pick and place robotic gripper.
- To do this, we need to determine:
  1) Object location
  2) Object type – put type 1 in Box A, type 2 in Box B.
- First, we define the feature vector that will solve this problem. *Area* and *center of area* will locate the object and *perimeter* will identify the object.
- The feature vector contains four feature measures, and the feature space is four-dimensional.
- The feature vector is:

[area, r, c, perimeter]

18

✓Distance and Similarity Measures

- To perform the classification we need methods to compare two feature vectors
- The primary methods are to either measure the *difference* between the two, or to measure the *similarity*
- Two vectors that are closely related will have a small difference and a large similarity
- Measurements made in the *n*-dimensional feature space

19

❖***Euclidean distance***:

- The most common metric for measuring the distance between two vectors, A & B

$$\mathbf{A} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} and\ \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

- The Euclidean distance is:

$$\sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2 + \ldots + (a_n - b_n)^2}$$

20

❖***City block*** or ***absolute value metric***:

- It is defined as follows:

$$\sum_{i=1}^{n} |a_i - b_i| \qquad \mathbf{A} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \text{ and } \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

- This metric is computationally faster than the Euclidean distance, but gives similar results

21

❖***Maximum value*** metric:

- A distance metric that considers only largest difference
- It is defined as:

$$\max\{|a_1 - b_1|, |a_2 - b_2|, \ldots, |a_n - b_n|\} \qquad \mathbf{A} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \text{ and } \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

- Useful for applications where the largest difference of the feature set is most important

22

❖***Minkowski distance***:

- A *generalized distance metric*:

$$\left[\sum_{i=1}^{n}|a_i - b_i|^r\right]^{1/r}$$

where r is a positive integer

$$\mathbf{A} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \text{ and } \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

- The Minkowski distance is referred to as generalized because, for instance, if *r = 2,* it is the same as Euclidean distance and when *r = 1,* it is the city block metric

23

➢ *Similarity measures* are another type of metric used to compare feature vectors

➢ Unlike distance measures, two close feature vectors will have a large similarity measure

➢ Two commonly used similarity measures are the *vector inner product* and the *Tanimoto metric*

24

❖*Vector inner product*:

- The vector inner product is defined by the following equation (for vectors A & B):

$$\sum_{i=1}^{n} a_i b_i = (a_1 b_1 + a_2 b_2 + \ldots + a_n b_n) \qquad \mathbf{A} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \; and \; \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

25

❖*Tanimoto metric*:

- It is defined as:

$$\frac{\sum_{i=1}^{n} a_i b_i}{\sum_{i=1}^{n} a_i^2 + \sum_{i=1}^{n} b_i^2 - \sum_{i=1}^{n} a_i b_i} \qquad \mathbf{A} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \; and \; \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

- This metric takes on values between 0 and 1, which can be thought of as a “percent of similarity” since the value is 1 (100%) for identical vectors and gets smaller as the vectors get farther apart.

26

❖ ***Correclation Coefficient***:

$$S_{cc} = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{n}(a_i - \bar{a})^2 \sum_{i=1}^{n}(b_i - b)^2}}$$

- This metric assumes that the features have been normalized with *Standard Normal Density (SND)*
- *Take on value as following:*
  - *+1 for identical vectors*
  - *-1 for identical vectors but opposite in direction*
  - *0 for uncorrelated/orthogonal vectors*

27

✓ **Data Preprocessing**

- The data preprocessing consists of primarily three steps:

1. Noise removal
2. Data normalization and/or decorrelation
3. Insertion of missing data

➢ Classification algorithms may assume specific distributions such as zero mean, Gaussian distributed feature data

28

- The first step is noise removal, also called *outlier removal*, a data point that is alone and very far from the average value
- Classification algorithm development involves trying to find a model for the underlying structure
- Possibly a mistake was made during measurement, and the outlier does not really represent the underlying structure

29

- The second step, *data normalization and/or decorrelation*, is done to avoid biasing distance and similarity measures due to the varying ranges on different vector components
- The third step, *data insertion*, may be used to complete missing data in the data used for algorithm development
- Next, we consider various methods for data normalization (step 2)

30

❖***Range-normalize***:

- Performed by dividing each vector component by the data range for that component
- The range is the maximum value for that component minus the minimum
- Advantage of this method is simplicity

31

❖***Unit vector normalization***:

- Modifies the feature vectors so that they all have a magnitude of 1
- Divide the vector components by the vector length (magnitude)
- Retains only directional information about the vector, which preserves relationships between the features, but loses magnitudes

32

EXAMPLE 6.3.2:

Given the two vectors

$$\mathbf{A} = \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix} \textit{ and } \mathbf{B} = \begin{bmatrix} 6 \\ 10 \\ 2 \end{bmatrix}$$

