picture will not be the same as the equalized histogram of a picture with some noise added to it. You cannot avoid noise in electrical systems, however well you design a system to reduce its effect. Accordingly, histogram equalization finds little use in generic image processing systems, although it can be potent in *specialized* applications. For these reasons, intensity normalization is often preferred when a picture's histogram requires manipulation.

In implementation, the function `equalize` in Code 3.3, we shall use an output range where $\mathbf{N}_{min} = 0$ and $\mathbf{N}_{max} = 255$. The implementation first determines the cumulative histogram for each level of the brightness histogram. This is then used as a LUT for the new output brightness at that level. The LUT is used to speed implementation of Equation 3.9, since it can be precomputed from the image to be equalized.

```
equalize(pic):= | range ← 255
                | number ← rows(pic).cols(pic)
                | for bright∈ 0..255
                |   pixels_at_level_bright ← 0
                | for x∈0..cols(pic)-1
                |   for y∈0..rows(pic)-1
                |     pixels_at_level_pic_y,x ← pixels_at_level_pic_y,x +1
                |
                | sum← 0
                | for level∈0..255
                |   | sum ← sum+pixels_at_level_level
                |   | hist_level ← floor[(range/number)·sum+0.00001]
                |
                | for x∈0..cols(pic)-1
                |   for y∈0..rows(pic)-1
                |     newpic_y,x ← hist_pic_y,x
                |
                | newpic
```

Code 3.3 Histogram equalization

An alternative argument against use of histogram equalization is that it is a non-linear process and is irreversible. We cannot return to the original picture after equalization, and we cannot separate the histogram of an unwanted picture. In contrast, intensity normalization is a linear process and we can return to the original image, should we need to, or separate pictures, if required.

3.3.4 Thresholding

The last point operator of major interest is called *thresholding*. This operator selects pixels that have a particular value, or are within a specified range. It can be used to find objects within a picture if their brightness level (or range) is known. This implies that the object's brightness must be known as well. There are two main forms: uniform and adaptive thresholding. In *uniform thresholding*, pixels above a specified level are set to white, those below the specified level are set to black. Given the original eye image, Figure 3.7 shows a thresholded image where all pixels *above* 160 brightness levels are set to white, and those below 160 brightness levels are set to black. By this process, the parts pertaining to the facial skin are separated from the background; the cheeks, forehead and other bright areas are separated from the hair and eyes. This can therefore provide a way of isolating points of interest.

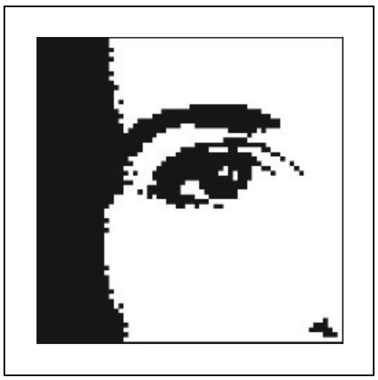

Figure 3.7 Thresholding the eye image

Uniform thresholding clearly requires knowledge of the grey level, or the target features might not be selected in the thresholding process. If the level is not known, histogram equalization or intensity normalization can be used, but with the restrictions on performance stated earlier. This is, of course, a problem of image interpretation. These problems can only be solved by simple approaches, such as thresholding, for very special cases. In general, it is often prudent to investigate the more sophisticated techniques of feature selection and extraction, to be covered later. Before that, we shall investigate group operators, which are a natural counterpart to point operators.

There are more advanced techniques, known as *optimal thresholding*. These usually seek to select a value for the threshold that separates an object from its background. This suggests that the object has a different range of intensities to the background, in order that an appropriate threshold can be chosen, as illustrated in Figure 3.8. Otsu's method (Otsu, 1979) is one of the most popular techniques of optimal thresholding; there have been surveys (Sahoo et al., 1988; Lee et al., 1990; Glasbey, 1993) which compare the performance different methods can achieve. Essentially, Otsu's technique maximizes the likelihood that the threshold is chosen so as to split the image between an object and its background. This is achieved by selecting a threshold that gives the best separation of classes, for all pixels in an image. The theory is beyond the scope of this section and we shall merely survey its results and give their implementation. The basis is use of the normalized histogram where the number of points at each level is divided by the total number of points in the image. As such, this represents a probability distribution for the intensity levels as

