Chapter 8

Basic feature extraction

Vision systems will typically try to reduce the amount of information they have to process early on in the processing stream. Whereas an image has a size of about 1 MByte, the extraction of basic features such as edges, lines, or corners, will reduce this amount to a few kBytes. Here we try to find pixels that have the right kind of intensity (or colour) pattern around them to be lying on an edge, a line, or to correspond to a corner. Edges correspond to places in the image with an elongated discontinuity in intensity or colour. If the discontinuity has a high curvature, there is a 'corner' at that location. Lines are two edges that are close to eachother and more or less parallel. So, if an edge corresponds to a dark-bright or bright-dark transition, a line corresponds to a dark-bright-dark or bright-dark-bright transition.

This chapter discusses techniques to find these basic features. It consists of three sections. The first discusses edge extraction, both on the basis of convolution-based gradient estimation and the so-called zero-crossing approach. The second discusses the extraction of lines and introduce the matched filter principle which suggests how to design filters for other kinds of intensity patters as well. The third discusses the extraction of corners with the Harris corner detector, a method that can actually also be used to detect edges.

8.1 I - Edge pixels

Ideally, a grey level edge is a border between two regions, each of which has approximately uniform grey levels. Edges in images often result from occluding contours (outlines) of objects. In this case the two regions are projections of two different surfaces. Edges also arise from discontinuities or sharp changes in surface orientation, as well as from discontinuities in surface reflectance (e.g. markings). Taking a cross section of the image along a line at right angles to an edge, the resulting grey level profile will often only approximate a step discontinuity. Also, some edge transitions are better modeled as step transitions in the first derivative of brightness, rather than in the brightness itself. Such edges are called *ramp edges* or *roof edges*. For now, the simple step model is used to gain some insight into operators that enhance edges in images by producing high values near

them. The edge-enhanced images must be processed further to extract one-pixel thick, connected curves. This in fact is the hardest part, as we will see in chapters 9 on edge detection and 14 on segmentation.

Two lines of thought will be pursued here. The first is based on the calculation of the local intensity gradient, the second is based on the calculation of the Laplacian operator. There are several interesting alternatives available, more specifically operators with proven optimal behaviour in several respects. We cannot consider them here, however.

8.1.1 Gradient operators

Suppose the image $f(x,y)$ is sufficiently smooth as to allow the calculation of all the derivatives to be mentioned in this section. Whatever the edge detection operator, it first of all ought to be isotropic, as we have to detect edges that might have any orientation. A first approach to the problem is to look at the local change of intensity and to situate edges where this change is sufficiently large. As the image is two-dimensional we have two first-order derivatives $\partial f/\partial x$ and $\partial f/\partial y$. The gradient magnitude is then obtained as

$$\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}.$$

This is an isotropic measure. Indeed, upon rotation

$$\begin{aligned} x &= x'\cos\theta - y'\sin\theta \ , \\ y &= x'\sin\theta + y'\cos\theta \ , \end{aligned}$$

where (x,y) are the unrotated and (x',y') are the rotated coordinates.

$$\begin{aligned} \frac{\partial f}{\partial x'} &= \frac{\partial f}{\partial x}\frac{\partial x}{\partial x'} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial x'} = \frac{\partial f}{\partial x}\cos\theta + \frac{\partial f}{\partial y}\sin\theta \ , \\ \frac{\partial f}{\partial y'} &= \frac{\partial f}{\partial x}\frac{\partial x}{\partial y'} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial y'} = -\frac{\partial f}{\partial x}\sin\theta + \frac{\partial f}{\partial y}\cos\theta \ . \end{aligned}$$

Whereas the partial derivatives themselves are clearly not rotation invariant, the magnitude of the gradient is easily shown to be invariant:

$$\sqrt{\left(\frac{\partial f}{\partial x'}\right)^2 + \left(\frac{\partial f}{\partial y'}\right)^2} = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Note that we can just as well use the magnitude squared. The direction for which the change in intensity is steepest is given by the θ that extremizes $\partial f/\partial x'$. Simple differentiation with respect to θ yields

$$\theta_{extr} = \tan^{-1}\left(\frac{\partial f}{\partial y}\Big/\frac{\partial f}{\partial x}\right) + k\pi \ ,$$

