where Δ denotes del, the (directional) gradient operator from vector algebra, and with the initial condition that $I_0 = I_{x,y}(0)$. The heat equation itself describes the temperature T changing with time t as a function of the thermal diffusivity (related to conduction) κ as

$$\partial T \big/ \partial t = \kappa \nabla^2 T \tag{3.30}$$

and in one-dimensional (1D) form this is

$$\partial T \big/ \partial t = \kappa \frac{\partial^2 T}{\partial x^2} \tag{3.31}$$

so the temperature measured along a line is a function of time, distance, the initial and boundary conditions, and the properties of a material. The relation of this with image processing is clearly an enormous ouch! There are clear similarities between Equations 3.31 and 3.29. They have the same functional form and this allows for insight, analysis and parameter selection. The heat equation (Equation 3.29) is the anisotropic diffusion equation

$$\partial \mathrm{I} \big/ \partial \mathrm{t} = \nabla \cdot \left(c_{x,y}(t) \nabla I_{x,y}(t) \right) \tag{3.32}$$

where $\nabla\cdot$ is the divergence operator (which essentially measures how the density within a region changes), with diffusion coefficient $c_{x,y}$. The diffusion coefficient applies to the local change in the image $\nabla I_{x,y}(t)$ in different directions. If we have a lot of local change, we seek to retain it since the amount of change is the amount of boundary information. The diffusion coefficient indicates how much importance we give to local change: how much of it is retained. (The equation reduces to isotropic diffusion – Gaussian filtering – if the diffusivity is constant, since $\nabla c = 0$.) There is no explicit solution to this equation. By approximating differentiation by differencing (this is explored more in Section 4.2), the rate of change of the image between time step t and time step $t+1$, we have

$$\partial \mathrm{I} \big/ \partial \mathrm{t} = I(t+1) - I(t) \tag{3.33}$$

This implies that we have an iterative solution, and for later consistency we shall denote the image I at time step $t+1$ as $I^{<t+1>} = I(t+1)$, so we then have

$$I^{<t+1>} - I^{<t>} = \nabla \cdot \left(c_{x,y}(t) \nabla I_{x,y}^{<t>} \right) \tag{3.34}$$

and again by approximation, using differences evaluated this time over the four compass directions north, south, east and west, we have

$$\nabla_N(I_{x,y}) = I_{x,y-1} - I_{x,y} \tag{3.35}$$

$$\nabla_S(I_{x,y}) = I_{x,y+1} - I_{x,y} \tag{3.36}$$

$$\nabla_E(I_{x,y}) = I_{x-1,y} - I_{x,y} \tag{3.37}$$

$$\nabla_W(I_{x,y}) = I_{x+1,y} - I_{x,y} \tag{3.38}$$

The template and weighting coefficients for these are shown in Figure 3.26.

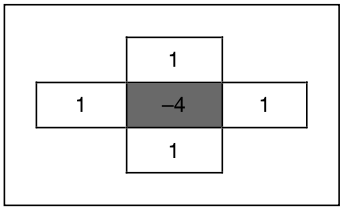

Figure 3.26 Approximations by spatial difference in anisotropic diffusion

When we use these as an approximation to the right-hand side in Equation 3.34, we then have $\nabla \cdot \left(c_{x,y}(t)\nabla I_{x,y}^{<t>}\right) = \lambda\left(cN_{x,y}\nabla_N(I) + cS_{x,y}\nabla_S(I) + cE_{x,y}\nabla_E(I) + cW_{x,y}\nabla_W(I)\right)$, which gives

$$I^{<t+1>} - I^{<t>} = \lambda\left(cN_{x,y}\nabla_N(I) + cS_{x,y}\nabla_S(I) + cE_{x,y}\nabla_E(I) + cW_{x,y}\nabla_W(I)\right)\Big|I = I_{x,y}^{<t>} \quad (3.39)$$

where $0 \leq \lambda \leq 1/4$ and where $cN_{x,y}$, $cS_{x,y}$, $cE_{x,y}$ and $cW_{x,y}$ denote the conduction coefficients in the four compass directions. By rearrangement of this we obtain the equation that we shall use for the anisotropic diffusion operator

$$I^{<t+1>} = I^{<t>} + \lambda\left(cN_{x,y}\nabla_N(I) + cS_{x,y}\nabla_S(I) + cE_{x,y}\nabla_E(I) + cW_{x,y}\nabla_W(I)\right)\Big|I = I_{x,y}^{<t>} \quad (3.40)$$

