6

Image analysis

6.1 *Introduction: inspection, location, and identification*

Image analysis is the term that is used to embody the idea of automatically extracting useful information from an image of a scene. The important point regarding image analysis is that this information is explicit and can be used in subsequent decision making processes. Techniques vary across a broad spectrum, depending on the complexity of the image and, indeed, on the complexity of the information to be extracted from it. The more commonly used image analysis techniques include template matching, statistical pattern recognition, and the Hough transform. Unfortunately, this classification is not particularly useful when one is trying to identify a technique for a potential application. However, we can also classify the types of analysis we wish to perform according to function. There are essentially three types of things we would wish to know about the scene in an image. First, we might wish to ascertain whether or not the visual appearance of objects is as it should be, i.e. we might wish to inspect the objects. The implicit assumption here is, of course, that we know what objects are in the image in the first place and approximately where they are. If we don't know where they are, we might wish to find out. This is the second function of image analysis: location. Note that the location of an object requires the specification of both position and orientation (in either two dimensions or three dimensions). Also, the coordinates might be specified in terms of the image frame of reference (where distance is specified in terms of pixels) or in the real world where distances correspond to millimetres, say. The latter obviously necessitates some form of calibration, since initial measurements will be made in the image frame of reference. Finally, if we do not know what the objects in the image are, we might have to perform a third type of analysis: identification.

Generally speaking, inspection applications utilize the template matching paradigm, location problems utilize the template matching paradigm and the

Hough transform, while the problem of identification can be addressed using all three techniques, depending on the complexity of the image and the objects. From time to time, all of them make use of some of the more advanced techniques to be described in later chapters. For example, if we are interested in the local shape or three-dimensional structure, we might exploit the structured-light techniques described in Chapter 8.

6.2 *Template matching*

Many of the applications of computer vision simply need to know whether an image contains some previously defined object or, in particular, whether a pre-defined sub-image is contained within a test image. The sub-image is called a *template* and should be an ideal representation of the pattern or object which is being sought in the image. The template matching technique involves the translation of the template to every possible position in the image and the evaluation of a measure of the match between the template and the image at that position. If the similarity measure is large enough then the object can be assumed to be present. If the template does represent the complete object for which you wish to check the image, then the technique is sometimes referred to as 'global template matching', since the template is in effect a global representation of the object. On the other hand, local template matching utilizes several templates of local features of the object, e.g. corners in the boundary or characteristic marks, to represent the object.

6.2.1 *Measures of similarity*

Apart from this distinction between global and local template matching, the only other aspect which requires detailed consideration is the measure of similarity between template and image. Several similarity measures are possible, some based on the summation of differences between the image and template, others based on cross-correlation techniques. Since similarity measures are widely used, not just in this image template matching situation, but also for evaluation of the similarity between any two signatures (i.e. characteristic signals), such as when comparing shape descriptors, it is worth discussing these similarity measures in more detail. We will look at measures based on Euclidean distance and cross-correlation.

A common measure employed when comparing the similarity of two images (e.g. the template $t(i, j)$ and the test image $g(i, j)$) is the metric based on the standard Euclidean distance between two sectors, defined by:

$$E(m, n) = \sqrt{\left\{\sum_i \sum_j [g(i, j) - t(i - m, j - n)]^2\right\}}$$

The summation is evaluated for all i, such that $(i - m)$ is a valid coordinate of the template sub-image. This definition amounts to translating the template $t(i, j)$ to a position (m, n) along the test image and evaluating the similarity measure at that point. Thus, when searching for a template shape, the template is effectively moved along the test image and the above template match is evaluated at each position. The position (m, n) at which the smallest value of $E(m, n)$ is obtained corresponds to the best match for the template.

The similarity measure based on the Euclidean distance is quite an appealing method, from an intuitive point of view. To see why, consider a complete one-dimensional entity, e.g. size (represented by, say, length). To compare the difference in size of two objects, we just subtract the values, square the difference and take the square root of the result, leaving us with the absolute difference in size:

$$d = \sqrt{[(s_1 - s_2)^2]}$$

Extending this to the two-dimensional case, we might wish to see how far apart two objects are on a table, i.e. to compute the distance between them. The difference in position is simply:

$$d = \sqrt{[(x_1 - x_2)^2 + (y_1 - y_2)^2]}$$

Similarly, in three dimensions

$$d = \sqrt{[(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2]}$$

We can easily extend this to n dimensions (although we lose the intuitive concept of 'distance') by just making each coordinate an independent variable which characterizes the entities we are comparing. For example, a 10×10 image template comprises 100 independent pixels, each of which specifies the template sub-image. Thus, we are now dealing with a $10 \times 10 = 100$-dimensional comparison and the difference between the two sub-images is:

$$d = \sqrt{\{[\text{image}(1,1) - \text{template}(1,1)]^2 + \cdots + [\text{image}(10,10) - \text{template}(10,10)]^2\}}$$

which is identical to our definition of the Euclidean metric.

A frequently used and simpler template matching metric is based on the absolute difference of $g(i, j)$ and $t(i - m, j - n)$ rather than the square of the difference. It is defined by:

$$S(m, n) = \sum_i \sum_j | g(i, j) - t(i - m, j - n) |$$

Alternatively, the square root in the Euclidean definition can be removed by squaring both sides of the equation and letting the similarity measure be $E^2(m, n)$. Hence:

$$E^2(m, n) = \sum_i \sum_j [g(i, j)^2 - 2g(i, j)\, t(i - m, j - n) + t(i - m, j - n)^2]$$

As before, the summation is evaluated for all i and j, such that $(i - m, j - n)$ is a valid coordinate of the template sub-image. Note that the summation of the last term is constant since it is a function of the template only and is evaluated over the complete domain of the template. If it is assumed that the first term is also

constant, or that the variation is small enough to be ignored, then $E^2(m, n)$ is small when the summation of the middle term is large. Thus, a new similarity measure might be $R(m, n)$, given by:

$$R(m, n) = \sum_i \sum_j g(i, j)\, t(i - m, j - n)$$

again summing over the usual range of i and j; $R(m, n)$ is the familiar cross-correlation function. The template $t(i - m, j - n)$ and the section of $g(i, j)$ in the vicinity of (m, n) are similar when the cross-correlation is large.

