where $\mathbf{x} = [x \quad y \quad \sigma]^T$. Taking the derivative of this function and setting it to zero

$$\frac{\partial I(\mathbf{x})^T}{\partial \mathbf{x}} + \frac{\partial^2 I(\mathbf{x})}{\partial \mathbf{x}^2}(\Delta \mathbf{x}) = 0$$

or

$$\Delta \mathbf{x} = -\left(\frac{\partial^2 I(\mathbf{x})}{\partial \mathbf{x}^2}\right)^{-1} \frac{\partial I(\mathbf{x})}{\partial \mathbf{x}}$$

Once this minimization has stabilized, the value at that point can be computed by plugging the computed value of $\Delta \mathbf{x}$ into Equation (7.43) using only the linear term for simplicity

$$I(\mathbf{x} + \Delta \mathbf{x}) \approx I(\mathbf{x}) + \frac{\partial I(\mathbf{x})^T}{\partial \mathbf{x}}(\Delta \mathbf{x})$$

The popularity of the SIFT feature detector is due to its leveraging of scale space to find features regardless of their scale in the image.

7.5 Feature Descriptors

Once features have been detected in multiple images using one of the techniques described above, the features are often matched across the images. For example, suppose we have an image of a known object from a database, along with a query image that contains the object at some arbitrary position, orientation, and scale. By matching features between the two images, we can infer whether the object is present and, if so, the pose at which the object is located. In order for such an approach to work, it is necessary to compute and match feature **descriptors** that are invariant to changes in pose and illumination.

7.5.1 SIFT Feature Descriptor

Among the many features descriptors that have been proposed, one of the most widely used is the **SIFT feature descriptor**. Although the SIFT feature descriptor typically goes hand-in-hand with the SIFT feature detector, this is not necessary, since the descriptor can be applied anywhere in the image. The algorithm works as follows. Once a feature has been detected using the SIFT feature detector or some other means, the first step is to sample the image gradient magnitudes and orientations in the neighborhood surrounding the feature using the scale at which the feature was detected to specify the size of the neighborhood and the amount of gradient smoothing. The dominant gradient orientation is computed in a manner similar to the one described earlier, and all gradient orientations are rotated relative to this orientation, to make the computation invariant to image rotation. The gradient magnitudes are then weighted by a single Gaussian (whose width is determined by the scale of the detected feature) in order to increase the weight of pixels near the center.

The gradient vectors are quantized into one of several possible orientations and then accumulated over discrete spatial regions into a 3D histogram over space and scale. For example, Figure 7.18 shows the gradients of all the pixels in an 8×8 neighborhood surrounding the feature; there are 8 possible orientations, and $8 \cdot 8 = 64$ gradient vectors which are accumulated into a 4×4 grid. The orientations of the gradient vectors of the 16 pixels in the top-left subarray are accumulated in the histogram bins associated with the top-left cell of the keypoint descriptor array, the values in the top-right subarray are accumulated in the histogram bins associated with the top-right cell of the keypoint descriptor array, and so forth. Each gradient vector votes for the appropriate bin in the histogram with a weight

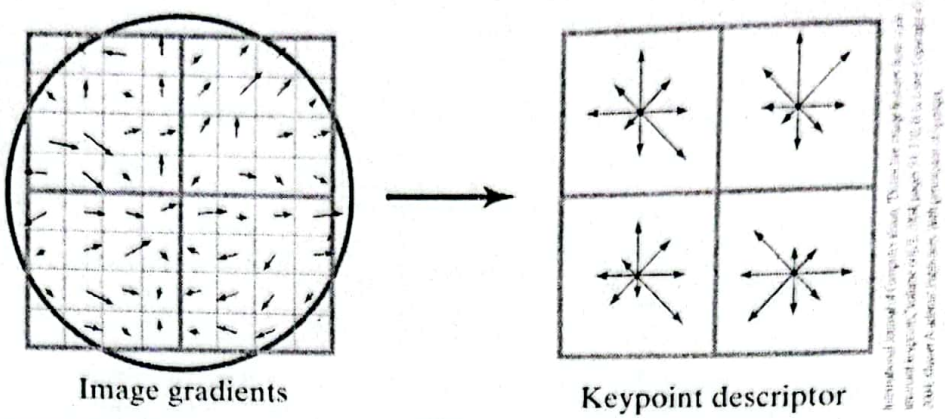


Figure 7.18 The SIFT feature descriptor is computed by accumulating the orientations of the gradient vectors in a neighborhood of the feature point into a 3D array over position and orientation.

proportional to the gradient magnitude, using trilinear interpolation to distribute values to neighboring bins in a robust manner. The values in the 3D histogram are then concatenated to form a vector that describes the feature. Because the figure shows 8 orientations and $2 \cdot 2 = 4$ subarrays, this example yields a 32-dimensional vector, but in practice there are usually 8 orientations and $4 \cdot 4 = 16$ positions, leading to a 128-dimensional vector. To achieve illumination invariance, the vector is normalized to unit length by dividing by its L^2-norm.