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

Figure 8.1: *Sobel convolution masks: left mask for the detection of vertical edges, right similar mask for the detection of horizontal edges.*

with k an arbitrary integer. For θ_{extr} we find

$$\begin{aligned} \cos\theta_{extr} &= \pm\frac{\partial f}{\partial x}/\sqrt{\left(\frac{\partial f}{\partial x}\right)^2+\left(\frac{\partial f}{\partial y}\right)^2} \\ \sin\theta_{extr} &= \pm\frac{\partial f}{\partial y}/\sqrt{\left(\frac{\partial f}{\partial x}\right)^2+\left(\frac{\partial f}{\partial y}\right)^2} \end{aligned}$$

and thus the extrema of $\partial f/\partial x'$ are given by the gradient:

$$\frac{\partial f}{\partial x}\cos\theta_{extr}+\frac{\partial f}{\partial y}\sin\theta_{extr}=\pm\sqrt{\left(\frac{\partial f}{\partial x}\right)^2+\left(\frac{\partial f}{\partial y}\right)^2}\ .$$

The gradient is obtained by a non-linear operation. The partial differentiations $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$, however, correspond to linear and shift-invariant operations. These can therefore be written as a simple convolution. Theoretically, they should be calculated in an infinitesimally small neighbourhood. The corresponding convolution kernels are so-called "doublets", not real functions but mathematical abstractions of the Dirac impulse type. In practice, both derivatives will be estimated on the basis of finite differences. They can be calculated as convolution with simple masks. The simplest possibility would be to approximate $\frac{\partial f}{\partial x}$ by $f(x+1,y)-f(x,y)$ and $\frac{\partial f}{\partial y}$ by $f(x,y+1)-f(x,y)$, assuming that the pixel dimension is the unit of length. The disadvantage would be that it is difficult to assign these values to an actual pixel, because they would best serve the midpoints between the pixels used in the calculations. Secondly, these expressions would be noise sensitive. Therefore, one has introduced larger masks with odd dimensions, often 3×3. Such masks are illustrated in fig. 8.1. The mask at the left will yield high outputs (absolute values) where vertical edges can be found, the mask at the right yields high outputs for horizontal edges. The outputs of the convolution masks are then combined according to the formulas given above, in order to obtain edge orientation and gradient magnitude. This particular set of masks are the so-called "Sobel masks". In principle, their outputs should be divided by 8 in order to estimate the partial derivatives. In practice, we are only interested in the relative outputs of both masks. The "Prewitt masks" are obtained by replacing all 2's by 1's. Both mask pairs can be implemented very efficiently. They have simple coefficients that can all be realized without multiplication (a 2 would simply require a shift). Moreover, they are separable. In fact, one could regard them as three alligned finite difference schemes, this time with 2 pixels in between, one placed besides the other. So first, we calculate the finite difference with a $(-1,0,1)$ or $(-1,0,1)^T$ mask depending

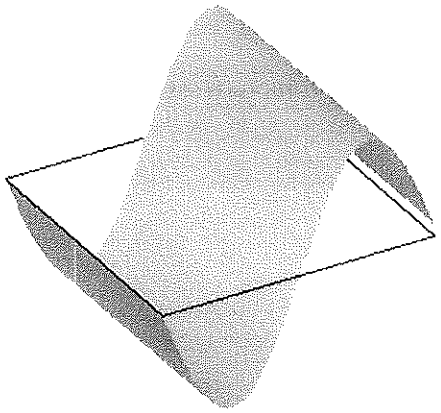


Figure 8.2: *MTF for a finite difference approximation of the first derivative.*

on the partial derivative. The MTF of such filter is shown in fig. 8.2. Secondly, we smooth these results by a $(1,2,1)^T$ or $(1,2,1)$ binomial mask for the Sobel pair, respectively. For the Prewitt convolutions masks this second step would consist of convolving the results with $(1,1,1)^T$ or $(1,1,1)$ instead. The effect is one of noise suppression when compared to smaller convolution masks.