If the assumption that the summation of $g(i, j)$ is independent of m and n is not valid, an alternative to computing R is to compute the normalized cross-correlation $N(m, n)$ given by:

$$N(m, n) = R(m, n) \Big/ \sqrt{\left[\sum_i \sum_j g(i, j)^2\right]}$$

summing over the usual range of i and j. Note that, by the Cauchy–Schwarz inequality:

$$N(m, n) \leqslant \sqrt{\left[\sum_i \sum_j t(i - m, j - n)^2\right]}$$

Hence, the normalized cross-correlation may be scaled so that it lies in the range 0 to 1 by dividing it by the above expression. Thus, the normalized cross-correlation may be redefined:

$$N(m, n) = R(m, n) \Big/ \left\{\sqrt{\left[\sum_i \sum_j g(i, j)^2\right]} \sqrt{\left[\sum_i \sum_j t(i - m, j - n)^2\right]}\right\}$$

Figure 6.1 illustrates the use of cross-correlation in which a template of a human eye was recorded and subsequently located in a series of images. The cross-hair denotes the position at which the maximum cross-correlation between template and image occurred.

6.2.2 Local template matching

One of the problems of template matching is that each template represents the object or part of it as we expect to find it in the image. No cognizance is taken of variations in scale or in orientation. If the expected orientation can vary, then we will require a separate template for each orientation and each one must be matched with the image. Thus template matching can become computationally expensive, especially if the templates are large. One popular way of alleviating this computational overhead is to use much smaller local templates to detect salient features in the image which characterize the object we are looking for. The spatial relationships between occurrences of these features are then analysed. We can infer the presence of the object if valid distances between these features occur.

In summary, template matching techniques are useful in applications that can

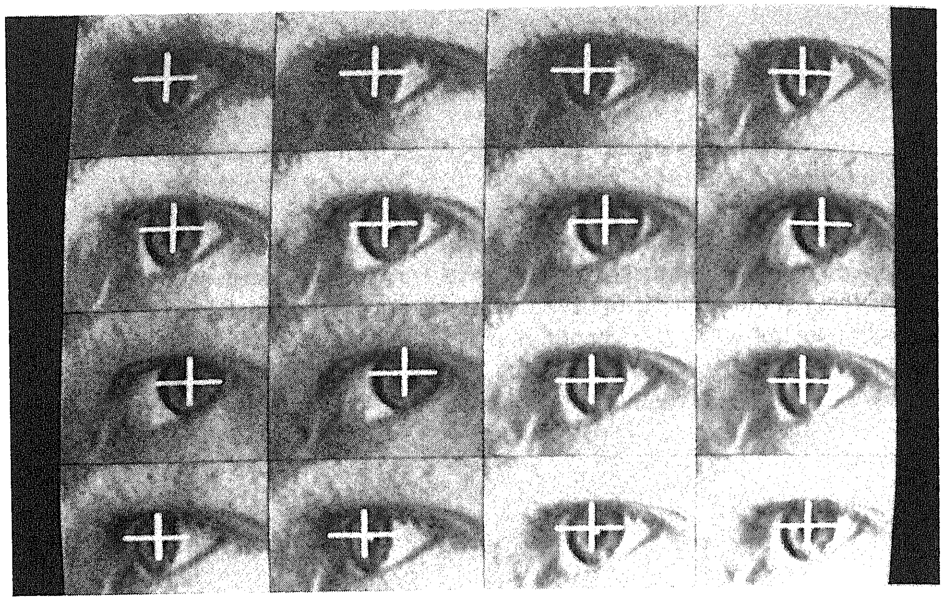

Figure 6.1 Eye-tracking using normalized cross-correlation.

be severely restricted and where the number of objects, and the variability of these objects, is small; it is not an approach that is applicable in general situations.

6.3 Decision-theoretic approaches

If the objective of the image analysis is to find objects within the image and identify, or classify, those objects then an approach based on statistical decision theory may be the most appropriate route to take. The central assumption in this approach is that the image depicts one or more objects and that each object belongs to one of several distinct and exclusive pre-determined classes, i.e. we know what objects exist and an object can only have one particular type or label.

6.3.1 Components of a statistical pattern recognition process

There are effectively three components of this type of pattern recognition process: an object isolation module, a feature extraction module, and a classification module. Each of these modules is invoked in turn and in the order given, the output of one module forming the input of the next. Thus, the object isolation module operates on a digital image and produces a representation of the object. The feature extraction module then abstracts one or more characteristic features and produces

a (so-called) feature vector. This feature vector is then used by the classification module to identify and label each object.

Since we will be covering some of these topics again in more detail later on, e.g. methods for object isolation and description, we will just give a brief overview of the representative techniques at this stage.

Object isolation, often referred to as 'segmentation', is in effect the grouping process which we discussed in the preceding chapter. The similarity measure upon which the grouping process is based in this instance is the grey-level of the region.

Once we have segmented the image, we have essentially identified the objects which we wish to classify or identify. The next phase of the recognition scheme is the extraction of features which are characteristic of the object and which will be used in the classification module. The selection of the features to be used is an extremely important task, since all subsequent decisions will be based on them, and frequently it is intuition and experience which guides the selection process. Normally, we will identify a number of reasonable feasible potential features, test these to check their performance, and then select the final set of features to be used in the actual application. When selecting features, you should bear in mind the desirability of each feature being *independent* (a change in one feature should not change the value of another feature significantly), *discriminatory* (each feature should have a significantly different value for each different object), *reliable* (features should have the same value for all objects in the same class/group). Finally, it is worth noting that the computational complexity of the pattern recognition exercise increases very rapidly as the number of features increases and hence it is desirable to use the fewest number of features possible, while ensuring a minimal number of errors.

6.3.2 Simple feature extraction

Before proceeding to discuss the mechanism by which we can classify the objects, let us first take a look at some representative features that we might use to describe that object.

Most features are either based on the size of the object or on its shape. The most obvious feature which is based on size is the area of the object: this is simply the number of pixels comprising the object multiplied by the area of a single pixel (frequently assumed to be a single unit). If we are dealing with grey-scale images, then the integrated optical density (IOD) is sometimes used: it is equivalent to the area multiplied by the average grey-level of the object and essentially provides a measure of the 'weight' of the object, where the pixel grey-level encodes the weight per unit area.

