lines in images as ridges and ravines using the facet model [Haralick and Shapiro, 1992]. Line detection is frequently used in remote sensing and in document processing; examples include [Venkateswar and Chellappa, 1992; Tang et al., 1997].

Local information about edges is the basis of a class of image segmentation techniques that are discussed in Chapter 6. Edges which are likely to belong to object boundaries are usually found by simple thresholding of the edge magnitude—such edge thresholding does not provide ideal contiguous boundaries that are one pixel wide. Sophisticated segmentation techniques that are dealt with in the next chapter serve this purpose. Here, much simpler edge thinning and filling methods are described. These techniques are based on knowledge of small local neighborhoods and are very similar to other local pre-processing techniques.

Thresholded edges are usually wider than one pixel, and **line thinning** techniques may give a better result. One line thinning method uses knowledge about edge orientation and in this case edges are thinned before thresholding. Edge magnitudes and directions provided by some gradient operator are used as input, and the edge magnitudes of two neighboring pixels perpendicular to the edge direction are examined for each pixel in the image. If at least one of these pixels has edge magnitude higher than the edge magnitude of the examined pixel, then the edge magnitude of the examined pixel is assigned a zero value. This technique is called **non-maximal suppression** and is similar to the idea mentioned in conjunction with the Canny edge detector.

There are many line thinning methods which we do not present here. In most cases the best line thinning is achieved using mathematical morphology methods which are explained in Chapter 13.

5.3.10 Detection of corners (interest points)

In many cases it is advantageous to find pairs of corresponding points in two similar images; we came across this fact in Section 5.2 when considering geometric transforms. Knowing the position of corresponding points enables the estimation of parameters describing geometric transforms from live data. The same transformation usually holds for almost all pixels of the image. The necessary number of corresponding pairs of points is usually rather small and is equal to the number of parameters of the transform. We shall see later on that finding corresponding points is also a core problem in the analysis of moving images (Chapter 16), and for recovering depth information from pairs of stereo images (Section 11.5).

In general, all possible pairs of pixels should be examined to solve this **correspondence problem**, and this is very computationally expensive. If two images have n pixels each, the complexity is $\mathcal{O}(n^2)$. This process might be simplified if the correspondence is examined among a much smaller number of points, called **interest points**. An interest point should have some typical local property [Ballard and Brown, 1982]. For example, if square objects are present in the image, then **corners** are very good interest points. In this section, two representatives from the wide class of interest points detectors will be explained. The first are corner detectors and the second one are the more rich structures called MSERs (Maximally Stable Extremal Regions).

Corners in images can be located using local detectors; input to the corner detector is the gray-level image, and output is the image in which values are proportional to the likelihood that the pixel is a corner. **Interest points** are obtained by thresholding the result of the corner detector.

Corners serve better than lines when the correspondence problem is to be solved. This is due to the **aperture problem**. Assume a moving line is seen through a small aperture. In such a case, only the motion vector perpendicular to the line can be observed. The component collinear with the line remains invisible. The situation is better with corners. They provide ground for unique matching, see Figure 5.34 for illustration.

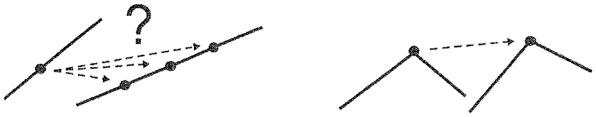

Figure 5.34: Ambiguity of lines for matching and unambiguity of corners.

Edge detectors themselves are not stable at corners. This is natural as the gradient at the tip of the corner is ambiguous. This is illustrated in Figure 5.35 in which a triangle with sharp corner is shown. However, near the corner there is a discontinuity in the gradient direction. This observation is used in corner detectors.

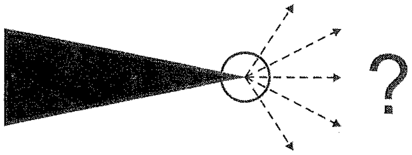

Figure 5.35: Ambiguity of edge detector at the corner tip.

The **corner** in the image can also be defined as a pixel in its small neighborhood where are two dominant and different edge directions. This definition is not precise as an isolated point of local intensity maximum or minimum, line endings, or an abrupt change in the curvature of a curve gives a response similar to a corner. Nevertheless, such detectors are named corner detectors in the literature and are widely used. If corners have to be detected then some additional constraints have to be applied.