It is interesting to see how these (linear) filters modify the different frequency components of the image. Due to the separability of the masks and their simple structure (only consisting of Dirac impulses), we can immediately write down their modulation transfer functions. Convolving the image with these filters has the effect of multiplying the Fourier transform of the image with the corresponding modulation transfer functions (cf. the convolution theorem, given in chapter 5). For the vertical Sobel mask (i.e. the mask at the left in fig. 8.1), we find the modulation transfer function

$$(2i \sin 2\pi u)(2 \cos 2\pi v + 2).$$

This is a pure imaginary function, resulting in phase shifts over $\pi/2$. Its magnitude (power spectrum) is shown in fig. 8.3, with the boundaries corresponding to $u = -\frac{1}{2}$, $u = \frac{1}{2}$, and $v = -\frac{1}{2}$, $v = \frac{1}{2}$. Two lobes occur in the u-direction and in between the power spectrum vanishes, indicating that *dc* is completely eliminated. In this direction, the filter emphasises a band of higher frequencies, thereby highlighting the position of vertical edges. The v-direction is characterized by one lobe, centered around DC. In this direction the filter has a low-pass (smoothing) effect.

Fig. 8.4 shows the use of this type of "gradient masks". At the top left the original image is shown. At the bottom two filtered images are shown, each the output of one of the Sobel masks (gray corresponds to zero, bright pixels have positive values, dark pixels have negative values). One of the filters highlights vertical edges, the other one horizontal edges, as expected. The top right image combines these results and shows the gradient magnitude. As can be seen from this example, edges are not always equally salient in the

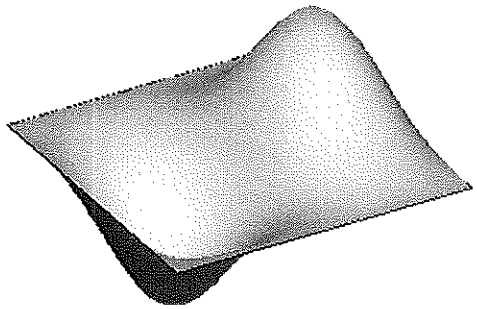


Figure 8.3: *The vertical Sobel filter's MTF.*

resulting image. At some edge pixels gradient magnitude is rather low. Also are edges at several places smeared out over several pixel wide ribbons rather than one-pixel thick lines. The Sobel masks are the optimal 3×3 filters with integer coefficients for the detection of step edges (i.e. edges arising at discontinuities in intensity that separate two regions of constant intensity). They will yield acceptable approximations to both edge contrast and orientation.

8.1.2 Zero-crossings

As a second possibility, we could consider edges to lie on the inflection points of the intensity function $f(x, y)$. Such points can be found at the zero-crossings of the Laplacian:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0.$$

The Laplacian clearly is a linear and shift-invariant operator. Therefore we can implement it as a simple convolution operation. It is also isotropic, as is easy to check. In fact, the Laplacian is the simplest linear, isotropic operator. A discrete version, based on finite difference approximations of the second derivatives, is given in fig. 8.5, on the left. The corresponding MTF is

$$2\cos(2\pi u) + 2\cos(2\pi v) - 4$$

and is shown in fig. 8.6 on the left. A more isotropic approximation is the mask in fig. 8.5. The better isotropy is corroborated by the MTF, shown in fig. 8.6 on the right.