The length and the width of an object also describe its size. However, since we will not know its orientation in general, we may have to first compute its orientation before evaluating the minimum and maximum extent of its boundary, providing us with a measure of its length and width. Thus, these measures should always be made with respect to some rotation-invariant datum line in the object, e.g. its major or minor axis. The *minimum bounding rectangle* is a feature which is related to this idea of length and width. This is the smallest rectangle which can completely enclose the object. The main axis of this rectangle is in fact the principal axis of the object itself and, hence, the dimensions of the minimum bounding rectangle correspond to the features of length and width.

Quite often, the distance around the perimeter of the object can be useful for discriminating between two objects (quite apart from the fact that one can compute the area of the object from the perimeter shape). Depending on how the object is represented, and this in turn depends on the type of segmentation used, it can be quite trivial to compute the length of the perimeter and this makes it an attractive feature for industrial vision applications.

Features which encode the shape of an object are usually very useful for the purposes of classification and because of this, Chapter 7 has been entirely given over to them. For the present, we will content ourselves by mentioning two very simple shape measures: rectangularity and circularity. There are two popular measures of rectangularity, both of which are easy to compute. The first is the ratio of the area of the object to the area of the minimum bounding rectangle:

$$R = \frac{A_{\text{object}}}{A_{\text{min. bound. rectangle}}}$$

This feature takes on a maximum value of 1 for a perfect rectangular shape and tends toward zero for thin curvy objects.

The second measure is the aspect ratio and is simply the ratio of the width of the minimum bounding rectangle to its length:

$$Aspect\ ratio = \frac{W_{\text{min. bound. rectangle}}}{L_{\text{min. bound. rectangle}}}$$

The most commonly used circularity measure is the ratio of the square of the perimeter length to the area:

$$C = \frac{A_{\text{object}}}{P^2_{\text{object}}}$$

This assumes a maximum value for discs and tends towards zero for irregular shapes with ragged boundaries.

6.3.3 Classification

The final stage of the statistical pattern recognition exercise is the classification of the objects on the basis of the set of features we have just computed, i.e. on the basis of the feature vector. If one views the feature values as 'coordinates' of a point in *n*-dimensional space (one feature value implies a one-dimensional space, two features imply a two-dimensional space, and so on), then one may view the object of classification as being the determination of the sub-space of the feature

space to which the feature vector belongs. Since each sub-space corresponds to a distinct object, the classification essentially accomplishes the object identification.

For example, consider a pattern recognition application which requires us to discriminate between nuts, bolts, and washers on a conveyor belt. Assuming that we can segment these objects adequately, we might choose to use two features on which to base the classification: washers and nuts are almost circular in shape, while bolts are quite long in comparison, so we decide to use a circularity measure as one feature. Furthermore, washers have a larger diameter than nuts, and bolts have an even larger maximum dimension. Thus, we decide to use the maximum length of the object (its diameter in the case of the nuts and washers) as the second feature. If we then proceed to measure these feature values for a fairly large set of these objects, called the training set, and plot the results on a piece of graph paper (representing the two-dimensional feature space, since there are two features) we will probably observe the clustering pattern shown in Figure 6.2 where nuts, bolts, and washers are all grouped in distinct sub-spaces.

At this stage, we are now ready to classify an unknown object (assuming, as always, that it is either a nut, a bolt or a washer). We generate the feature vector for this unknown object (i.e. compute the maximum dimension and its circularity measure A/P^2) and see where this takes us in the feature space (see Figure 6.3). The question is now: to which sub-space does the vector belong, i.e. to which class does the object belong? One of the most popular and simple techniques, the nearest-neighbour classification technique, classifies the object on the basis of the distance of the unknown object vector position from the centre of the three clusters, choosing the closest cluster as the one to which it belongs. In this instance, the

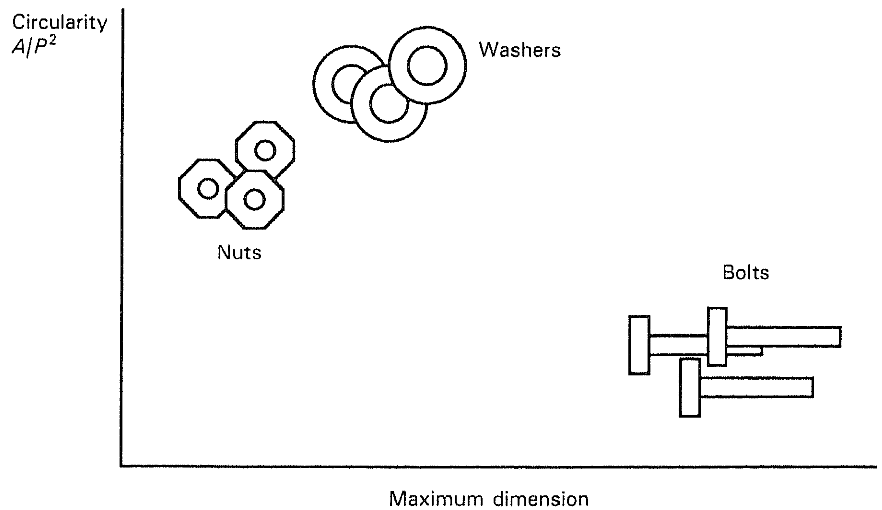

Figure 6.2 Feature space.

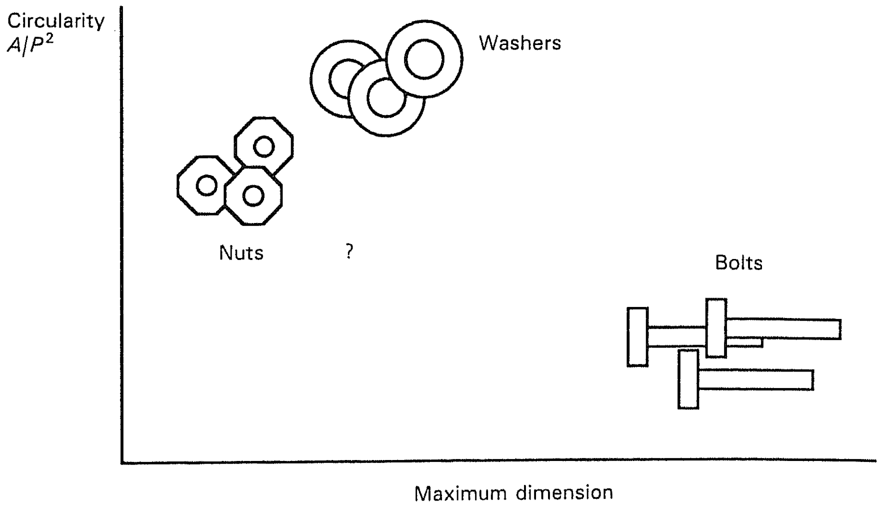

Figure 6.3 Coordinates of an unknown object in the feature space.