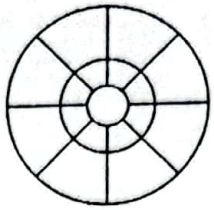
Figure 7.19 shows an application of SIFT feature detections and descriptors. On the left are images of a toy frog and toy train from a database. In the middle is the query image. On the right are the detected feature points that match features in the database, along with the detected objects that were obtained by aggregating the results of the individual features. Notice that the objects are detected despite significant difference in pose, as well as occlusion.

7.5.2 Gradient Location and Orientation Histogram (GLOH)

An extension of the SIFT descriptor is the **gradient location and orientation histogram (GLOH)**. As with SIFT, the gradient of the image is computed, and the gradient orientations are accumulated in a histogram. However, instead of using a rectangular grid of pixels, a log-polar grid is used to specify 17 spatial bins from 2 annuli and 8 orientations, in addition to one bin in the center, as shown in Figure 7.20. With 16 quantized gradient orientations, the histogram contains $17 \cdot 16 = 272$ bins, which are then reduced to a 128-element vector

Figure 7.19 SIFT feature matching results. SIFT feature descriptors from the query image (middle) are matched against descriptors from the database (left) to detect objects at various poses and lighting conditions, and even with severe occlusion (right).

Figure 7.20 The GLOH feature descriptor involves sampling gradient orientations in a log-polar grid.



using PCA[1] applied to a large database of image patches. GLOH feature descriptors have been shown to be slightly more distinctive than SIFT descriptors when matching images with rotation, scale, and viewpoint changes.

7.5.3 Shape Context

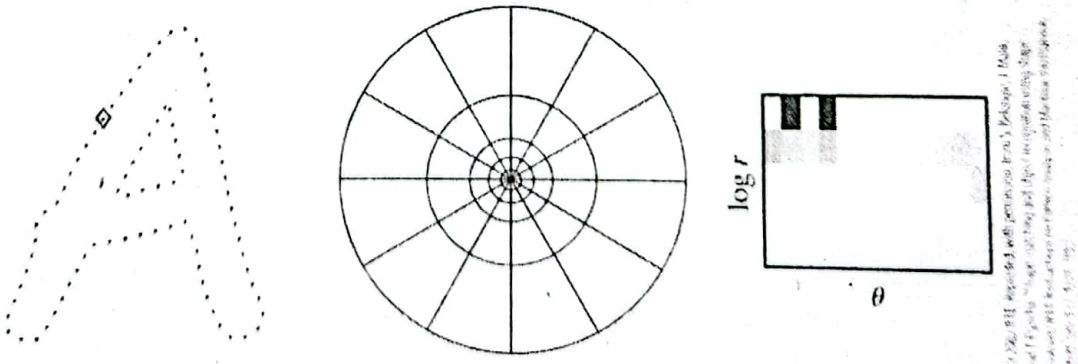
A closely related descriptor designed specifically for binary images is the **shape context**. As shown in Figure 7.21, the shape context of a point on the boundary of an object is computed as a 2D histogram over spatial locations arranged in a log-polar grid similar to that of GLOH, except that the center is also divided into wedges. With 5 radii and 12 angles, the resulting histogram contains 60 bins. Each bin of the histogram contains the sum of edge points within the region defined by the bin. Note that gradient orientation is not used, and all edge points contribute equally to the histogram. The shape context has been used successfully in matching binary shapes.

7.5.4 Histogram of Oriented Gradients (HOG)

Another popular image descriptor is the **histogram of oriented gradients (HOG)**, which is a vector of concatenated histograms of gradient orientations. Since the SIFT feature descriptor is also a vector of concatenated histograms of gradient orientations, it can in some sense be thought of as a HOG. However, the term HOG is usually reserved for a descriptor computed over a dense rectangular region of the image rather than just at a feature point.

