Figure 5.4: SIFT key points. The arrow starting point, length and the orientation signify respectively the key point position, scale, and dominant orientation. These features are covariant to any image similarity.

5.2.4 Final matching

The outcome is for each image, a few hundreds or thousands SIFT descriptors associated with as many key points. The descriptors of any image can be compared to the descriptors of any other image, or belonging to a database of descriptors built up from many images. The only remaining question is to decide when two descriptors match, or not. In the terms of Lowe again:

> The best candidate match for each keypoint is found by identifying its nearest neighbor in the database of keypoints from training images. The nearest neighbor is defined as the keypoint with minimum Euclidean distance for the invariant descriptor vector. However, many features from an image will not have any correct match in the training database because they arise from background clutter or were not detected in the training images. Therefore, it would be useful to have a way to discard features that do not have any good match to the database. A global threshold on distance to the closest feature does not perform well, as some descriptors are much more discriminative than others. A more effective measure is obtained by comparing the distance of the closest neighbor to that of the second-closest neighbor. (...) This measure performs well because correct matches need to have the closest neighbor significantly closer than the closest incorrect match to achieve reliable matching. For false matches, there will likely be a number of other false matches within similar distances due to the high dimensionality of the feature space. We can think of the second-closest match as providing an estimate of the density of false matches within this portion of the feature space

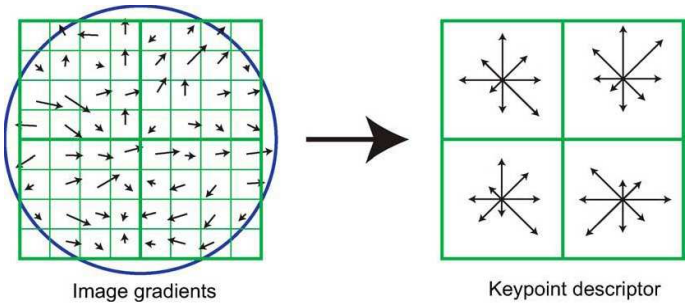

Figure 5.5: Example of a 2×2 descriptor array of orientation histograms (right) computed from an 8×8 set of samples (left). The orientation histograms are quantized into 8 directions and the length of each arrow corresponds to the magnitude of the histogram entry. (From [166])

> and at the same time identifying specific instances of feature ambiguity. (...) For our object recognition implementation, we reject all matches in which the distance ratio is greater than 0.8, which eliminates 90% of the false matches while discarding less than 5% of the correct matches.

5.3 Image acquisition model underlying SIFT

5.3.1 The camera model

We always work on the camera CCD plane, whose mesh unit is taken to be 1. We shall always assume that the camera pixels are indexed by $\boldsymbol{Z}^2$. The image sampling operator is therefore always $\mathbf{S}_1$. Our second assumption is that the digital initial image is well-sampled and obtained by a gaussian kernel. Thus, the digital image is $\mathbf{u} = \mathbf{S}_1 G_\delta A u_0$, where $\delta \geq \mathbf{c}$, $\mathbf{c} \simeq 0.6$ ensures well-sampling (see Chapter 4.2), and A is a similarity with positive determinant. (In fact Lowe's original paper assumes $\mathbf{c} \simeq 0.5$, which amounts to assume a slight under-sampling of the original image).

Definition 5.1. *We model all digital frontal images obtained from a given ideal planar object whose frontal infinite resolution image is u_0 as*

$$\mathbf{u}_0 =: \mathbf{S}_1 G_\delta A u_0 \tag{5.3}$$

where $\delta \geq \mathbf{c}$ and A is a $A = RH_\lambda \mathcal{T}$ is the composition of a translation and of a similarity.

So the possibility of aliasing (under-sampling, $\delta < \mathbf{c}$ is discarded). Taking into account the way the digital image is blurred and sampled in the SIFT method, we can now list the SIFT assumptions and formalize the method itself. The description is by far simpler if we do it without mixing in sampling issues. We need not mix them in, since the fact that images are well-sampled at all stages permits equivalently to describe all operations with the continuous images

directly, and to deduce afterwards the discrete operators on samples. We refer to section 4.2.1 for this passage from continuous to discrete operations in the well-sampled world.

5.3.2 Condensed description of the SIFT method

1. There is an underlying infinite resolution bounded planar image u_0;
2. The initial digital image is $\mathbf{S}_1 G_\delta A u_0$ where $\delta \geq \mathbf{c}$, and $A = RH_\lambda \mathcal{T}$ is the composition of a rotation, a zoom, and a translation;
3. the SIFT method computes a sufficient scale-space sampling of $u(\sigma, \mathbf{x}) = (G_\sigma G_\delta A u_0)(\mathbf{x})$, and deduces by the Newton method the accurate location or key points defined as extrema in scale-space of the spatial image Laplacian, $\Delta u(\sigma; \mathbf{x})$;
4. The blurred $u(\sigma, \cdot)$ image is then re-sampled around each characteristic point with sampling mesh $\sqrt{\sigma^2 + \mathbf{c}^2}$;
5. the directions of the sampling axes are fixed by a dominant direction of the gradient of $u(\sigma, \cdot)$ in a neighborhood, with size proportional to $\sqrt{\sigma^2 + \mathbf{c}^2}$ around the characteristic point;
6. the rest of the operations in the SIFT method is a contrast invariant encoding of the samples around each characteristic point. It is not needed for the discussion to follow.