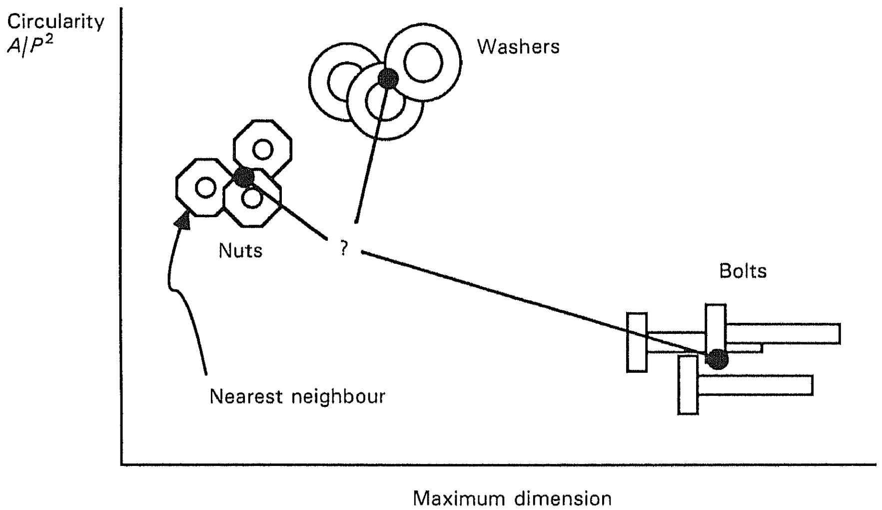

Figure 6.4 Nearest neighbour classification.

object is a nut (see Figure 6.4). This technique is called, not surprisingly, the nearest-neighbour classifier. Incidentally, the position of the centre of each cluster is simply the average of each of the individual training vector positions.

6.3.3.1 A synopsis of classification using Bayes' rule

There are, however, more sophisticated approaches to classification. The one which

we are going to describe here utilizes Bayes' theorem from statistical decision theory and is called the *maximum-likelihood classifier*. We will develop the discussion using an example which requires only one feature to discriminate between two objects; we do this because it is easier to visualize (and draw!) the concepts being discussed.

Suppose that in a situation similar to that described in the preceding example, we wish to distinguish between nuts and bolts (no washers this time). In this instance, the circularity measure will suffice and we now have just one feature and one-dimensional feature space with two classes of object: nuts and bolts. Let us refer to these classes as C_n and C_b. Let us also refer, for brevity, to the circularity feature value as x. The first thing that is required is the probability density functions (PDFs) for each of these two classes, i.e. a measure of the probabilities that an object from a particular class will have a given feature value. Since it is not likely that we will know these *a priori*, we will probably have to estimate them. The PDF for nuts can be estimated in a relatively simple manner by measuring the value of x for a large number of nuts, plotting the histogram of these values, smoothing the histogram, and normalizing the values so that the total area under the histogram equals one. The normalization step is necessary since probability values have values between zero and one and the sum of all the probabilities (for all the possible circularity measures) must necessarily be equal to a certainty of encountering that object, i.e. a probability value of one. The PDF for the bolts can be estimated in a similar manner.

Let us now continue to discuss in a little more detail the probability of each class occurring and the probability of objects in each class having a particular value of x. We may know, for instance, that the class of nuts is, in general, likely to occur twice as often as the class of bolts. In this case we say that the *a priori* probabilities of the two classes are:

$$\mathrm{P}(C_n) = 0.666 \text{ and } \mathrm{P}(C_b) = 0.333$$

In fact, in this case it is more likely that they will have the same *a priori* probabilities (0.5) since we usually have a nut for each bolt.

The PDFs tell us the probability that the circularity x will occur, given that the object belongs to the class of nuts C_n in the first instance and to the class of bolts C_b in the second instance. This is termed the 'conditional probability' of an object having a certain feature value, given that we know that it belongs to a particular class. Thus, the conditional probability:

$$\mathrm{P}(x \mid C_b)$$

enumerates the probability that a circularity x will occur, given that the object is a bolt. The two conditional probabilities $\mathrm{P}(x \mid C_b)$ and $\mathrm{P}(x \mid C_n)$ are shown in Figure 6.5. Of course, this is not what we are interested in at all. We want to determine the probability that an object belongs to a particular class, given that a particular value of x has occurred (i.e. been measured), allowing us to establish its identity. This is called the *a posteriori* probability $\mathrm{P}(C_i \mid x)$ that the object belongs to a

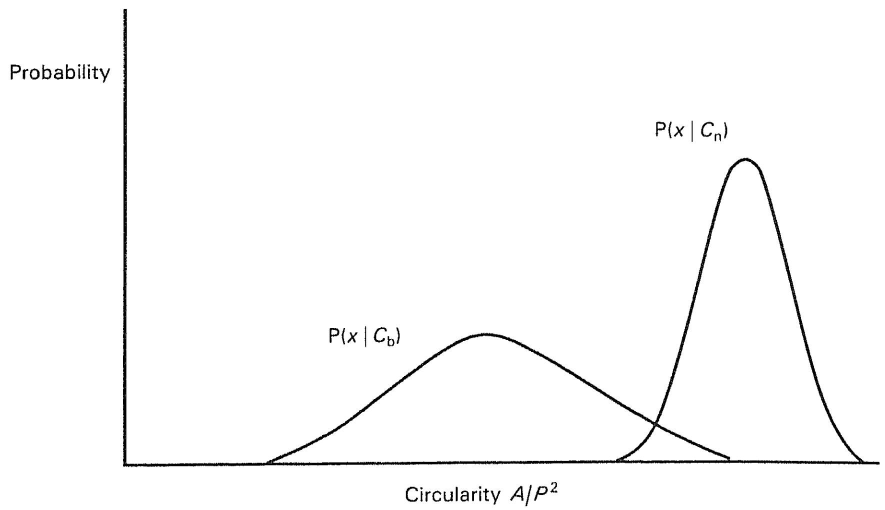

Figure 6.5 Conditional probabilities.

particular class i and is given by Bayes' theorem:

$$P(C_i \mid x) = \frac{P(x \mid C_i)\ P(C_i)}{P(x)}$$

where:

$$P(x) = \sum_{i=1}^{2} P(x \mid C_i)\ P(C_i)$$

$P(x)$ is a normalization factor which is used to ensure that the sum of the *a posteriori* probabilities sums to one, for the same reasons as mentioned above.