Because this operator is based on second-order derivatives, it is highly sensitive to noise. Usually, the operator should therefore be combined with a smoothing operation, e.g. convolution with a Gaussian filter. If G represents the Gaussian convolution filter, and L the convolution filter corresponding to the Laplacian, then $L*(G*f) = (L*G)*f$ and

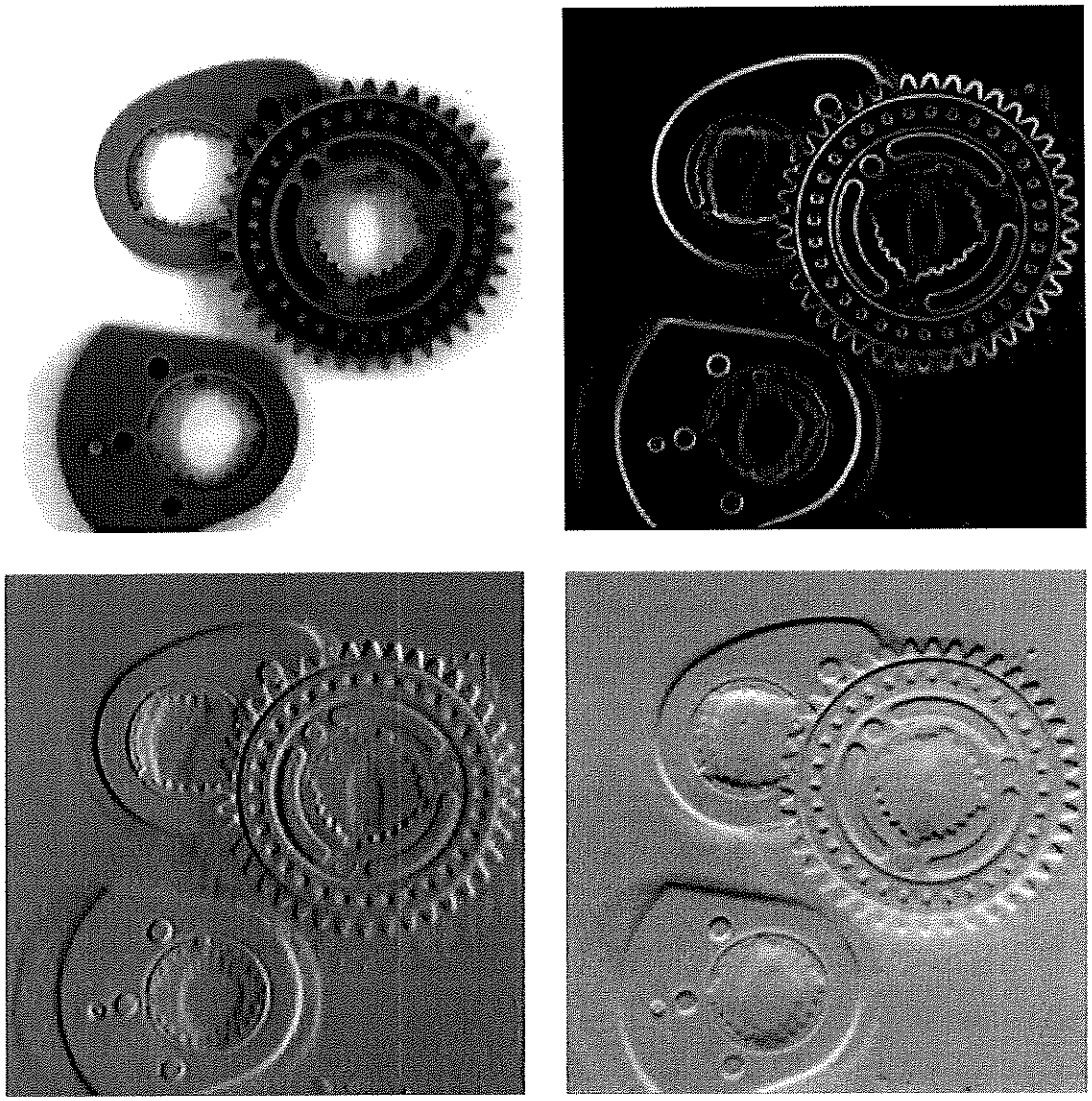


Figure 8.4: *The top left image is processed with the two Sobel masks, yielding the outputs at the bottom. The gradient magnitude combines these outputs and is shown in the top right image.*