5.4 Scale and SIFT: consistency of the method

In this section, in conformity with the SIFT model of Sect. 5.3.2, the digital image is a frontal view of an infinite resolution ideal image u_0. In that case, $A = H\mathcal{T}R$ is the composition of a homothety H, a translation $\mathcal{T}$ and a rotation R. Thus the digital image is $\mathbf{u} = \mathbf{S}_1 G_\delta H \mathcal{T} R u_0$, for some H, $\mathcal{T}$, R as above. Assuming that the image is not aliased boils down, by the experimental results of Sect. 4.2, to assuming $\delta \geq 0.8$.

Consider $\mathcal{T}$ an arbitrary image translation, R an arbitrary image rotation, H_λ an arbitrary image homothety, G an arbitrary gaussian convolution, D the gradient and Δ the Laplacian, all applied to continuous images. We say that there is strong commutation of two of these operators if we can exchange the order of their application to any image. We say that there is weak commutation between two of these operators if we can exchange their order by changing one of the parameters of one of the operators. For example we have $R\mathcal{T} = \mathcal{T}'R$, meaning that given R and $\mathcal{T}$ there is $\mathcal{T}'$ such that the former relation occurs. The next lemma is straightforward.

Lemma 5.2. *All of the aforementioned operators weakly commute, with the following exceptions: R and G commute strongly, $DH_\lambda = \lambda H_\lambda D$, $\Delta H_\lambda = \lambda^2 H_\lambda \Delta$, and D and Δ do not commute.*

Exercise 5.4. Check all of the mentioned commutations and give their exact formula. There are six kinds of operators: translations, rotations, homotheties, gaussian convolutions, gradients, and Laplacians. Thus, there are 15 verifications to make. ∎

Lemma 5.3. *For any rotation R and any translation $\mathcal{T}$, the SIFT descriptors of $\mathbf{S}_1 G_\delta H \mathcal{T} R u_0$ are identical to those of $\mathbf{S}_1 G_\delta H u_0$.*

Proof. By the weak commutation of translations and rotations with all other operators (Lemma 5.2), the SIFT descriptors of a rotated or translated image are identical to those of the original. Indeed, the set of scale space Laplacian extrema is covariant to space translations and rotations. The normalization process for each SIFT descriptor situates the origin at each extremum in turn, thus canceling the translation. The local sampling grid defining the SIFT patch has axes given by peaks in its gradient direction histogram. Such peaks are translation invariant and rotation covariant. Thus, the normalization of the direction also cancels the rotation. □

Lemma 5.4. *Let $\mathbf{u}$ and $\mathbf{v}$ be two digital images that are frontal snapshots of the same continuous flat image u_0, $\mathbf{u} = \mathbf{S}_1 G_\beta H_\lambda u_0$ and $\mathbf{v} =: \mathbf{S}_1 G_\delta H_\mu u_0$, taken at different distances, with different gaussian blurs and possibly different sampling rates. Let $w(\sigma, \mathbf{x}) = (G_\sigma u_0)(\mathbf{x})$ denote the scale space of u_0. Then the scale spaces of $u = G_\beta H_\lambda u_0$ and $v = G_\delta H_\mu u_0$ are*

$$u(\sigma, \mathbf{x}) = w(\lambda\sqrt{\sigma^2 + \beta^2}, \lambda\mathbf{x}) \quad \textit{and} \quad v(\sigma, \mathbf{x}) = w(\mu\sqrt{\sigma^2 + \delta^2}, \mu\mathbf{x}).$$

If $(s_0, \mathbf{x}_0)$ is a key point of w satisfying $s_0 \geq \max(\lambda\beta, \mu\delta)$, then it corresponds to a key point of u at the scale σ_1 such that $\lambda\sqrt{\sigma_1^2 + \beta^2} = s_0$, whose SIFT descriptor is sampled with mesh $\sqrt{\sigma_1 + \mathbf{c}^2}$. In the same way $(s_0, \mathbf{x}_0)$ corresponds to a key point of v at scale σ_2 such that $s_0 = \mu\sqrt{\sigma_2^2 + \delta^2}$, whose SIFT descriptor is sampled with mesh $\sqrt{\sigma_2^2 + \mathbf{c}^2}$.

Proof. The interpolated initial images are by (4.2)

$$u =: I\mathbf{S}_1 G_\beta H_\lambda u_0 = G_\beta H_\lambda u_0 \text{ and } v =: I\mathbf{S}_1 G_\delta H_\mu u_0 = G_\delta H_\mu u_0.$$

Computing the scale-space of these images amounts to convolve these images for every $\sigma > 0$ with G_σ, which yields, using the commutation relation (4.7) and the semigroup property (4.6):

$$u(\sigma, \cdot) = G_\sigma G_\beta H_\lambda u_0 = G_{\sqrt{\sigma^2+\beta^2}} H_\lambda u_0 = H_\lambda G_{\lambda\sqrt{\sigma^2+\beta^2}} u_0.$$

By the same calculation, this function is compared by SIFT with

$$v(\sigma, \cdot) = H_\mu G_{\mu\sqrt{\sigma^2+\delta^2}} u_0.$$

Set $w(s, \mathbf{x}) =: G_s u_0$. Then the scale spaces compared by SIFT are

$$u(\sigma, \mathbf{x}) = w(\lambda\sqrt{\sigma^2 + \beta^2}, \lambda\mathbf{x}) \text{ and } v(\sigma, \mathbf{x}) = w(\mu\sqrt{\sigma^2 + \delta^2}, \mu\mathbf{x}).$$

Let us consider an extremal point $(s_0, \mathbf{x}_0)$ of the Laplacian of the scale space function w. If $s_0 \geq \max(\lambda\beta, \mu\delta)$, an extremal point occurs at scales σ_1 for the Laplacian of $u(\sigma, \mathbf{x})$ and at scale σ_2 for the Laplacian of $v(\sigma, \mathbf{x})$ satisfying

$$s_0 = \lambda\sqrt{\sigma_1^2 + \beta^2} = \mu\sqrt{\sigma_2^2 + \delta^2}. \tag{5.4}$$