In effect, this Bayes' theorem allows us to use the *a priori* probability of objects occurring in the first place, the conditional probability of an object having a particular feature value given that it belongs to a particular class, and actual measurement of a feature value (to be used as the parameter in the conditional probability) to estimate the probability that the measured object belongs to a given class.

Once we can estimate the probability that, for a given measurement, the object is a nut and the probability that it is a bolt, we can make a decision as to its identity, choosing the class with the higher probability. This is why it is called the maximum likelihood classifier. Thus, we classify the object as a bolt if:

$$P(C_b \mid x) > P(C_n \mid x)$$

Using Bayes' theorem again, and noting that the normalizing factor $P(x)$ is the same for both expressions, we can rewrite this test as:

$$P(x \mid C_b)P(C_b) > P(x \mid C_n)\ P(C_n)$$

Figure 6.6 illustrates the advantage of the maximum-likelihood classifier over the nearest-neighbour classifier: if we assume that the chances of an unknown object being either a nut or a bolt are equally likely (i.e. $P(C_b) = P(C_n)$), then we classify the unknown object as a bolt if:

$$P(x \mid C_b) > P(x \mid C_n)$$

For the example shown in Figure 6.6, $P(x \mid C_b)$ is indeed greater than $P(x \mid C_n)$ for the measured value of circularity and we classify the object as a bolt. If, on the other hand, we were to use the nearest-neighbour classification technique, we would choose the class whose mean value 'is closer to' the measured value. In this case,

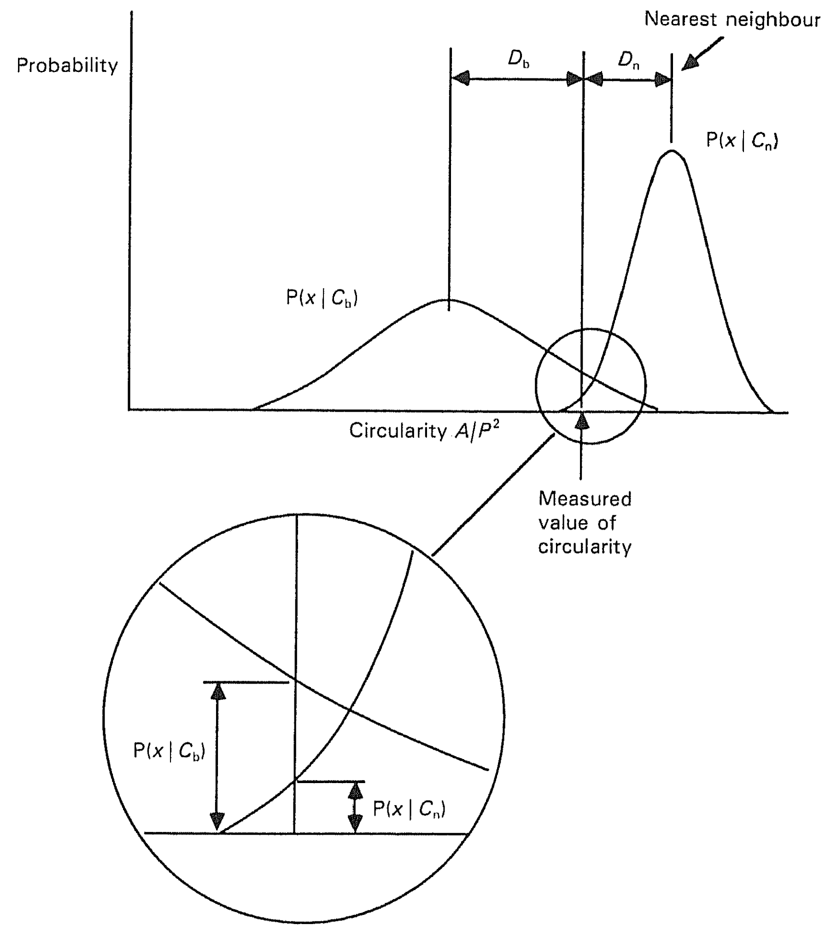

Figure 6.6 Advantage of using maximum likelihood classifier over nearest neighbour classifier.

the distance D_n from the measured value to the mean of the PDF for nuts is less than D_b, the distance from the measured value to the mean of the PDF for bolts; we would erroneously classify the object as a nut.

We have restricted ourselves to a simple example with just one feature and a one-dimensional feature space. However, the argument generalizes directly to an n-dimensional case, where we have n features, in which case the conditional probability density functions are also n-dimensional. In the two-dimensional case, the PDFs can be represented by grey-scale images: the grey-level encoding the probability.

6.4 *The Hough transform*

The Hough transform is a technique which is used to isolate curves of a given shape in an image. The classical Hough transform requires that the curve be specified in some parametric form and, hence, is most commonly used in the detection of regular curves such as lines, circles, and ellipses. Fortunately, this is not as restrictive as it might first seem since most manufactured parts do, in fact, have boundaries which are defined by such curves. However, the Hough transform has been generalized so that it is capable of detecting arbitrary curved shapes. The main advantage of this transform technique is that it is very tolerant of gaps in the actual object boundaries or curves and it is relatively unaffected by noise. We will begin by describing the classical Hough transform for the detection of lines; we will indicate how it can be applied to the detection of circles; and then we will discuss the generalized Hough transform and the detection of arbitrary shapes.

6.4.1 Hough transform for line detection and circle detection

We wish to detect a set of points lying on a straight line. The equation of a straight line is given in parametric form by the equation:

$$x \cos \phi + y \sin \phi = r$$

where r is the length of a normal to the line from the origin and ϕ is the angle this normal makes with the X-axis (refer to Figure 6.7).