Corner detectors are not usually very robust. This deficiency is overcome either by manual expert supervision or large redundancies introduced to prevent the effect of individual errors from dominating the task. The latter means that many more corners are detected in two or more images than are needed for estimating a transformation sought between these images.

The simplest corner detector is the **Moravec detector** [Moravec, 1977] which is maximal in pixels with high contrast. These points are on corners and sharp edges. The Moravec operator MO is given by

$$\mathrm{MO}(i,j) = \frac{1}{8} \sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} \left| f(k,l) - f(i,j) \right| . \tag{5.71}$$

Better results are produced by computationally more expensive corner operators such as those proposed by Zuniga–Haralick [Zuniga and Haralick, 1983; Haralick and Shapiro, 1992] or Kitchen–Rosenfeld [Huang, 1983] which are based on the facet model (Section 5.3.6). The image function f is approximated in the neighborhood of the pixel (i,j) by a cubic polynomial with coefficients c_k:

The Zuniga–Haralick operator ZH is given by

$$\mathrm{ZH}(i,j) = \frac{-2\left(c_2^2\, c_6 - c_2\, c_3\, c_5 + c_3^2\, c_4\right)}{\left(c_2^2 + c_3^2\right)^{3/2}} \,. \tag{5.73}$$

The Kitchen-Rosenfeld KR operator has the same numerator as equation (5.73), but the denominator is $(c_2^2 + c_3^2)$. The ZH operator has been shown to outperform the KR corner detector in test images [Haralick and Shapiro, 1992].

The **Harris corner detector** [Harris and Stephen, 1988] improved upon Moravec's by considering the differential of the corner score (sum of square differences). Consider a 2D gray-scale image f. An image patch $W \in f$ is taken and is shifted by $\Delta x, \Delta y$. The sum of square differences S between values of the image f given by the patch W and its shifted variant by Δx, Δy is given by:

$$S_W(\Delta x, \Delta y) = \sum_{x_i \in W} \sum_{y_i \in W} \left(f(x_i, y_i) - f(x_i - \Delta x, y_i - \Delta y)\right)^2 . \tag{5.74}$$

A corner point not suffering from the aperture problem must have a high response of $S_W(\Delta x, \Delta y)$ for all Δx, Δy. If the shifted image patch is approximated by the first-order Taylor expansion

$$f(x_i - \Delta x, y_i - \Delta y) \approx f(x_i, y_i) + \left[\frac{\partial f(x_i, y_i)}{\partial x}, \frac{\partial f(x_i, y_i)}{\partial y}\right] \begin{bmatrix}\Delta x \\ \Delta y\end{bmatrix} , \tag{5.75}$$

then the minimum of $S_W(\Delta x, \Delta y)$ can be obtained analytically. Substitute this approximation given by equation (5.75) into equation (5.74)

$$\begin{aligned}
S(x,y) &= \sum_{x_i \in W} \sum_{y_i \in W} \left(f(x_i, y_i) - f(x_i, y_i) - \left[\frac{\partial f(x_i, y_i)}{\partial x}, \frac{\partial f(x_i, y_i)}{\partial y}\right] \begin{bmatrix}\Delta x \\ \Delta y\end{bmatrix}\right)^2 \\
&= \sum_{x_i \in W} \sum_{y_i \in W} \left(-\left[\frac{\partial f(x_i, y_i)}{\partial x}, \frac{\partial f(x_i, y_i)}{\partial y}\right] \begin{bmatrix}\Delta x \\ \Delta y\end{bmatrix}\right)^2 \\
&= \sum_{x_i \in W} \sum_{y_i \in W} \left(\left[\frac{\partial f(x_i, y_i)}{\partial x}, \frac{\partial f(x_i, y_i)}{\partial y}\right] \begin{bmatrix}\Delta x \\ \Delta y\end{bmatrix}\right)^2 \\
&\qquad \text{having in mind that } \mathbf{u}^2 = \mathbf{u}^\top \mathbf{u} \\
&= \sum_{x_i \in W} \sum_{y_i \in W} [\Delta x, \Delta y] \left(\begin{bmatrix}\frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y}\end{bmatrix} \begin{bmatrix}\frac{\partial f}{\partial x} & \frac{\partial f}{\partial y}\end{bmatrix}\right) \begin{bmatrix}\Delta x \\ \Delta y\end{bmatrix} \\
&= [\Delta x, \Delta y] \left(\sum_{x_i \in W} \sum_{y_i \in W} \begin{bmatrix}\frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y}\end{bmatrix} \begin{bmatrix}\frac{\partial f}{\partial x} & \frac{\partial f}{\partial y}\end{bmatrix}\right) \begin{bmatrix}\Delta x \\ \Delta y\end{bmatrix} \\
&= [\Delta x, \Delta y]\, A_W(x,y) \begin{bmatrix}\Delta x \\ \Delta y\end{bmatrix} ,
\end{aligned}$$