□

Theorem 5.5. *Let* $\mathbf{u}$ *and* $\mathbf{v}$ *be two digital images that are frontal snapshots of the same continuous flat image* u_0, $\mathbf{u} = \mathbf{S}_1 G_\beta H_\lambda \mathcal{T} R u_0$ *and* $\mathbf{v} =: \mathbf{S}_1 G_\delta H_\mu u_0$, *taken from arbitrary distances, with possibly different camera gaussian blurs, with an arbitrary camera translation parallel to its focal plane, and an arbitrary rotation around its optical axe. Without loss of generality, assume* $\lambda \leq \mu$. *Then if the camera blurs are standard* ($\beta = \delta = \mathbf{c}$), *each SIFT descriptor of* $u = I\mathbf{u}$ *is identical to some SIFT descriptor of* $v = I\mathbf{v}$. *If* $\beta \neq \delta$ *(or* $\beta = \delta \neq \mathbf{c}$), *the SIFT descriptors of* u *become (quickly) similar to SIFT descriptors of* v *when their scales grow, namely as soon as* $\frac{\sigma_1}{\max(\mathbf{c},\beta)} \gg 1$ *and* $\frac{\sigma_2}{\max(\mathbf{c},\delta)} \gg 1$.

Proof. By the result of Lemma 5.3, we can neglect the effect of translations and rotations. Therefore assume w.l.o.g. that the images under comparison are as in Lemma 5.4. Assume a key point $(s_0, \mathbf{x}_0)$ of w has scale $s_0 \geq \max(\lambda\beta, \mu\delta)$. This key point has a sampling rate proportional to s_0. There is a corresponding key point $(\sigma_1, \frac{\mathbf{x}_0}{\lambda})$ for u with sampling rate $\sqrt{\sigma_2^2 + \mathbf{c}^2}$ and a corresponding key point $(\sigma_2, \frac{\mathbf{x}_0}{\mu})$ with sampling rate $\sqrt{\sigma_2^2 + \mathbf{c}^2}$ for $\mathbf{v}$. To have a common reference for these sampling rates, it is convenient to refer to the corresponding sampling rates for $w(s_0, \mathbf{x})$, which are $\lambda\sqrt{\sigma_1^2 + \mathbf{c}^2}$ for the SIFT descriptors of u at scale σ_1, and $\mu\sqrt{\sigma_2^2 + \mathbf{c}^2}$ for the descriptors of $\mathbf{v}$ at scale σ_2. Thus the SIFT descriptors of u and v for $\mathbf{x}_0$ will be identical if and only if $\lambda\sqrt{\sigma_1^2 + \mathbf{c}^2} = \mu\sqrt{\sigma_2^2 + \mathbf{c}^2}$. Now, we have $\lambda\sqrt{\sigma_1^2 + \beta^2} = \mu\sqrt{\sigma_2^2 + \delta^2}$, which implies $\lambda\sqrt{\sigma_1^2 + \mathbf{c}^2} = \mu\sqrt{\sigma_2^2 + \mathbf{c}^2}$ if and only if

$$\lambda^2\beta^2 - \mu^2\delta^2 = (\lambda^2 - \mu^2)\mathbf{c}^2. \tag{5.5}$$

Since λ and μ are proportional to camera distances to the observed object u_0, they are arbitrary and generally different. Thus, the only way to ensure (5.5) is to have $\beta = \delta = \mathbf{c}$, which means that the blurs of both images (or of both cameras) are ideal and gaussian. In any case, $\beta = \delta = \mathbf{c}$ does imply that the SIFT descriptors of both images are identical.

The second statement is straightforward: If σ_1 and σ_2 are large enough with respect to β, δ and $\mathbf{c}$, the relation $\lambda\sqrt{\sigma_1^2 + \beta^2} = \mu\sqrt{\sigma_2^2 + \delta^2}$, implies $\lambda\sqrt{\sigma_1^2 + \mathbf{c}^2} \simeq \mu\sqrt{\sigma_2^2 + \mathbf{c}^2}$. □

The almost perfect scale invariance of SIFT stated in Theorem 5.5 is illustrated by the striking example of Fig. 5.6. The 28 SIFT key points of a very small digital image $\mathbf{u}$ are compared to the 86 key points obtained by zooming in $\mathbf{u}$ by a 32 factor: The resulting digital image is the digital image $\mathbf{v} = \mathbf{S}_{\frac{1}{32}} I\mathbf{u}$, again obtained by zero-padding. For better observability, both images are displayed with the same size by enlarging the pixels of $\mathbf{u}$. Almost each key point (22 out of 28) of $\mathbf{u}$ finds its counterpart in $\mathbf{v}$. 22 matches are detected between the descriptors as shown on the right. If we trust Theorem 5.5, all descriptors of $\mathbf{u}$ should have been retrieved in $\mathbf{v}$. This does not fully happen for two reasons. First, the SIFT method thresholds (not taken into account in the theorem) eliminate many potential key points. Second, the zero-padding interpolation giving $\mathbf{v}$ is imperfect near the image boundaries.