If we have a point (x_i, y_i) on this line, then:

$$x_i \cos \phi + y_i \sin \phi = r$$

For a given line, r and ϕ are constant. Suppose, however, that we do not know which line we require (i.e. r and ϕ are unknown) but we do know the coordinates of the point(s) on the line. Now we can consider r and ϕ to be variable and x_i and y_i to be constants. In this case, the equation:

$$x_i \cos \phi + y_i \sin \phi = r$$

defines the values of r and ϕ such that the line passes through the point (x_i, y_i). If

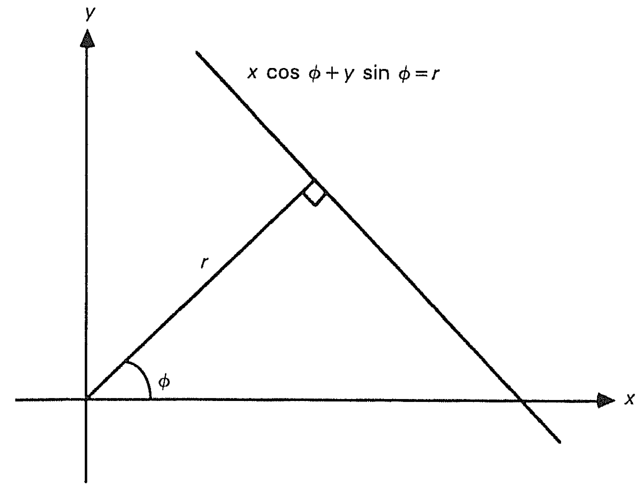

Figure 6.7 Parametric representation of a straight line.

we plot these values of r and ϕ, for a given point (x_i, y_i), on a graph (see Figure 6.8) we see that we get a sinusoidal curve in the $(r - \phi)$ space, i.e. in a space where r and ϕ are the variables. The transformation between the image plane (x- and y-coordinates) and the parameter space (r- and ϕ-coordinates) is known as the Hough transform. Thus, the Hough transform of a point in the image plane is a sinusoidal curve in the Hough $(r - \phi)$ space. However, collinear points in the image plane will give rise to transform curves which all intersect in one point since they share common r_i and ϕ_i and they all belong to the line given by:

$$x \cos \phi_i + y \sin \phi_i = r_i$$

This, then, provides us with the means to detect collinear points, i.e. lines. First of all we must sample the Hough transform space, i.e. we require a discrete representation of $(r - \phi)$ space. Since ϕ varies between 0 and 2π radians, we need only decide on the required angular resolution to define the sampling. For example, a 6° resolution on the angle of the line might suffice, in which case we will have $360^\circ/6^\circ = 60$ discrete values of ϕ. Similarly, we can limit r by deciding on the maximum distance from the origin (which is effectively going to be the maximum size of the image, 256 pixels in length, say). Our representation of $(r - \phi)$ space is now simply a two-dimensional array of size 256×60, each element corresponding to a particular value of r and ϕ: see Figure 6.9. This is called an accumulator since we are going to use it to collect or accumulate evidence of curves given by particular boundary points (x, y) in the image plane. For each boundary point (x_i, y_i) in the image we increment all accumulator cells such that the cell coordinates (r, ϕ) satisfy the equation:

$$x_i \cos \phi + y \sin \phi = r$$

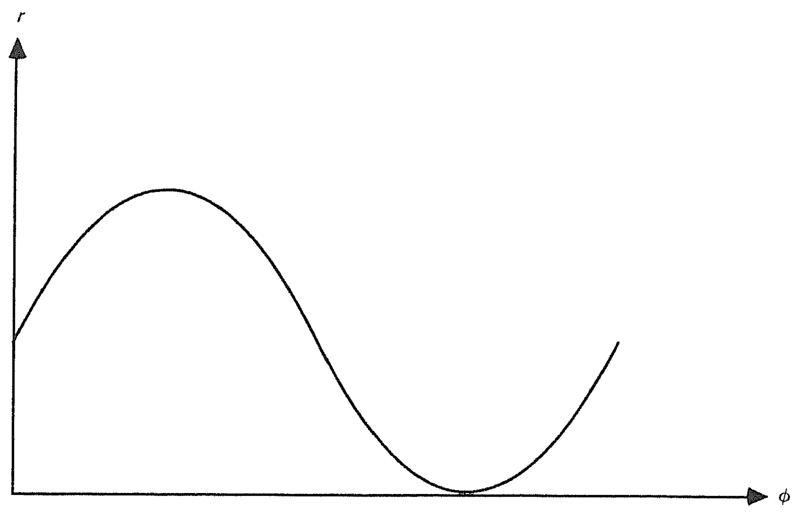

Figure 6.8 Hough transform of a point (x_i, y_i).

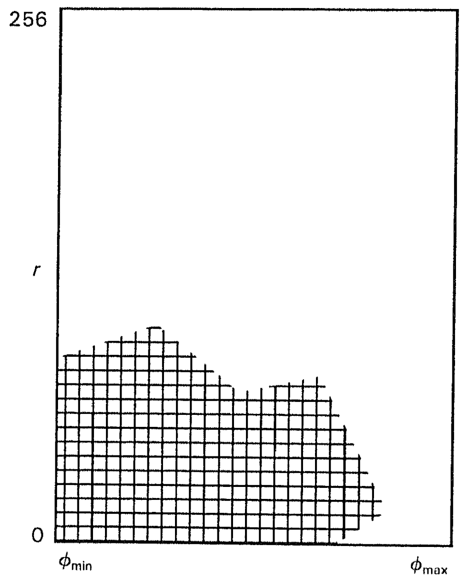

Figure 6.9 Hough transform accumulator array.

When we have done this for all available (x_i, y_i) points we can scan the accumulator searching for cells which have a high count since these will correspond to lines for which there are main points in the image plane. In fact, because there are likely to be some errors in the actual position of the x- and y-coordinates, giving rise to errors in r and ϕ, we search for clusters of points in the accumulator having high counts, rather than searching for isolated points.