The HOG descriptor was developed in the context of pedestrian detection. A 64×128 rectangular window is slid across the image, and at each location the HOG descriptor of the window is computed and then evaluated[2] to determine whether a pedestrian is in the window. The window is divided into a dense array of non-overlapping *cells* consisting of $8 \cdot 8 = 64$ pixels. Within each cell a histogram of gradient orientations is computed by allowing each pixel

Figure 7.21 The shape context captures the shape of a binary region by counting the number of edge pixels in a log-polar grid. From left to right: A binary shape, the log-polar grid, and the resulting histogram at a particular point.



[1] Section 12.3.5 (p. 589).
[2] Using the techniques of Chapter 12 (p. 560).

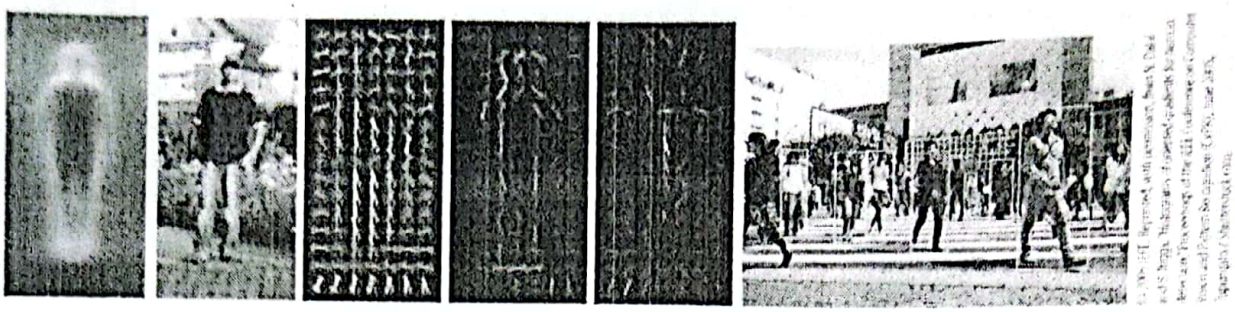


Figure 7.22 Histograms of oriented gradients (HOGs) are widely used for pedestrian detection.

to cast a weighted vote for its orientation, where the weight is given by its gradient magnitude, and the orientations are quantized into either 9 possibilities (if 0° to 180° orientations are used, that is, the sign of the gradient is ignored) or 18 possibilities (if 0° to 360° orientations are used). The cells are grouped into overlapping *blocks*, where each block contains $2 \cdot 2 = 4$ neighboring cells. To provide some amount of illumination invariance, the histograms of the cells within a block are concatenated to form a vector, and the vector is then normalized by dividing by its L^2-norm. Note that, because blocks are overlapping, each cell's histogram is used multiple times to create the vectors for the blocks within which it lies. As with any computer vision algorithm, many variations of this approach are possible by changing the shape of the cells, the manner of normalization, and so forth, but the description presented here is of one of the more common variations. Due to their dense nature, HOG descriptors are able to capture subtle variations in the window, making them enormously successful in the task of detecting pedestrians and other shape-based object classes.

7.6 Further Reading

The Laplacian pyramid is due to Burt and Adelson [1983], where a description of the equal contribution property can also be found. Mallat [1989] is the classic paper that links wavelets, multiresolution analysis, and pyramid algorithms. Scale space is introduced in the paper by Witkin [1983]. A thorough discussion of the Gaussian and Laplacian pyramids and their relationship to scale space can be found in Lindeberg [1994]. An even more in-depth treatment of scale space can be found in Lindeberg [1993]. The causality criterion and the concept of the deep structure of the image are due to Koenderink [1984], where the Hessian matrix is also discussed. Babaud et al. [1986] show the uniqueness of the Gaussian kernel for constructing a scale space. Applications of scale space to feature detection can be found in Canny [1986], Mallat and Zhong [1992], Lindeberg [1998a], and Lindeberg [1998b].

Everyone should read the delightful early paper of Attneave [1954], which connects intensity edges with the notions of predictability and redundancy. The early work on interpreting line drawing images of polyhedral objects is due to Roberts [1963], Huffman [1971], and Clowes [1971]. The Marr-Hildreth edge detector was proposed by Marr and Hildreth [1980], which was replaced by the classic work of Canny [1986]. Canny's paper is a dense read but contains some real gems for anyone patient enough to read it carefully. The sign of the Laplacian of Gaussian is used by Nishihara [1984]. Although space has not permitted a thorough discussion, edge detectors have been compared empirically using Pratt's figure of merit, see Abdou and Pratt [1979], and by the approach of Bowyer and Phillips [1998]. For a more recent approach to edge detection, see the probability of boundary (Pb) detector by Martin et al. [2004].