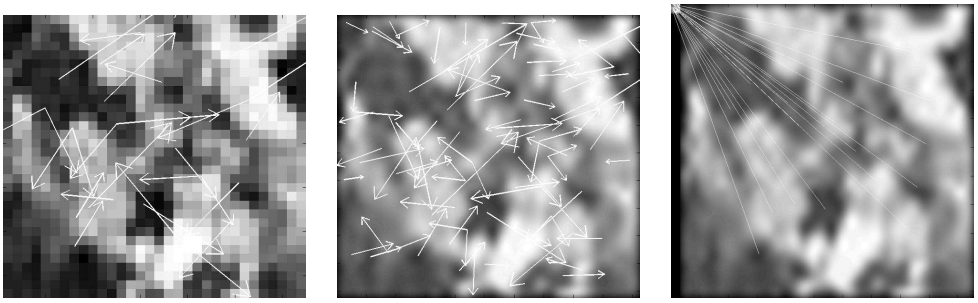

Figure 5.6: Scale invariance of SIFT, an illustration of Theorem 5.5. Left: a very small digital image $\mathbf{u}$ with its 28 key points. For the conventions to represent key points and matches, see the comments in Fig. 5.4. Middle: this image is over sampled by a 32 factor to $\mathbf{v} = \mathbf{S}_{\frac{1}{32}} I\mathbf{u}$. It has 86 key points. Right: 22 matches found between $\mathbf{u}$ and $\mathbf{S}_{\frac{1}{32}} I\mathbf{u}$.

5.5 Exercises

Exercise 5.5. The aim of the exercise is to explain why the experiment of Fig. 5.6 works, and to illustrate Theorem 5.5. The digital zoom in by a factor λ is nothing but the discrete over-sampling operator $\mathcal{S}_{\frac{1}{\lambda}}$ with sampling step $\frac{1}{\lambda}$, defined in (4.11). Here, $\lambda = 32$. In the experiment an original digital image $\mathbf{u} = \mathbf{S}_1 G_\delta u$ is zoomed into $\mathbf{v} = \mathcal{S}_{\frac{1}{\lambda}} \mathbf{u}$.

1) Using the definition of the discrete zoom and the right commutation relations given in this chapter and in the former one (give their numbers), show that

$$\mathbf{v} = \mathbf{S}_{\frac{1}{\lambda}} G_\delta I\mathbf{u} = \mathbf{S}_1 G_{\lambda\delta} H_{\frac{1}{\lambda}} I\mathbf{u}.$$

2) Is $\mathbf{v}$ well-sampled if $\mathbf{u}$ was?

3) By applying carefully Theorem 5.5, assuming that $\delta \simeq \mathbf{c}$, discuss why SIFT manages to match SIFT descriptors of $\mathbf{u}$ and $\mathbf{v}$. ■

5.6 Comments and references

Many variations exist on the computation of interest points, following the pioneering work of Harris and Stephens [115]. The Harris-Laplace and Hessian-Laplace region detectors [186, 189] are invariant to rotation and scale changes. Some moment-based region detectors [160, 29] including Harris-Affine and Hessian-Affine region detectors [187, 189], an edge-based region detector [256], an intensity-based region detector [256], an entropy-based region detector [137], and two independently developed level line-based region detectors MSER ("maximally stable extremal region") [182] and LLD ("level line descriptor") [198, 200, 202] are designed to be invariant to affine transformations. These two methods stem from the Monasse image registration method [193] that used well contrasted extremal regions to register images. MSER is the most efficient one and has shown better performance than other affine invariant detectors [191]. However, as pointed out in [166], no known detector is actually fully affine invariant: All

of them start with initial feature scales and locations selected in a non-affine invariant manner. The difficulty comes from the scale change from an image to another: This change of scale is actually an under-sampling, which means that the images differ by a blur.

In his milestone paper [166], Lowe has addressed this central problem and has proposed the so called scale-invariant feature transform (SIFT) descriptor, that is invariant to image translations and rotations, to scale changes (blur), and robust to illumination changes. It is also surprisingly robust to large enough orientation changes of the viewpoint (up to 60 degrees). Based on the scale-space theory [159], the SIFT procedure simulates all gaussian blurs and normalizes local patches around scale covariant image key points that are Laplacian extrema. A number of SIFT variants and extensions, including PCA-SIFT [139] and gradient location-orientation histogram (GLOH) [190], that claim to have better robustness and distinctiveness with scaled-down complexity have been developed ever since [93, 157]. Demonstrated to be superior to other descriptors [188, 190], SIFT has been popularly applied for scene recognition [86, 196, 230, 261, 105, 240] and detection [94, 206], robot localization [31, 208, 133], image registration [275], image retrieval [114], motion tracking [257, 142], 3D modeling and reconstruction [224, 262], building panoramas [3, 39], or photo management [274, 155, 55].

As pointed out by several benchmarks, the robustness and repeatability of the SIFT descriptors outperforms other methods. However, such benchmarks mix three very different criteria that, in our opinion, should have been discussed separately. The first one is the formal real invariance of each method when all thresholds have been eliminated. This real invariance has been proved here for SIFT. The second criterion is the practical validity of the sampling method used in SIFT, that has been again checked in Chapter 4.2. The last criterion is the clever fixing of several thresholds in the SIFT method ensuring robustness, repeatability, and a low false alarm rate. This one has been extensively tested and confirmed in previous benchmark papers (see also the very recent and complete report [87]). We think, however, that the success of SIFT in these benchmarks is primarily due to its full scale invariance.

Chapter 6

Linear Scale Space and Edge Detection

The general analysis framework in which an image is associated with smoothed versions of itself at several scales is called *scale space*. Following the results of Chapter 3, a linear scale space must be performed by applying the heat equation to the image. The main aim of this smoothing is to find out *edges* in the image. We shall first explain this doctrine. In the second section, we discuss experiments and several serious objections to such an image representation.