Since edge detection processes are often employed in generating the candidate boundary points in the image and, in general, these yield not only the position of the edge (x_i, y_i) but also its orientation θ, where $\theta = \phi + 90°$.* We can use this information to simplify the Hough transform and, knowing x_i, y_i, and ϕ, use:

$$x_i \cos \phi + y_i \sin \phi = r$$

to compute r giving the coordinates of the appropriate accumulator cell to be incremented. The following pseudo-code summarizes this procedure:

```
/* Pseudo-code for Hough Transform: Line Detection */

• Quantize the Hough transform space: identify maximum
  and minimum values of r and φ and the total number of
  r and φ values.

• Generate an accumulator array A(r,φ); set all values
  to 0.

• For all edge points (xi, yi) in the image
  Do
   compute the normal direction φ (gradient direction
   or orientation -90°)†
   compute r from xi cos φ+yi sin φ=r
   increment A(r,φ)

• For all cells in the accumulator array
  Do
    search for maximum values
    the coordinates r and φ give the equation of the
       corresponding line in the image.
```

Just as a straight line can be defined parametrically, so can a circle. The equation of a circle is given by:

$$(x - a)^2 + (y - b)^2 = r^2$$

* Some edge detectors, e.g. the Sobel operator, are usually formulated such that they directly yield the gradient direction which is equivalent to ϕ.

† Remember to normalize the result so that it lies in the interval 0–2π.

where (a, b) are the coordinates of the centre of the circle and r is its radius. In this case, we have three coordinates in the parameter space: a, b, and r. Hence, we require a three-dimensional accumulator with an attendant increase in the computational complexity of the algorithm. Figure 6.10 illustrates the use of the Hough transform to detect the boundary between the cornea and iris of a human eye.

One further point is worth noting: the Hough transform identifies the parameter of the curve (or line) which best fits the data (the set of edge points). However, the circles that are generated are complete circles and the lines are infinite. If one wishes to identify the actual *line segments* or *curve segments* which generated these transform parameters, further image analysis will be required.

6.4.2 The generalized Hough transform

In the preceding formulation of the classical Hough transform, we used the parametric equation of the shape to effect the transform from image space to transform space. In the case where the shape we wish to isolate does not have a simple analytic equation describing its boundary, we can still use a generalized form of the Hough transform. The essential idea is that, instead of using the parametric equation of the curve, we use a look-up table to define the relationship between the boundary coordinates and orientation, and the Hough parameters. Obviously, the look-up table values must be computed during a training phase using a prototype shape.

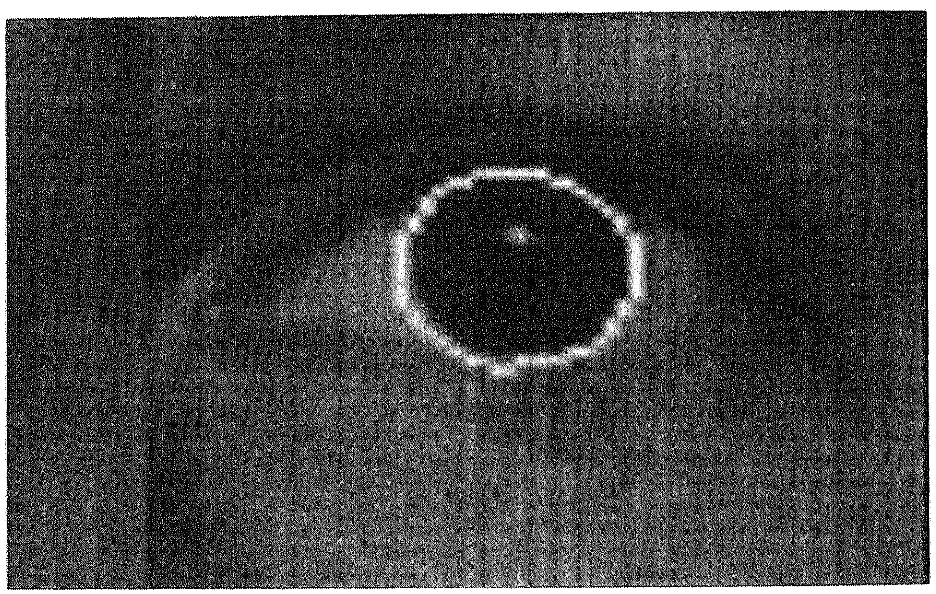

Figure 6.10 Hough transform for detection of circular shapes.

Suppose we know the shape and orientation of the required object, e.g. see Figure 6.11, the first step is to select an arbitrary reference point (x_{ref}, y_{ref}) in the object; we now define the shape in terms of the distance and angle of lines from the boundary to this reference point. For all points of the boundary, we draw a line to the reference point. We then compute the orientation of the boundary, Ω_i say, and make a note in the look-up table of the distance and direction from the boundary point to the reference point at a location in the look-up table indexed by the boundary orientation Ω_i. Since it is probable that there will be more than one occurrence of a particular orientation as we travel around the boundary, we have to make provision for more than one pair of distance and angle values. This look-up table is called an R-table.

The Hough transform space is now defined in terms of the possible positions of the shape in the image, i.e. the possible ranges of x_{ref} and y_{ref} (instead of r and ϕ in the case of the Hough transform for line detection). To perform the transform on an image we compute the point (x_{ref}, y_{ref}) from the coordinates of the boundary point, the distance r and the angle β:

$$x_{ref} = x + r \cos \beta$$
$$y_{ref} = y + r \sin \beta$$

The question is: what values of r and β do we use? These are derived from the R-table by computing the boundary orientation Ω at that point and using it as an index to the R-table, reading off all the (r, β) pairs. The accumulator array cell (x_{ref}, y_{ref}) is then incremented. We reiterate this process for all edge points in the image. As before, we infer the presence of the shape by identifying local maxima in the accumulator array.

There is just one problem: we have assumed that we know the orientation of the shape. If this is not the case, we have to extend the accumulator by incorporating an extra parameter ϕ to take changes in orientation into consideration. Thus, we now have a three-dimensional accumulator indexed by

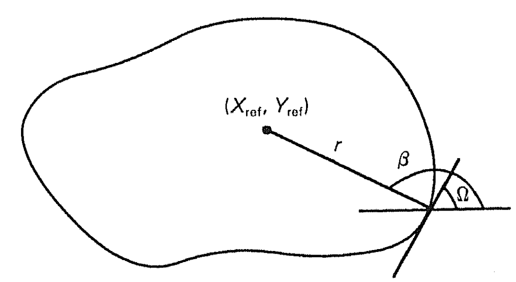

Figure 6.11 Generalized Hough transform – definition of R-table components.