$$p(l) = \frac{\mathbf{N}(l)}{N^2} \tag{3.11}$$

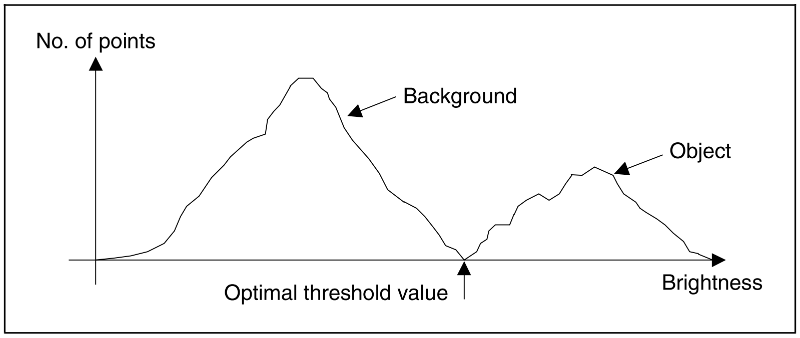

Figure 3.8 Optimal thresholding

This can be used to compute the zero- and first-order cumulative moments of the normalized histogram up to the kth level as

$$\omega(k) = \sum_{l=1}^{k} p(l) \tag{3.12}$$

and

$$\mu(k) = \sum_{l=1}^{k} l \cdot p(l) \tag{3.13}$$

The total mean level of the image is given by

$$\mu T = \sum_{l=1}^{N_{\max}} l \cdot p(l) \tag{3.14}$$

The variance of the class separability is then the ratio

$$\sigma_B^2(k) = \frac{(\mu T \cdot \omega(k) - \mu(k))^2}{\omega(k)(1-\omega(k))} \qquad \forall k \in 1, N_{\max} \tag{3.15}$$

The optimal threshold is the level for which the variance of class separability is at its maximum, namely the optimal threshold T_{opt} is that for which the variance

$$\sigma_B^2(T_{\text{opt}}) = \max_{1 \le k < N_{\max}} \left(\sigma_B^2(k)\right) \tag{3.16}$$

A comparison of uniform thresholding with optimal thresholding is given in Figure 3.9 for the eye image. The threshold selected by Otsu's operator is actually slightly lower than the value selected manually, and so the thresholded image does omit some detail around the eye, especially in the eyelids. However, the selection by Otsu is *automatic*, as opposed to *manual*, and this can be to application advantage in automated vision. Consider for example the need to isolate the human figure in Figure 3.10(a). This can be performed automatically by Otsu as shown in Figure 3.10(b). Note, however, that there are some extra points, due to illumination, which have appeared in the resulting image together with the human subject. It is easy to remove the isolated points, as we will see later, but more difficult to remove the connected ones. In this instance, the size of the human shape could be used as information to remove the extra points, although you might like to suggest other factors that could lead to their removal.

(a) Thresholding at level 160 | (b) Thresholding by Otsu (level = 127)

Figure 3.9 Thresholding the eye image: manual and automatic

The code implementing Otsu's technique is given in Code 3.4, which follows Equations 3.11–3.16 to provide the results in Figures 3.9 and 3.10. Here, the histogram function of Code 3.1 is used to give the normalized histogram. The remaining code refers directly to the earlier description of Otsu's technique.

```
                     k
ω(k,histogram):=     Σ   histogram_(l-1)
                    l=1
                     k
μ(k,histogram):=     Σ   l·histogram_(l-1)
                    l=1
                   256
μT(histogram):=     Σ   l·histogram_(l-1)
                   l=1

Otsu(image):= | image_hist ← histogram(image) / (rows(image)·cols(image))
              | for k∈1..255
              |    values_k ← (μT(image_hist)·ω(k,image_hist) - μ(k,image_hist))^2 / (ω(k,image_hist)·(1 - ω(k,image_hist)))
              | find_value(max(values),values)
```

Code 3.4 Optimal thresholding by Otsu's technique

So far, we have considered *global* techniques, methods that operate on the entire image. There are also *locally adaptive* techniques that are often used to binarize document images before character recognition. As mentioned before, surveys of thresholding are available, and one (more recent) approach (Rosin, 2001) targets thresholding of images whose histogram is unimodal (has a single peak). One survey (Trier and Jain, 1995) compares global and local techniques with reference to document image analysis. These techniques are often used in statistical pattern recognition: the thresholded object is classified according to its statistical properties. However, these techniques find less use in image interpretation, where a common paradigm is that there is more than one object in the scene, such as Figure 3.7 where the thresholding operator has