6.1 The edge detection doctrine

One of the uses of linear theory in two dimensions is *edge detection*. The assumption of the edge detection doctrine is that relevant information is contained in the traces produced in an image by the apparent contours of physical objects. If a black object is photographed against a white background, then one expects the silhouette of the object in the image to be bounded by a closed curve across which the light intensity u_0 varies strongly. We call this curve an *edge*. At first glance, it would seem that this edge could be detected by computing the gradient Du_0, since at a point $\mathbf{x}$ on the edge, $|Du_0(\mathbf{x})|$ should be large and $Du(\mathbf{x})$ should point in a direction normal to the boundary of the silhouette. It would therefore appear that finding edges amounts to computing the gradient of u_0 and determining the points where the gradient is large. This conclusion is unrealistic for two reasons:

(a) There may be many points where the gradient is large due to small oscillations in the image that are not related to real objects. Recall that digital images are always noisy, and thus there is no reason to assume the existence or computability of a gradient.

(b) The points where the gradient exceeds a given threshold are likely to form regions and not curves.

As we emphasized in the Introduction, objection (a) is dealt with by smoothing the image. We associate with the image u_0 smoothed versions $u(t, \cdot)$, where the scale parameter t indicates the amount of smoothing. In the classical linear theory, this smoothing is done by convolving u_0 with the Gaussian G_t.

One way that objection (b) has been approached is by redefining edge points. Instead of just saying an edge point is a point $\mathbf{x}$ where $|Du_0(\mathbf{x})|$ exceeds a threshold, one requires the gradient to satisfy a maximal property. We illustrate this in one dimension. Suppose that $u \in C^2(\mathbb{R})$ and consider the points where $|u'(x)|$ attains a local maximum. At some of these points, the second derivative u'' changes sign, that is, $\operatorname{sign}(u''(x-h)) \neq \operatorname{sign}(u''(x+h))$ for sufficiently small h. These are the points where u'' *crosses zero*, and they are taken to be the edge points. Note that this criterion avoids classifying a point x as an edge point if the gradient is constant in an interval around x. Marr and Hildreth generalized this idea to two dimensions by replacing u'' with the Laplacian Δu, which is the only isotropic linear differential operator of order two that generalizes u'' [179]. Haralick's edge detector is different but in the same spirit [111]. Haralick gives up linearity and defines edge points as those points where the gradient has a local maximum in the direction of the gradient. In other words, an edge point $\mathbf{x}$ satisfies $g'(0) = 0$, where $g(t) = |Du(\mathbf{x}+tDu(\mathbf{x})|/|Du(\mathbf{x})|$. This implies that $D^2u(\mathbf{x})(Du(\mathbf{x}), Du(\mathbf{x})) = 0$ (see Exercise 6.2). We are now going to state these two algorithms formally. They are illustrated in Figures 6.2 and 6.3, respectively.

Algorithm 6.1 (Edge detection: Marr–Hildreth zero-crossings).

(1) Create the multiscale images $u(t, \cdot) = G_t * u_0$ for increasing values of t.

(2) At each scale t, compute all the points where $Du \neq 0$ and Δu changes sign. These points are called zero-crossings of the Laplacian, or simply zero-crossings.

(3) (Optional) Eliminate the zero-crossings where the gradient is below some prefixed threshold.

(4) track back from large scales to fine scales the "main edges" detected at large scales.

Algorithm 6.2 (Edge detection: The Haralick–Canny edge detector).

(1) As before, create the multiscale images $u(t, \cdot) = G_t * u_0$ for increasing values of t.

(2) At each scale t, find all points $\mathbf{x}$ where $Du(\mathbf{x}) \neq 0$ and $D^2u(\mathbf{x})(\mathbf{z}, \mathbf{z})$ crosses zero, $\mathbf{z} = Du/|Du|$. At such points, the function $s \mapsto u(\mathbf{x} + s\mathbf{z})$ changes from concave to convex, or conversely, as s passes through zero.

(3) At each scale t, fix a threshold $\theta(t)$ and retain as *edge points at scale t* only those points found above that satisfy $|Du(\mathbf{x})| > \theta(t)$. The backtracking step across scales is the same as for Marr–Hildreth.

In practice, edges are computed for a finite number of dyadic scales, $t = 2^n$, $n \in \mathbb{Z}$.

6.1.1 Discussion and critique

The Haralick–Canny edge detector is generally preferred for its accuracy to the Marr–Hildreth algorithm. Their use and characteristics are, however, essentially

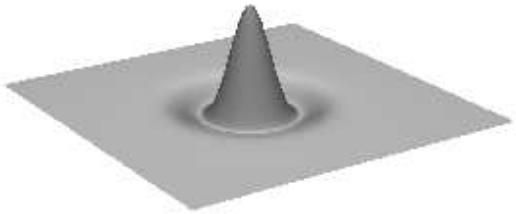

Figure 6.1: A three-dimensional representation of the Laplacian of the Gaussian. This convolution kernel, which is a wavelet, is used to estimate the Laplacian of an image at different scales of linear smoothing.

the same. There are also many variations—attempted improvements—of the algorithms we have described, and the following discussion adapts easily to these related edge detection schemes. The first thing to notice is that, by Proposition 2.5, $u(t,\cdot) = G_t * u_0$ is a C^∞ function for each $t > 0$ if $u_0 \in \mathcal{F}$. Thus we can indeed compute second order differential operators applied to $u(t,\cdot) = G_t * u_0$, $t > 0$. In the case of linear operators like the Laplacian or the gradient, the task is facilitated by the formula proved in the mentioned proposition. For example, we have $\Delta u(t,\mathbf{x}) = \Delta(G_t * u_0)(\mathbf{x}) = (\Delta G_t) * u_0(\mathbf{x})$, where in dimension two (Figure 6.1),

$$\Delta G_t(\mathbf{x}) = \frac{|\mathbf{x}|^2 - 4t}{16\pi t^3} \mathrm{e}^{-|\mathbf{x}|^2/4t}.$$

In the same way, Haralick's edge detector makes sense, because u is C^∞, at all points where $Du(\mathbf{x}) \neq 0$. If $Du(\mathbf{x}) = 0$, then $\mathbf{x}$ cannot be an edge point, since u is "flat" there. Thus, thanks to the filtering, there is no theoretical problem with computing edge points. There are, however, practical objections to these methods, which we will now discuss.