This shows that the solution is *iterative*: images at *one* time step (denoted by $^{<t+1>}$) are computed from images at the *previous* time step (denoted $^{<t>}$), given the initial condition that the first image is the original (noisy) image. Change (in time and in space) has been approximated as the difference between two adjacent points, which gives the iterative equation and shows that the new image is formed by adding a controlled amount of the local change consistent with the main idea: that the smoothing process retains some of the boundary information.

We are not finished yet, though, since we need to find values for $cN_{x,y}$, $cS_{x,y}$, $cE_{x,y}$ and $cW_{x,y}$. These are chosen to be a function of the difference along the compass directions, so that the boundary (edge) information is preserved. In this way we seek a function that tends to zero with increase in the difference (an edge or boundary with greater contrast) so that diffusion does not take place across the boundaries, keeping the edge information. As such, we seek

$$\begin{aligned} cN_{x,y} &= g\left(\|\nabla_N(I)\|\right) \\ cS_{x,y} &= g\left(\|\nabla_S(I)\|\right) \\ cE_{x,y} &= g\left(\|\nabla_E(I)\|\right) \\ cW_{x,y} &= g\left(\|\nabla_W(I)\|\right) \end{aligned} \quad (3.41)$$

and one function that can achieve this is

$$g(x,k) = e^{-x^2/k^2} \quad (3.42)$$

[There is potential confusion with using the same symbol as for the Gaussian function (Equation 3.24), but we have followed the original authors' presentation.] This function clearly has the desired properties since when the values of the differences ∇ are large the function g is very small; conversely, when ∇ is small then g tends to unity. k is another parameter whose

value we have to choose: it controls the rate at which the conduction coefficient decreases with increasing difference magnitude. The effect of this parameter is shown in Figure 3.27. Here, the solid line is for the smaller value of k and the dotted one is for a larger value. Evidently, a larger value of k means that the contribution of the difference reduces less than for a smaller value of k. In both cases, the resulting function is near unity for small differences and near zero for large differences, as required. An alternative to this is to use the function

$$g2(x, k) = \frac{1}{1 + {}^{x^2}/_{k^2}} \tag{3.43}$$

which has similar properties to the function in Equation 3.42.