0	1	0
1	- 4	1
0	1	0

1	2	1
2	-12	2
1	2	1

Figure 8.5: *Two discrete approximations for the Laplacian operator.*

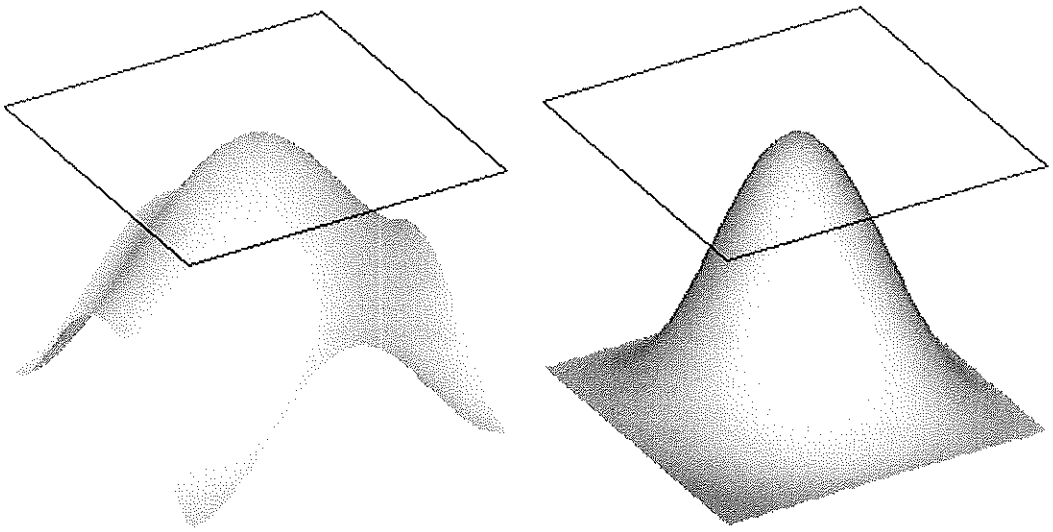


Figure 8.6: *MTFs for the discrete Laplacian masks of fig. 8.5.*

we see that we can combine both operations into a single convolution with a (usually large) mask $L * G$. The profile of such convolution filter is shown in fig. 8.7. Such filter is also often referred to as the "Mexican hat" filter or "Marr-Hildreth edge detector" when applied for edge detection purposes. An example of the zero-crossings obtained with such filter is shown in fig. 8.8 Moreover, the Laplacian (of Gaussian) filter can be well approximated by a difference of a narrow Gaussian and a wide Gaussian (with the same volume underneath) and such approximation is referred at as DOG filter (Difference Of Gaussians).

The use of the Laplacian has often been advocated with the argument that it yields closed contours for most intensity functions. But it turns out that the edges found this way often do not really correspond to what a human observer would regard as an edge. Closure of edges is sometimes rather artificial. This phenomenon is clearly visible in the example of fig. 8.8. Some real edges have disappeared, whereas spurious edges have been generated. Zero-crossings are found where neighbouring pixels have opposite signs for the output. As can be seen, this operation leads to one-pixel thick edges, as opposed to the result obtained with the Sobel filters and other gradient masks.

8.2 II - Line pixels, matched filter theorem

It is interesting to note that the Laplacian type of filters can also be used to detect spots, i.e. dark blobs on bright backgrounds or v.v.. In that case we should not look at zero-crossings, but at extrema of the filter output. Similarly, other types of structures can be found with appropriate convolution filters. A filter for detecting horizontal lines might e.g.