Linear scale space

The first serious problems are associated with the addition of an extra dimension: Having many images $u(t,\cdot)$ at different scales t confounds our understanding of the image and adds to the cost of computation. We no longer have an absolute definition of an edge. We can only speak of edges at a certain scale. Conceivably, a way around this problem would be to track edges across scales. In fact, it has been observed in experiments that the "main edges" persist under convolution as t increases, but they lose much of their spatial accuracy. On the other hand, filtering with a sharp low-pass filter, that is, with t small, keeps these edges in their proper positions, but eventually, as t becomes very small, even these main edges can be lost in the crowd of spurious edge signals due to noise and texture. The scale space theory of Witkin proposes to identify the main edges at some scale t and then to track them backward as t decreases [273]. In theory, it would seem that this method could give an accurate location of the main edges. In practice, any implementation of these ideas is computationally costly due to the problems involved with multiple thresholdings and following edges across scales. In fact, tracking edges across scales is incompatible with having thresholds for the gradients, since such thresholds may remove edges at

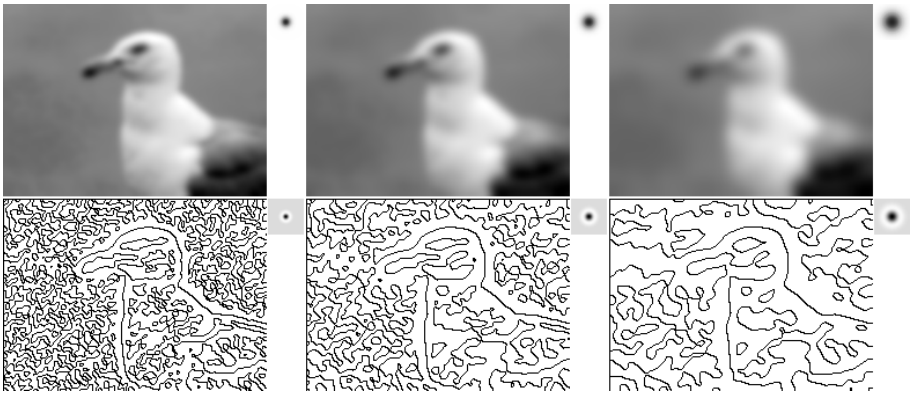

Figure 6.2: Zero-crossings of the Laplacian at different scales. This figure illustrates the original scale space theory as developed by David Marr [177]. To extract more global structure, the image is convolved with Gaussians whose variances are powers of two. One computes the Laplacian of the smoothed image and displays the lines along which this Laplacian changes sign: the zero-crossings of the Laplacian. According to Marr, these zero-crossings represent the "raw primal sketch" of the image, or the essential information on which further vision algorithms should be based. Above, left to right: the results of smoothing and the associated Gaussian kernels at scales 1, 2, and 4. Below, left to right: the zero-crossings of the Laplacian and the corresponding kernels, which are the Laplacians of the Gaussians used above.

certain scales and not at others. The conclusion is that one should trace all zero-crossings across scales without considering whether they are true edges or not. This makes matching edges across scales very difficult. For example, experiments show that zero-crossings of sharp edges that are sparse at small scales are no longer sparse at large scales. (Figure 6.4 shows how zero-crossings can be created by linear smoothing.) The Haralick–Canny detector suffers from the same problems, as is well demonstrated by experiments.

Other problems with linear scale space are illustrated in Figures 6.5 and 6.6. Figure 6.5 illustrates how linear smoothing can create new gray levels and new extrema. Figure 6.6 shows that linear scale space does not maintain the inclusion between objects. The shape inclusion principal will be discussed in Chapter 21.

We must conclude that the work on linear edge detection has been an attempt to build a theory that has not succeeded. After more than thirty years of activity, it has become clear that no robust technology can be based on these ideas. Since edge detection algorithms depend on multiple thresholds on the gradient, followed by "filling-the-holes" algorithms, there can be no scientific agreement on the identification of edge points in a given image. In short, the problems associated with linear smoothing followed by edge detection have not been resolved by the idea of chasing edges across scales.