where the Harris matrix $A_W(x,y)$ is the second derivative of S around the point $(x,y) = (0,0)$. A is

$$A(x,y) = \begin{bmatrix} \sum\limits_{x_i \in W} \sum\limits_{y_i \in W} \frac{\partial^2 f(x_i,y_i)}{\partial x^2} & \sum\limits_{x_i \in W} \sum\limits_{y_i \in W} \frac{\partial f(x_i,y_i)}{\partial x} \frac{\partial f(x_i,y_i)}{\partial y} \\ \sum\limits_{x_i \in W} \sum\limits_{y_i \in W} \frac{\partial f(x_i,y_i)}{\partial x} \frac{\partial f(x_i,y_i)}{\partial y} & \sum\limits_{x_i \in W} \sum\limits_{y_i \in W} \frac{\partial^2 f(x_i,y_i)}{\partial y^2} \end{bmatrix} . \qquad (5.76)$$

Usually an isotropic window is used, such as a Gaussian. The response will be isotropic too.

The local structure matrix A represents the neighborhood—the Harris matrix A is symmetric and positive semi-definite. Its main modes of variation correspond to partial derivatives in orthogonal directions and are reflected in eigenvalues λ_1, λ_2 of matrix A. These modes of variations can be found using principal component analysis (PCA), see Section 3.2.10. Three distinct cases can appear:

1. Both eigenvalues are small. This means that image f is flat in the examined pixel. There are no edges or corners in this location.
2. One eigenvalue is small and the second one large. The local neighborhood is ridge-shaped. Significant change of image f occurs if a small movement is made perpendicularly to the ridge.
3. Both eigenvalues are rather large. A small shift in any direction causes significant change of image f. A corner is found.

Cases 2 and 3 are illustrated in Figure 5.36.

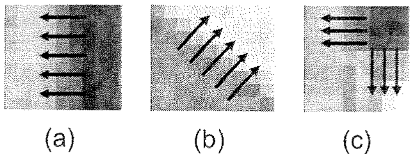

Figure 5.36: Illustration of the decision within Harris corner detector according to eigenvalues of the local structure matrix. (a), (b) Ridge detected, no corner at this position. (c) Corner detected.

Harris suggested that exact eigenvalue computation can be avoided by calculating the response function $R(A) = \det(A) - \kappa \, \mathrm{trace}^2(A)$, where $\det(A)$ is the determinant of the local structure matrix A, trace(A) is the trace of matrix A (sum of elements on the main diagonal), and κ is a tunable parameter where values from 0.04 to 0.15 were reported in literature as appropriate.

An example of Harris corners applied to a real scene is in Figure 5.37. Corners are marked by red crosses.

Algorithm 5.5: Harris corner detector

1. Filter the image with a Gaussian.
2. Estimate intensity gradient in two perpendicular directions for each pixel, $\frac{\partial f(x,y)}{\partial x}$, $\frac{\partial f(x,y)}{\partial y}$. This is performed by twice using a 1D convolution with the kernel approximating the derivative.