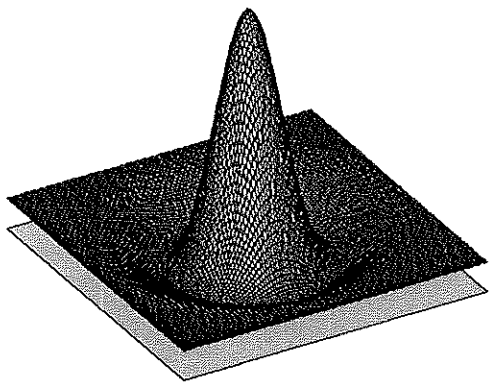


Figure 8.7: *Sign reversed profile of Laplacian of Gaussian filter.*

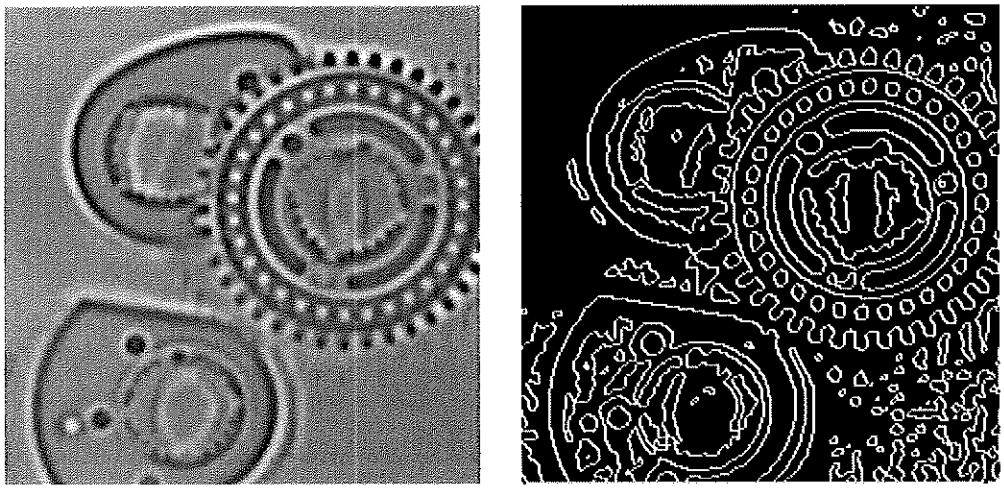


Figure 8.8: *Left original image and right the zero-crossings found with a Laplacian of Gaussian filter. Edges are closed but do not always correspond to structures that humans would regard as edges.*

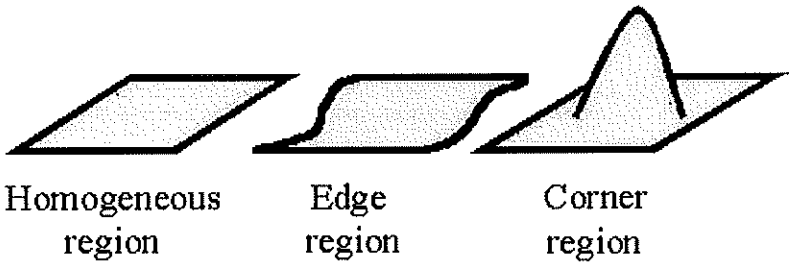


Figure 8.9: *Typical intensity profiles for (a) a flat region, (b) an edge region, and (c) a corner region.*

look like

$$\begin{matrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{matrix}$$

Notice that so far, irrespective of the pattern we would be looking for (an edge, a spot, a line, $\cdots$), the corresponding convolution filter coefficients reflect the pattern of intensities corresponding to such structures. This is in line with the so-called *matched filter theorem.* This theorem states that the best convolution mask (in this context often called a template) to find a certain reference pattern in a signal, will often be that reference pattern itself. The "often" in the formulation indicates that certain conditions have to be fulfilled. The signal over noise ratio for such filter will e.g. be maximal if the noise is white, i.e. if the noise has a flat spectrum.

Text on matched filter theorem not available yet

8.3 II - Corners

As mentioned before, convolution – as a linear operation – is not necessarily the most appropriate type of processing. Here we present an approach that analyses the overall shape of the local intensity profile. In doing so it is able to distinguish between three types: flat regions, edges, and corners. The three types of profiles are shown in fig. 8.9. Flat regions show little variations in intensity. Edges correspond to intensity profiles with major changes in one direction and small changes in the orthogonal direction. Corner regions have large changes in two orthogonal directions.