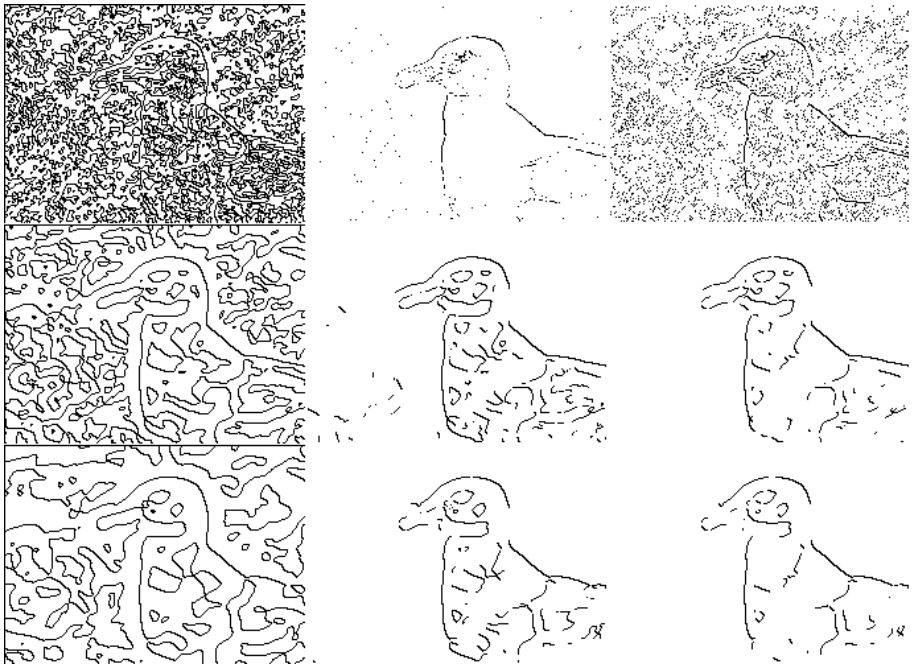

Figure 6.3: Canny's edge detector. These images illustrate the Canny edge detector. Left column: result of the Canny filter without the threshold on the gradient. Middle column: result with a visually "optimal" scale and an image-dependent threshold (from top to bottom: 15, 0.5, 0.6). Right column: result with a fixed gradient threshold equal to 0.7. Note that such an edge detection theory depends on no fewer than two parameters that must be fixed by the user: smoothing scale and gradient threshold .

Figure 6.4: Zero-crossings of the Laplacian of a synthetic image. Left to right: the original image; the image linearly smoothed by convolution with a Gaussian; the sign of the Laplacian of the filtered image (the gray color corresponds to values close to 0, black to clear-cut negative values, white to clear-cut positive values); the zero-crossings of the Laplacian. This experiment clearly shows a drawback of the Laplacian as edge detector.

Contrast invariance

The use of contrast-invariant operators can solve some of the technical problems associated with linear smoothing and other linear image operators. An (image) operator $u \mapsto Tu$ is contrast invariant if T commutes with all nondecreasing

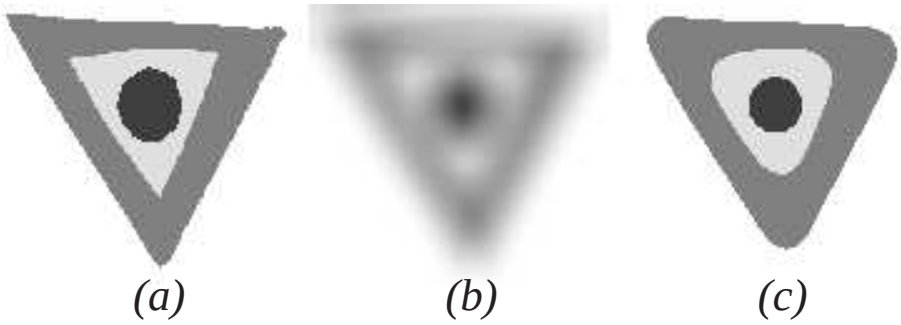

Figure 6.5: The heat equation creates structure. This experiment shows that linear scale space can create new structures and thus increase the complexity of an image. Left to right: The original synthetic image (a) contains three gray levels. The black disk is a regional and absolute minimum. The "white" ring around the black disk is a regional and absolute maximum. The outer gray ring has a gray value between the other two and is a regional minimum. The second image (b) shows what happens when (a) is smoothed with the heat equation: New local extrema have appeared. Image (c) illustrates the action on (a) of a contrast-invariant local filter, the iterated median filter, which is introduced in Chapter 13.

functions g, that is, if

$$g(Tu) = T(g(u)). \tag{6.1}$$

If image analysis is to be robust, it must be invariant under changes in lighting that produce contrast changes. It must also be invariant under the nonlinear response of the sensors used to capture an image. These, and perhaps other, contrast changes are modeled by g. If g is strictly increasing, then relation (6.1) ensures that the filtered image $Tu = g^{-1}(T(g(u)))$ does not depend on g. A problem with linear theory is that linear smoothing, that is, convolution, is not generally contrast invariant:

$$g(k * u) \neq k * (g(u)).$$

In the same way, the operator T_t that maps u_0 into the solution of the heat equation, $u(t, \cdot)$ is not generally contrast invariant. In fact, if g is C^2, then

$$\frac{\partial(g(u))}{\partial t} = g'(u)\frac{\partial u}{\partial t}$$

and

$$\Delta(g(u)) = g'(u)\Delta u + g''(u)|Du|^2.$$

Exercise 6.1. Prove this last relation. Prove that if $g(s) = as + b$ then $g(u)$ satisfies the heat equation if u does. ∎