The Euclidean distance of vector **A** from the origin, also called the magnitude is:

$$\sqrt{(3-0)^2 + (5-0)^2 + (1-0)^2} = \sqrt{9+25+1} = \sqrt{35}$$

The distance of vector **B** from the origin, also called the magnitude is:

$$\sqrt{6^2 + 10^2 + 2^2} = \sqrt{36+100+4} = \sqrt{140}$$

33

EXAMPLE 6.3.2 (contd):

To normalize these two vectors to unit vectors, we divide each component by the vector magnitude:

$$\mathbf{A}' = \begin{bmatrix} \frac{3}{\sqrt{35}} \\ \frac{5}{\sqrt{35}} \\ \frac{1}{\sqrt{35}} \end{bmatrix} \approx \begin{bmatrix} 0.507 \\ 0.845 \\ 0.169 \end{bmatrix} \textit{ and } \mathbf{B}' = \begin{bmatrix} \frac{6}{\sqrt{140}} \\ \frac{10}{\sqrt{140}} \\ \frac{2}{\sqrt{140}} \end{bmatrix} \approx \begin{bmatrix} 0.507 \\ 0.845 \\ 0.169 \end{bmatrix}$$

So we can see that **A′** = **B′**. In this case, the second vector is in the same direction as the first, but is twice as long, that is, each component is multiplied by 2

34

❖*Standard normal density normalization*:

- A commonly used statistical-based method to normalize the features by taking each vector component and subtracting the mean and dividing by the standard deviation
- Requires knowledge of the probability distribution of the feature measurements
- In practice the probability distributions are often estimated by using the existing data

35

This is done as follows, given a set of k feature vectors, $\mathbf{F}_j = \{\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_k\}$, with n features in each vector:

$$\mathbf{F}_j = \begin{bmatrix} f_{1j} \\ f_{2j} \\ \vdots \\ f_{nj} \end{bmatrix} \text{ for } j = 1,2,\ldots,k$$

$$\text{means} \Rightarrow m_i = \frac{1}{k}\sum_{j=1}^{k} f_{ij} \quad \text{for } i = 1,2,\ldots,n$$

$$\text{standard deviation} \Rightarrow \sigma_i = \sqrt{\frac{1}{k}\sum_{j=1}^{k}(f_{ij} - m_i)^2} = \sqrt{\frac{1}{k}\sum_{j=1}^{k}(f_{ij})^2 - m_i^2} \quad \text{for } i = 1,2,\ldots,n$$

Now, for each feature component, we subtract the mean and divide by the standard deviation:

$$f_{ijSND} = \frac{f_{ij} - m_i}{\sigma_i} \text{ for all } i, j$$

➢ Provides new feature vectors with normalized distribution with means of 0 and standard deviations of 1

➢ The resulting distribution on the each vector component is called the *standard normal density (SND)*

36

❖*Min-max normalization*:

- It is used to map the data to a specified range, S_{MIN} to S_{MAX}, but still retain the relationship between the values
- It is defined as:

$$f_{ijMINMAX} = \left(\frac{f_{ij} - f_{MIN}}{f_{MAX} - f_{MIN}}\right)(S_{MAX} - S_{MIN}) + S_{MIN}$$

where

S_{MIN} and S_{MAX} are minimum and maximum value for the specified range

and

f_{MIN} and f_{MAX} are minimum and maximum value on the original feature data

37

❖*Softmax scaling*:

- A nonlinear normalization method for use with skewed data, requires two steps:

$$STEP1 \Rightarrow y = \frac{f_{ij} - m_i}{r\,\sigma_i}$$

$$STEP2 \Rightarrow f_{ijSMC} = \frac{1}{1+e^{-y}} \text{ for all } i, j$$

  - ➢Compresses the data into the range of 0 to 1
  - ➢The first step is similar to mapping the data to the SND, but with a user defined factor, *r*

38

- *Softmax scaling* is approximately linear for small values of y with respect to f_{ij}, and then compresses the data exponentially as it gets farther away from the mean
- The factor r determines the range of values for the feature, f_{ij}, that will fall into the linear range
- In addition to moving the mean and normalizing the spread of the data, this transform changes the shape of the distribution

39

- Changing the spread and/or shape of the data distribution must be done with caution
- If useful information is contained in the mean, spread or shape of the data distribution, loss of the information may not be desired
- Choice of the wrong normalization method will effectively filter out useful information

40

- Performing a principal components transform (PCT) decorrelates feature data
- It is a desirable preprocessing step for some classification algorithm development methods, such as neural networks
- The PCT provides new features that are linear transforms of the original features, and are uncorrelated
- Use of the PCT, also referred to as *principal components analysis (PCA)*, is also useful for data visualization in feature space

41

- The final step in data preprocessing is to *insert missing data*
- Analyze the distribution of the sample feature vectors and, based on a desired data distribution, create feature vectors to insert
- To avoid biasing results requires a complete understanding of the problem, including how the features relate to the desired output and the underlying structure of each feature's distribution
- Typically this information is unknown, so a trial and error approach is used during development

42