A mathematical approach to classify the intensity profiles into these three categories is the following. First, in an environment of the pixel under scrutiny the second order moments of the intensity variations $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ are calculated, i.e. the following averages are calculated locally:

$$\begin{pmatrix} \widehat{\left(\frac{\partial f}{\partial x}\right)^2} & \widehat{\frac{\partial f}{\partial x}\frac{\partial f}{\partial y}} \\ \widehat{\frac{\partial f}{\partial x}\frac{\partial f}{\partial y}} & \widehat{\left(\frac{\partial f}{\partial y}\right)^2} \end{pmatrix}$$

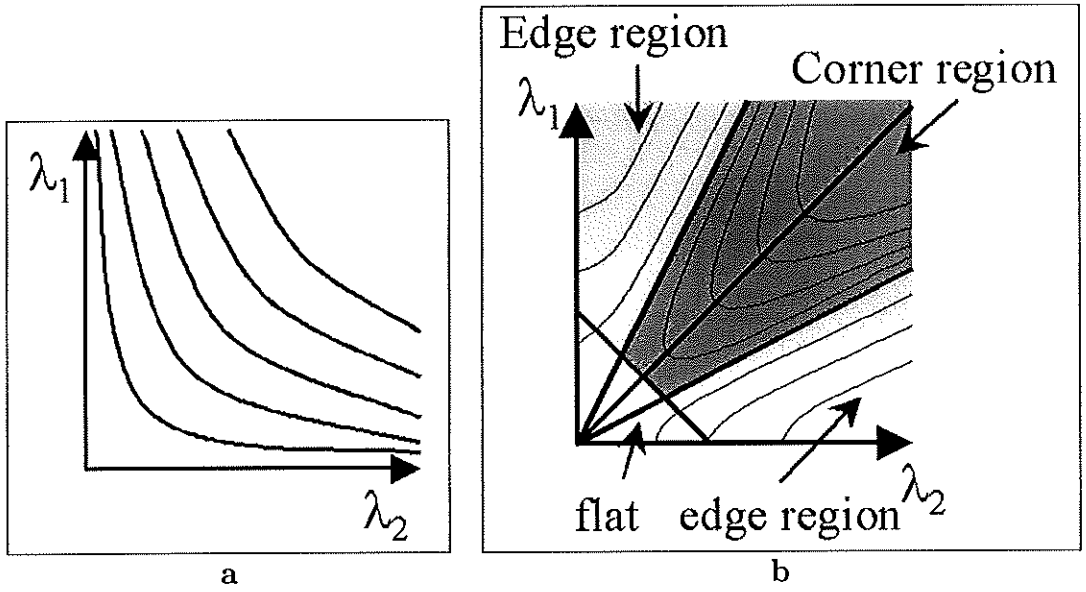
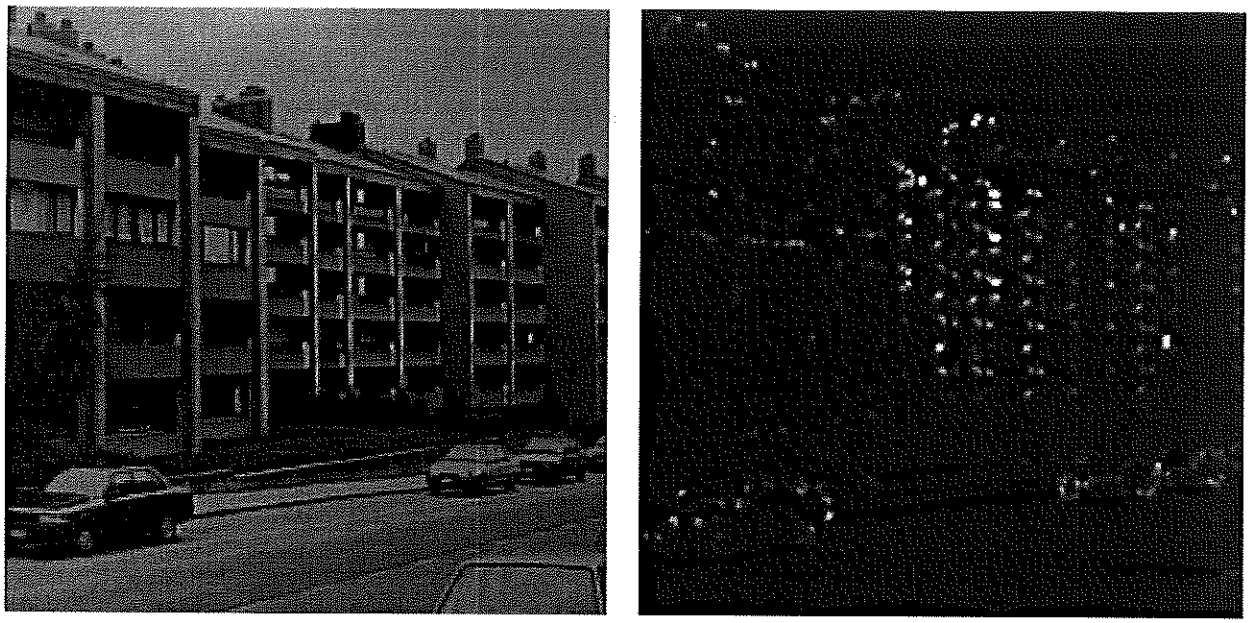