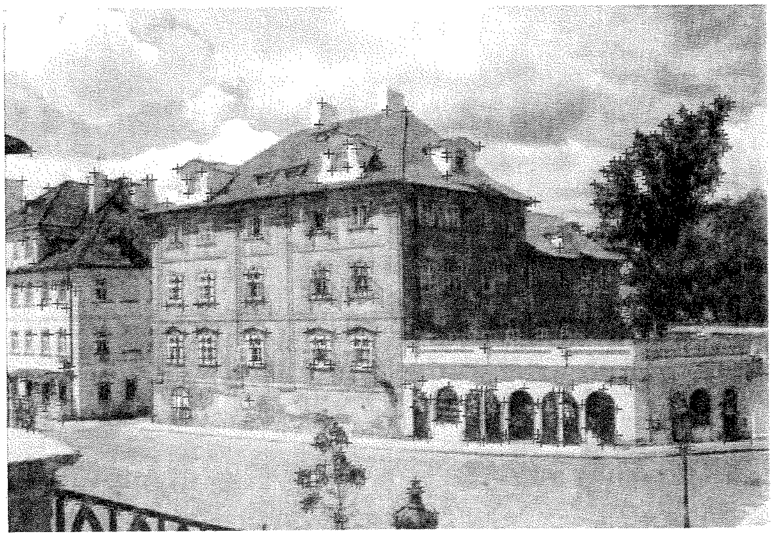

Figure 5.37: Example of Harris corners in the image. *Courtesy of Martin Urban, Czech Technical University in Prague, who used such images for 3D reconstruction. A color version of this figure may be seen in the color inset—Plate 7.*

- Calculate the local structure matrix A.
- Evaluate the response function $R(A)$.

4. Choose the best candidates for corners by selecting a threshold on the response function $R(A)$ and perform non-maximal suppression.

The Harris corner detector has been very popular. Its advantages are insensitivity to 2D shift and rotation, to small illumination variations, to small viewpoint change, and its low computational requirements. On the other hand, it is not invariant to larger scale change, viewpoint changes and significant changes in contrast.

Many more corner-like detectors exist, and the reader is referred to the overview papers [Mikolajczyk and Schmid, 2004], [Mikolajczyk et al., 2005].

5.3.11 Detection of maximally stable extremal regions

The Harris interest point detector is an example of an algorithm producing output that is invariant with rotation and translation in this case. When effects of noise and discretization are negligible, the output of a Harris detector on a rotated and/or translated image is a rotated and/or translated set of points. However, if the image is rescaled or transformed projectively, the output of Harris detector changes rapidly.

Maximally Stable Extremal Regions (MSERs) [Matas et al., 2002] are an example of an image structure that can be repeatably detected not only after translations and rotations, but also after similarity and affine transforms of an image.

The MSER detection process can be explained informally as follows. Imagine all possible thresholdings of an input gray-level image I, say with a common range $\mathcal{S} = [0, 1, \ldots, 255]$. We will refer to the pixels below a threshold as 'black' and to those above

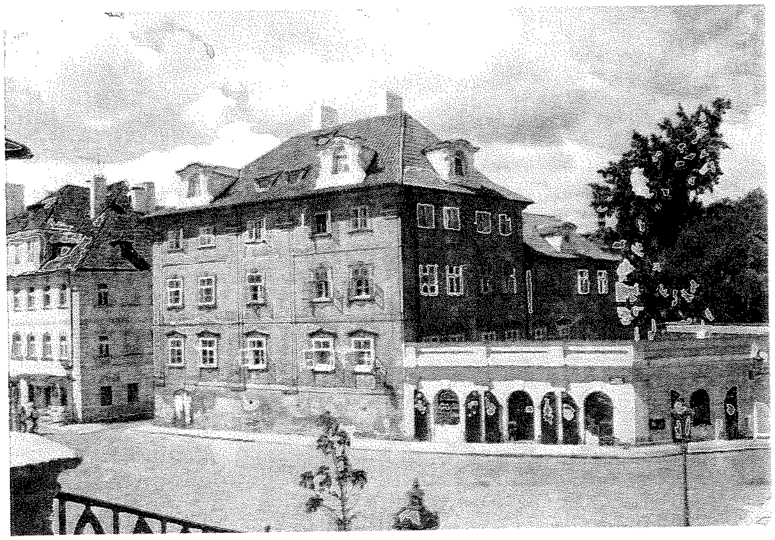

Figure 5.38: Example of MSERs detected in the image. Regions with the red border are results of the algorithm on the increasingly ordered list of intensities. Regions with the green borders come from the list with decreasing ordering. *Courtesy of Jiri Matas, Czech Technical University in Prague, who used such images for 3D reconstruction. A color version of this figure may be seen in the color inset—Plate 8.*