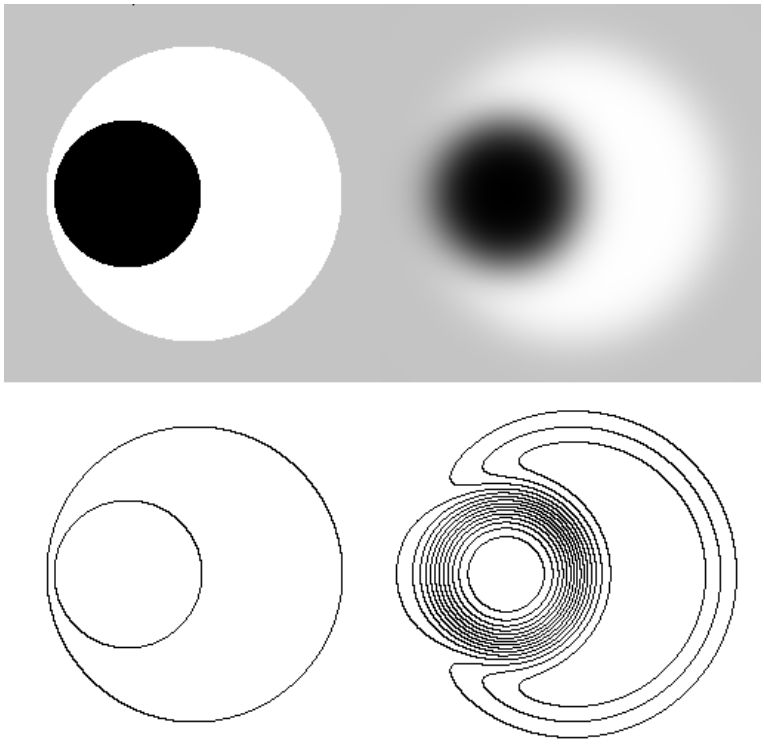

Figure 6.6: Violation of the inclusion by the linear scale space. Top, left: an image that contains a black disk enclosed by a white disk. Top, right: At a certain scale, the black and white circles mix together. Bottom, left: The boundaries of the two circles. Bottom, right: After smoothing with a certain value of t, the inclusion that existed for very small t in no longer preserved. We display the level lines of the image at levels multiples of 16.

6.2 Exercises

Exercise 6.2. Define an edge point $\mathbf{x}$ in a smooth image u as a point $\mathbf{x}$ at which $g(t)$ attains a maximum, where

$$g(t) = |Du\left(\mathbf{x} + t\frac{Du(\mathbf{x})}{|Du(\mathbf{x})|}\right)|.$$

Prove by differentiating $g(t)$ that edge points satisfy $D^2u(\mathbf{x})(Du(\mathbf{x}), Du(\mathbf{x})) = 0$ ■

Exercise 6.3. Construct simple functions u, g, and k such that $g(k * u) \neq k * (g(u))$. ■

Exercise 6.4. Consider the Perona–Malik equation in divergence form:

$$\frac{\partial u}{\partial t} = \mathrm{div}(g(|Du|)Du), \tag{6.2}$$

where $g(s) = 1/(1 + \lambda^2 s^2)$. It is easily checked that we have a diffusion equation when $\lambda|Du| \leq 1$ and an inverse diffusion equation when $\lambda|Du| > 1$. To see this, consider the second derivative of u in the direction of Du,

$$u_{\xi\xi} = D^2u\left(\frac{Du}{|Du|}, \frac{Du}{|Du|}\right),$$

and the second derivative of u in the orthogonal direction,

$$u_{\eta\eta} = D^2 u\left(\frac{Du^\perp}{|Du|}, \frac{Du^\perp}{|Du|}\right),$$

where $Du = (u_x, u_y)$ and $Du^\perp = (-u_y, u_x)$. The Laplacian can be rewritten in the intrinsic coordinates (ξ, η) as $\Delta u = u_{\xi\xi} + u_{\eta\eta}$. Prove that the Perona–Malik equation then becomes

$$\frac{\partial u}{\partial t} = \frac{1}{1 + \lambda^2 |Du|^2} u_{\eta\eta} + \frac{1 - \lambda^2 |Du|^2}{(1 + \lambda^2 |Du|^2)^2} u_{\xi\xi}.$$

Interpret the local behavior of the equation as a heat equation or a reverse heat equation according to the size of $|Du|$ compared to λ^{-1}. ■

6.3 Comments and references

Scale space. The term "scale space" was introduced by Witkin in 1983. He suggested tracking the zero-crossings of the Laplacian of the smoothed image across scales [273]. Yuille and Poggio proved that these zero-crossings can be tracked for one-dimensional signals [278]. Hummel and Moniot [124, 127] and Yuille and Poggio [279] analyzed the conjectures of Marr and Witkin according to which an image is completely recoverable from its zero-crossings at different scales. Mallat formulated Marr's conjecture as an algorithm in the context of wavelet analysis. He replaced the Gaussian with a two-dimensional cubic spline, and he used both the zero-crossings of the smoothed images and the nonzero values of the gradients at these points to reconstruct the image. This algorithm works well in practice, and the conjecture was that these zero-crossings and the values of the gradients determined the image. A counterexample given by Meyer shows that this is not the case. Perfect reconstruction is possible in the one-dimensional case for signals with compact support if the smoothing kernel is the Tukey window, $k(x) = 1 + \cos x$ for $|x| \leq \pi$ and zero elsewhere. An account of the Mallat conjecture and these examples can be found in [131]. Koenderink presents a general and insightful theory of image scale space in [148].

Gaussian smoothing and edge detection. The use of Gaussian filtering in image analysis is so pervasive that it is impossible to point to a "first paper." It is, however, safe to say that David Marr's famous book, *Vision* [177], and the original paper by Hildreth and Marr [179] have had an immeasurable impact on edge detection and image processing in general. The term "edge detection" appeared as early as 1959 in connection with television transmission [134]. The idea that the computation of derivatives of an image necessitates a previous smoothing has been extensively developed by the Dutch school of image analysis [34, 92]. See also the books by Florack [91], Lindeberg [158], and Romeny [255], and the paper [85]. Haralick's edge detector [111], as implemented by Canny [43], is probably the best known image analysis operator. A year after Canny's 1986 paper, Deriche published a recursive implementation of Canny's criteria for edge detection [72].