The Douglas-Peucker line-fitting algorithm is due to Douglas and Peucker [1973], which was slightly preceded by the independent work of Ramer [1972] — hence the name Ramer-Douglas-Peucker. Hershberger and Snoeyink [1992] propose a speedup to Douglas-Peucker with a worst-case running time of $O(n \log n)$, whereas Douglas-Peucker is $O(n^2)$. The algorithm to repeatedly eliminate the smallest area is due to Visvalingam and Whyatt [1992].

The Moravec interest operator is from Moravec [1977]. The classic operators of Beaudet [1978] and

Kitchen and Rosenfeld [1982] are primarily of historic interest only, having been replaced by more recent approaches. The Harris feature detector, which is still widely used, is presented in Harris and Stephens [1988]. The Tomasi-Kanade detector was first described by Tomasi and Kanade [1991], although it is more widely known from the paper by Shi and Tomasi [1994], which explains the alternate name of Shi-Tomasi. Although the second-moment matrix is called the Hessian in Baker and Matthews [2004] in the context of Gauss-Newton minimization for point feature tracking, the term Hessian is usually reserved for the matrix of second-derivatives, as in Bay et al. [2008]. Several widely cited studies have been conducted to compare different feature detectors, such as that of Schmid et al. [2000] and Mikolajczyk and Schmid [2005], which have largely concluded that Harris (or some variation of it) is the most repeatable. Another interesting study is that of Kenney et al. [2005], which concluded that Tomasi-Kanade is the best feature detector according to a particular set of axioms.

The SIFT feature detector and descriptor were introduced by Lowe [2004], the GLOH descriptor is from Mikolajczyk and Schmid [2005], the shape context is due to Belongie et al. [2002], and HOG is presented in Dalal and Triggs [2005]. A number of other feature detectors and/or descriptors have emerged over the years, such as SURF from Bay et al. [2008], FAST from Rosten and Drummond [2006], and DAISY from Tola et al. [2010]. Other work of historical interest is the discovery of receptive fields in the human visual system by Hubel and Wiesel [1962] and Olshausen and Field [1996] and the local jets of Koenderink and van Doorn [1987]. Another relevant piece of work is that of Ozuysal et al. [2007] on fast keypoint recognition.

We did not have space to discuss texture in detail, but the classic work of Julesz [1981] and Julesz and Bergen [1983] on textons should be consulted for historical context. Another classic work on texture is that of Laws [1980]. Steerable filters were introduced by Freeman and Adelson [1991]. A remarkably simple and effective algorithm for texture synthesis can be found in the well-known work of Efros and Leung [1999]. Additional information on visual texture can be found in the overview of Tuceryan and Jain [1993].

PROBLEMS

7.1 Given that the area (i.e., the number of pixels) of the original image is a, and the downsampling factor is $\sqrt[4]{2}$,

(a) Compute the area of the following levels of a Gaussian pyramid: $I^{(1)}, I^{(2)}, I^{(5)}, I^{(6)}$.

(b) Verify that the following relation holds:

$$\frac{\text{area of } I^{(1)}}{\text{area of } I^{(2)}} = \frac{\text{area of } I^{(5)}}{\text{area of } I^{(6)}}$$

7.2 Assuming a downsampling factor of 2, calculate the family of 5-element symmetric kernels that satisfies the equal contribution property. Do any of these kernels look familiar from Pascal's triangle?

7.3 Suppose we wish to construct a Gaussian pyramid with $n = 3$ images per octave.

(a) What is the downsampling factor?

(b) How should σ'^2 be chosen to ensure that the overall smoothing between octaves is $\sigma^2 = 1.2$?

7.4 Suppose we wish to construct a Laplacian pyramid with $n = 5$ images per octave.

(a) What should be the variance ratio ρ in order to ensure that each octave is convolved with the same sequence of variances relative to the image size?

(b) What variance should be applied for pyramid levels 1, 2, and 3 (i.e., what are σ_0^2, σ_1^2, σ_2^2)?

7.5 Explain why the causality criterion is important in computing the scale space.