or equal as 'white'. If we were shown a movie of thresholded images I_t, with frame t corresponding to threshold t, we would see first a white image. Subsequently black spots corresponding to local intensity minima will appear and grow. At some point regions corresponding to two local minima will merge. Finally, the last image will be black. The union of all connected components of all frames of the movie is identical to the set of all maximal regions; minimal regions could be obtained by inverting the intensity of I and running the same process. On many images one observes that local binarization is stable over a large range of thresholds in certain regions. Such regions are of interest since they have the following properties:

- Invariance to monotonic transformation. $M : \mathcal{S} \rightarrow \mathcal{S}$ of image intensities.

  The set of extremal regions is unchanged after transformation M, $I(p) < I(q) \rightarrow M\big(I(p)\big) = I'(p) < I'(q) = M\big(I(q)\big)$ since M does not affect adjacency (and thus contiguity). The intensity ordering is preserved.
- Invariance to adjacency preserving (continuous) transformation. $T : \mathcal{D} \rightarrow \mathcal{D}$ on the image domain.
- Stability, since only extremal regions whose support is virtually unchanged over a range of thresholds are selected.
- Multi-scale detection. Since no smoothing is involved, both very fine and very large structure is detected.
- The set of all extremal regions can be enumerated in $\mathcal{O}(n \log \log n)$, i.e., almost in linear time for 8 bit images.

Algorithm 5.6: Enumeration of Extremal Regions.

```
Input: Image $I$.
Output: List of nested extremal regions.
 1. For all pixels sorted by intensity:
      • Place pixel in the image.
      • Update the connected component structure.
      • Update the area for the effected connected component.
 2. For all connected components:
      • Local minima of the rate of change of the connected component area define
        stable thresholds.
```

The computational complexity of step 1 is $\mathcal{O}(n)$ if the image range $\mathcal{S}$ is small, e.g. the typical $[0, 1, \ldots, 255]$, and sorting can be implemented as 'binsort' [Sedgewick, 1998]. As pixels ordered by intensity are placed in the image (either in decreasing or increasing order), the list of connected components and their areas is maintained using the efficient union-find algorithm [Sedgewick, 1998]. The complexity of the algorithm is $\mathcal{O}(n \log \log n)$.

The process produces a data structure holding the area of each connected component as a function of a threshold. A merge of two components is viewed as the end of existence of the smaller component and the insertion of all pixels of the smaller component into the larger one. Finally, intensity levels that are local minima of the rate of change of the area function are selected as thresholds. In the output, each MSER is represented by a local intensity minimum (or maximum) and a threshold.

The structure of Algorithm 5.6 and an efficient watershed algorithm [Vincent and Soille, 1991] (Sections 6.3.4 and 13.7.3) is essentially identical. However, the structure of output of the two algorithms is different. In watershed computation, the focus is on thresholds where regions merge and watershed basins touch. Such thresholds are highly unstable – after a merge, the region area changes abruptly. In MSER detection, a range of thresholds is sought that leaves the watershed basin effectively unchanged.

Detection of MSER is also related to thresholding. Every extremal region is a connected component of a thresholded image. However, no global or 'optimal' threshold is needed, all thresholds are tested and the stability of the connected components evaluated. Finally, the watershed is a partitioning of the input image, where MSER can be nested, if in some parts of the image multiple stable thresholds exist.

In empirical studies [Mikolajczyk et al., 2005; Frauendorfer and Bischof, 2005], the MSER has shown the highest repeatability of affine-invariant detectors in a number of experiments. MSER has been used successfully for challenging wide baseline matching problems [Matas et al., 2004] and in state-of-the-art object recognition systems [Obdrzalek and Matas, 2002; Sivic and Zisserman, 2004].

5.4 Image restoration

Pre-processing methods that aim to suppress degradation using knowledge about its nature are called **image restoration**. Most image restoration methods are based on convolution applied globally to the whole image.