(x_{ref}, y_{ref}, ϕ) and we compute:

$$x_{ref} = x + r\cos(\beta + \phi)$$
$$y_{ref} = y + r\sin(\beta + \phi)$$

for all values of ϕ and update each accumulator cell for each value of ϕ. The following pseudo-code again summarizes this procedure:

```
/* Pseudo-code for Generalized Hough Transform

• Train the shape by building the R-table:
    For all points on the boundary
      compute orientation Ω (gradient direction +90°)
      compute r and β
      add an (r, β) entry into the R-table
      at a location indexed by Ω

• Quantize the Hough transform space: identify maximum
  and minimum values of xref, yref, and φ and identify
  the total number of xref, yref, and φ values.

• Generate an accumulator array A(xref, yref, φ);
  set all values to 0.

• For all edge points (xi, yi) in the image
  Do
    compute the orientation Ω (gradient direction
  +90°)
    compute possible reference points xref, yref
    For each table entry, indexed by Ω
      For each possible shape orientation φ
        compute xref=xi+r cos (β+φ)
                yref=yi+r sin (β+φ)
        increment A(xref, yref, φ)

• For all cells in the accumulator array
  Do
    search for maximum values
  the coordinates xref, yref, and φ give the position and
  orientation of the shape in the image.
```

6.5 Histogram analysis

We began this chapter with a function classification of image analysis, noting that there are essentially three common objectives: inspection, location, and

identification. As we have seen, each of these can be tackled using various combinations of the techniques which have been discussed in Sections 6.2–6.4. It would be wrong, however, if the impression was given that this is all there is to image analysis. Indeed, in the introduction we did note that some of the more advanced techniques to be dealt with in the remaining chapters are often brought to bear. In certain circumstances, however, there are other simpler approaches which can be taken and often it is the very simplicity which makes them attractive. Such analysis techniques are often heuristic in nature, but no less useful for that. The following material on histogram analysis is intended to give a feel for this type of simple and practical approach.

The grey-level histogram of an image often contains sufficient information to allow analysis of the image content and, in particular, to discriminate between objects and to distinguish objects with defects. It has the distinct advantage that it is not necessary to segment the image first and it is not dependent on the location of the object in the image. The analysis is based exclusively on the visual appearance of the scene or image as a whole. As a simple example, consider the case where one is inspecting a bright shiny object, i.e. one which exhibits specular reflectivity. In bright field illumination, where the camera and the light source are approximately aligned, a great deal of light will be reflected and the histogram of the imaged object will be biased towards the bright end of the scale. Blemishes, or unwanted surface distortions, will tend to diffuse the light and less bright specular reflections will be imaged. In this case, the histogram will be biased more towards the dark end of the spectrum. In effect, the two cases can be distinguished by considering the distribution of grey-levels in the image, i.e. by analysis of the histogram.

There are two ways to consider histogram analysis:

(a) by extracting features which are descriptive of the shape of the histogram;
(b) by matching two histogram signatures.

In the former case, discrimination can be achieved using the classification techniques we discussed in Section 6.3, whereas in the latter case, the template matching paradigm of Section 6.2 is more appropriate.

The following (statistical) features are frequently used as a means of describing the shape of histograms:

Mean: $$\bar{b} = \sum_{b=0}^{L-1} bP(b)$$

Variance: $$\sigma_b^2 = \sum_{b=0}^{L-1} (b - \bar{b})^2 P(b)$$

Skewness: $$b_S = \frac{1}{\sigma_b^3} \sum_{b=0}^{L-1} (b - \bar{b})^3 P(b)$$

Kurtosis: $b_K = \frac{1}{\sigma_b^4} \sum_{b=0}^{L-1} (b - \bar{b})^4 P(b) - 3$

Energy: $b_N = \sum_{b=0}^{L-1} [P(b)]^2$

These features are based on 'normalized' histograms, $P(b)$, defined as:

$$P(b) = \frac{N(b)}{M}$$

where M represents the total number of pixels in the image, $N(b)$ is the conventional histogram (i.e. a function which represents the number of pixels of a given grey-level b). Note that L is the number of grey-levels in the grey-scale.

Exercises

1. Statistical pattern recognition is sometimes used in industrial vision systems, but it is not always an appropriate technique. Identify the characteristics of an application for which the approach would be suitable. Detail the component processes of the statistical pattern recognition procedure. Given that one wishes to distinguish between integrated circuit chips and other discrete components on a printed circuit board during an inspection phase of an integrated FMS (Flexible Manufacturing System), identify an appropriate feature space and describe, in detail, how the classification might be effected.
2. Two types of defects commonly occur during the manufacture of 6 cm wide strips of metal foil. They are circular (or near circular) pin-holes of various sizes and longitudinal hairline cracks. Describe how an automated visual inspection system would distinguish between these flaws, given that the metal foil is bright silver in colour.
3. Explain with the use of diagrams, how the use of Bayes' rule facilitates improved classification *vis-à-vis* the nearest-neighbour classification scheme.
4. Show how the Hough transform can be generalized to cater for arbitrary two-dimensional shapes. What are the limitations of such a generalization?

References and further reading

Dessimoz, J-D., Birk, J.R., Kelley, R.B., Martins, A.S. and Chi Lin, I. 1984 'Matched filters for bin-picking', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No. 6, pp. 686–97.

Duda, R.O. and Hart, P.E. 1973 *Pattern Classification and Scene Analysis*, New York, Wiley.

Fu, K. 1982 'Pattern recognition for automatic visual inspection', *Computer*, Vol. 15, No. 12, pp. 34–40.

Hough, P.V.C. 1962 *Methods and Measures for Recognising Complex Patterns*, US Patent 3069654, 18 December.

Illingworth, J. and Kittler, J. 1987 'The adaptive Hough transform', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 5, pp. 690–8.

Sklansky, J. 1978 'On the Hough technique for curve detection', Vol. C-27, No. 10, pp. 923–6.

Turney, J.L., Mudge, T.N. and Volz, R.A. 1985 'Recognizing partially occluded parts', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, No. 4, pp. 410–21.