Figure 8.10: *(a) Iso-determinant lines, with corners on one side of a chosen line. (b) Refined criterion, with separate regions for flat regions, edge regions and corner regions, as indicated.*

Then, the direction of the largest changes in the intensity is calculated as the eigenvector of this matrix with the largest corresponding eigenvalue. This eigenvalue quantifies the size of observed changes. This analysis is based on so-called Principal Component Analysis, for which a mathematical justification is given in chapter 6. Similarly, the other eigenvector, with the smallest eigenvalue, yields the direction of least changes in the intensity. This direction is orthogonal to the first. Again, the eigenvalue for this direction quantifies the amount of change observed in that direction.

Summarizing, regions with homogeneous intensity have two small eigenvalues, edge regions have one small and one large eigenvalues, and corner regions have two large eigenvalues. A possible criterion therefore is to take the determinant of the 2nd-order moment matrix, which corresponds with the product of the two eigenvalues. Its value will only be large if both eigenvalues are large. Fig. 8.10(a) shows the hyperbolae that correspond to iso-lines for the value of the determinant $\lambda_1\lambda_2$, with λ_1 and λ_2 the two eigenvalues. Some threshold value can then be chosen and regions with the determinant above that value are considered 'corners'. In the figure this corresponds to positions to the right of the hyperbola for the threshold value. The problem is that cases where one of the eigenvalues is very large and the other rather small can still yield high values for the determinant. These are cases one would like to exclude from the corner category. One way to achieve this is to use iso-lines for $Determinant - k(Trace)^2$. These are illustrated in fig. 8.10(b). Three areas are distinguished: one for flat intensity profiles where both eigenvalues are small. This is the white region near the origin. The corner area has two large eigenvalues and corresponds to the darkest area. Finally, two disjoint regions – light grey in the figure



a b

Figure 8.11: *(a) Example image for the application of the Harris detector. (b) Output of the Harris detector. Positive values (bright) correspond to corners, values close to 0 (grey) to homogeneous regions, and negative values (dark) are found at the edges.*

– correspond to cases where one of the eigenvalues is small and the other one is large, i.e. edge regions in the image. Following this strategy, every pixel in an image can be classified accordingly, depending on the intensity profile in its neighbourhood. This is the so-called *Harris operator*, often used to specifically find the corners (*Harris corner detector*).

As an example, the Harris operator was applied to the image of fig. 8.11(a). The result is shown in fig. 8.11(b). Dark intensities correspond to negative values, bright intensities to positive values. As can be observed from this example, pixels at edge positions are dark, pixels at corner positions are bright, and most of the scene, where the intensity profile is almost flat, has values around 0 (grey). In this example $k = 0.04$. Typically, the next step is to look for local extrema of the operator's output, in order to localise the corners and edges with better precision.

8.4 Further reading

Not available yet