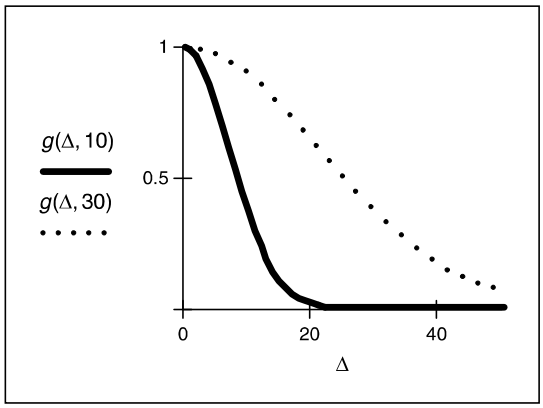

Figure 3.27 Controlling the conduction coefficient in anisotropic diffusion

This all looks rather complicated, so let's recap. First, we want to filter an image by retaining boundary points. These are retained according to the value of k chosen in Equation 3.42. This function is operated in the four compass directions, to weight the brightness difference in each direction (Equation 3. 41). These contribute to an iterative equation which calculates a new value for an image point by considering the contribution from its four neighbouring points (Equation 3.40). This needs the choice of one parameter, λ. Further, we need to choose the number of iterations for which calculation proceeds. For information, Figure 3.25(b) was calculated over 20 iterations and we need to use sufficient iterations to ensure that convergence has been achieved. We also need to choose values for k and λ. By analogy, k is the conduction coefficient and low values preserve edges and high values allow diffusion (conduction) to occur; and how much smoothing can take place. The two parameters are interrelated, although λ largely controls the amount of smoothing. Given that low values of either parameter means that no filtering effect is observed, we can investigate their effect by setting one parameter to a high value and varying the other. In Figure 3.28(a)–(c) we use a high value of k, which means that edges are not preserved, and we can observe that different values of λ control the amount of smoothing. (A discussion of how this Gaussian filtering process is achieved can be inferred from Section 4.2.4.) Conversely, we can see how different values for k control the level of edge preservation in Figure 3.28(d)–(f), where some structures around the eye are not preserved for larger values of k.

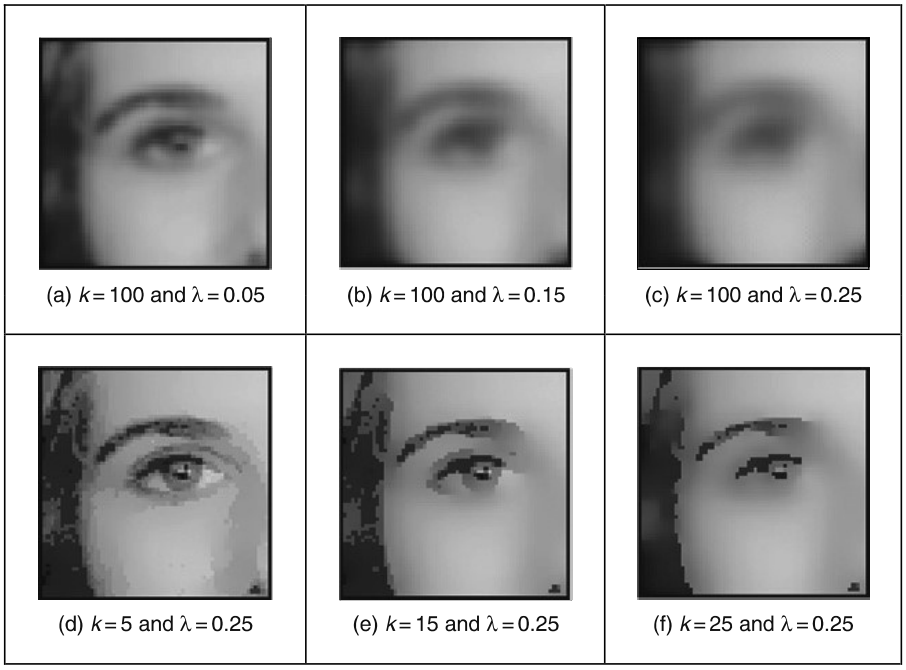

Figure 3.28 Applying anisotropic diffusion

The original presentation of anisotropic diffusion (Perona and Malik, 1990) is extremely lucid and well worth a read if you consider selecting this technique. It has greater detail on formulation and on analysis of results than space here allows for (and is suitable at this stage). Among other papers on this topic, one (Black et al., 1998) studied the choice of conduction coefficient leading to a function which preserves sharper edges and improves automatic termination. As ever, with techniques that require much computation there have been approaches that speed implementation or achieve similar performance more rapidly (e.g. Fischl and Schwartz, 1999).

3.5.5 Force field transform

There are many more image filtering operators; we have so far covered those that are among the most popular. Others offer alternative insight, sometimes developed in the context of a specific application. By way of example, Hurley developed a transform called the force field transform (Hurley et al., 2002, 2005) which uses an analogy to gravitational force. The transform pretends that each pixel exerts a force on its neighbours which is inversely proportional to the square of the distance between them. This generates a force field where the net force at each point is the aggregate of the forces exerted by all the other pixels on a 'unit test pixel' at that point. This very large-scale summation affords very powerful averaging which reduces the effect of noise. The approach was developed in the context of ear biometrics, recognizing people by their ears, which has unique advantage as a biometric in that the shape of people's ears does not change with age, and of course, unlike a face, ears do not smile! The force field transform of an ear (Figure 3.29a) is shown in Figure 3.29(b). Here, the averaging process is reflected in the reduction of the effects of hair. The transform itself has highlighted ear structures, especially the top of the ear and the lower 'keyhole' (the notch).

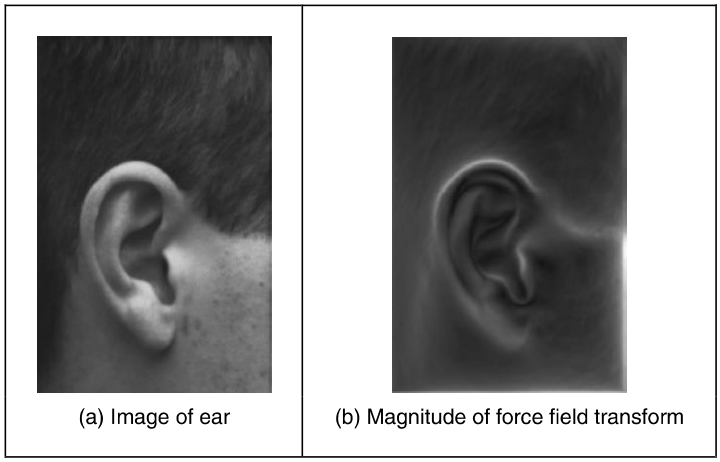

Figure 3.29 Illustrating the force field transform

The image shown is the magnitude of the force field. The transform itself is a vector operation, and includes direction (Hurley, 2002). The transform is expressed as the calculation of the force $\mathbf{F}$ between two points at positions $\mathbf{r}_i$ and $\mathbf{r}_j$, which is dependent on the value of a pixel at point $\mathbf{r}_i$ as

$$\mathbf{F}_i\left(\mathbf{r}_j\right) = \mathbf{P}\left(\mathbf{r}_i\right)\frac{\mathbf{r}_i - \mathbf{r}_j}{\left|\mathbf{r}_i - \mathbf{r}_j\right|^3} \tag{3.44}$$

which assumes that the point $\mathbf{r}_j$ is of unit 'mass'. This is a directional force (which is why the inverse square law is expressed as the ratio of the difference to its magnitude cubed) and the magnitude and directional information has been exploited to determine an ear 'signature' by which people can be recognized. In application, Equation 3.44 can be used to define the coefficients of a template that is convolved with an image (implemented by the FFT to improve speed); as with many of the techniques that have been covered in this chapter; a Mathcad implementation is also given (Hurley et al., 2002). Note that this transform exposes low-level features (the boundaries of the ears), which is the focus of the next chapter. How we can determine shapes is a higher level process, and the processes by which we infer or recognize identity from the low- and the high-level features will be covered in Chapter 8.

3.5.6 Comparison of statistical operators

The different image filtering operators are shown by way of comparison in Figure 3.30. All operators are 5×5 and are applied to the earlier ultrasound image (Figure 3.24a). Figure 3.30(a)–(d) are the result of the mean (direct averaging), Gaussian averaging, median and truncated median, respectively. We have just shown the advantages of anisotropic diffusion compared with Gaussian smoothing, so we will not repeat them here. Each operator shows a different performance: the mean operator removes much noise, but blurs feature boundaries; Gaussian averaging retains more features, but shows little advantage over direct averaging (it is not Gaussian-distributed noise anyway); the median operator retains some noise, but with clear feature boundaries; and the truncated median removes more noise, but along with picture detail. Clearly, the increased size of the truncated median template, by the results in Figure 3.24(b)