PDEs for Image Processing - Lecture 2

Martin Robinson

August 21, 2014

1 Anisotropic Diffusion

The classic isotropic diffusion equation uses a diffusion parameter D which is constant over the entire domain. While this smooths out noise, it also smooths out image features, most notably edges (jumps in pixel intensity). If we want to preserve an edge Γ, we wish to reduce diffusion across (or perpendicular) to the edge while keeping normal diffusion along (tangential) to the edge. We therefore need to identify the perpendicular and tangential directions near an edge in the image. We can do this using the grandient vector of u, which we know to be (a) large near Γ and (b) perpendicular to Γ. Hence the unit vector N normal to Γ and the tangential vector T are given by

$$N = \frac{1}{|\nabla u|}\nabla u = \frac{1}{|\nabla u|}\begin{pmatrix} u_x \\ u_y \end{pmatrix} \tag{1}$$

$$T = N^{\perp} = \frac{1}{|\nabla u|}\begin{pmatrix} -u_y \\ u_x \end{pmatrix} \tag{2}$$

We can then define the first and second derivatives of u with respect to N and T.

$$\begin{align} u_N &= N \cdot \nabla u, \tag{3}\\ u_{NN} &= N \cdot H(u)N, \tag{4}\\ u_T &= T \cdot \nabla u, \tag{5}\\ u_{TT} &= T \cdot H(u)T, \tag{6}\\ & \tag{7} \end{align}$$

where $H(u)$ is the hessian matrix of u

$$H(u) = \begin{pmatrix} u_{xx} & u_{xy} \\ u_{xy} & u_{yy} \end{pmatrix} \tag{8}$$

Note that $u_T = 0$, since the gradient of u is zero tangential to the edge.

Now that we have defined a new local coordinate system in terms of N and T, we can use this to construct an anisotropic diffusion equation, where the pixel values diffuse differently either normal or tangent to an edge. We therefore redefine the diffusion constant K to be a 2x2 matrix. For example,

$$K = \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix}. \tag{9}$$

Defining $\nabla u = \begin{pmatrix} u_N \\ u_T \end{pmatrix}$, our anisotropic heat equation then becomes

$$u_t = \nabla \cdot (K \nabla u) = \nabla \cdot \left(\begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix} \begin{pmatrix} u_N \\ u_T \end{pmatrix} \right) \tag{10}$$

$$u_t = \alpha u_{NN} + \beta u_{TT} \tag{11}$$

As you can see, now we have two parameters for the equation, α and β. These control the diffusive strength normal to and tangential to the edges in the image. So if we wished to remove diffusion across the edge then we could set $\alpha = 0$.

Before we can use finite differences to solve this equation, we need to get it in terms involving u and its derivatives with respect to the normal Cartesian coordinates x and y.

$$u_t = \alpha u_{NN} + \beta u_{TT} \tag{12}$$

$$u_t = \alpha N \cdot H(u) N + \beta T \cdot H(u) T \tag{13}$$

$$u_t = \alpha \frac{u_x^2 u_{xx} + 2u_x u_y u_{xy} + u_y^2 u_{yy}}{u_x^2 + u_y^2} + \beta \frac{u_y^2 u_{xx} - 2u_x u_y u_{xy} + u_x^2 u_{yy}}{u_x^2 + u_y^2} \tag{14}$$

$$u_t = \frac{A u_{xx} + B u_{xy} + C u_{yy}}{u_x^2 + u_y^2}, \tag{15}$$

where

$$A = \alpha u_x^2 + \beta u_y^2 \tag{16}$$

$$B = (\alpha - \beta) u_x u_y \tag{17}$$

$$C = \beta u_x^2 + \alpha u_y^2 \tag{18}$$

We can discritize the single derivatives using a *central difference* $D_c^x u \approx u_x$

$$u_x \approx D_c^x u = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x}, \tag{19}$$

The mixed derivative u_{xy} can be constructed by using the central difference operator twice on x and y

$$u_{xy} = (u_x)_y \approx D_c^y (D_c^x u), \tag{20}$$

which can be evaluated as

$$D_c^y (D_c^x u) = \frac{\frac{u_{i+1,j+1} - u_{i-1,j+1}}{2\Delta x} - \frac{u_{i+1,j-1} - u_{i-1,j-1}}{2\Delta x}}{2\Delta y}, \tag{21}$$

$$= \frac{u_{i+1,j+1} - u_{i-1,j+1} - u_{i+1,j-1} + u_{i-1,j-1}}{4 \Delta x \Delta y}. \tag{22}$$

As before, the double derivatives can be constructed using a combination of *forward difference* D_+^x and *backwards difference* D_-^x operators.

$$u_x \approx D_+^x u = \frac{u_{i+1,j} - u_{i,j}}{\Delta x}, \tag{23}$$
$$u_x \approx D_-^x u = \frac{u_{i,j} - u_{i-1,j}}{\Delta x}, \tag{24}$$
$$u_{xx} \approx D_-^x(D_+^x u), \tag{25}$$

where

$$D_-^x(D_+^x u) = \frac{\frac{u_{i+1,j}-u_{i,j}}{\Delta x} - \frac{u_{i,j}-u_{i-1,j}}{\Delta x}}{\Delta x}, \tag{26}$$
$$= \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2}. \tag{27}$$

Substitute these approximations back into Eq. 13 and choose a suitable discritization for the time derivative to numerically calculate the solution to the anisotropic diffusion equation.

However, notice that we still have the diffusion parameters α and β as constants. Another option is to have these parameters depend on x and y, or even some property of the image, such as $|\nabla u|$. In the next section we will examine the effect of spatially varying diffusion.

2 Spatially Varying Diffusion

The heat equation $u_t = K\nabla u$ involves a diffusion constant K. We could replace that constant with a function of space

$$u_t = \nabla \cdot (K(\mathbf{x})\nabla u). \tag{28}$$

In one dimension, this would be

$$u_t = (K(x)u_x)_x. \tag{29}$$

A common approach to discretizing this is to use a *forward difference* $D_+^x \approx u_x$ on the u_x term

$$u_x \approx D_+^x u = \frac{u_{i+1} - u_i}{\Delta x}, \tag{30}$$

then evaluate $K(x)$ at the midpoint: $K(x_{i+\frac{1}{2}})$, so that

$$K(x)u_x \approx K(x_{i+\frac{1}{2}})\frac{u_{i+1} - u_i}{\Delta x}, \tag{31}$$

Then, differentiate the result approximately with a *backwards difference* D_-^x.

$$u_x \approx D_-^x u = \frac{u_i - u_{i-1}}{\Delta x}, \tag{32}$$

so that

$$(K(x)u_x)_x \approx D_-^x(K(x_{i+\frac{1}{2}})D_+^x u) = \frac{K(x_{i+\frac{1}{2}})\frac{u_{i+1}-u_i}{\Delta x} - K(x_{i-\frac{1}{2}})\frac{u_i-u_{i-1}}{\Delta x}}{\Delta x} \tag{33}$$

$$\approx \frac{K(x_{i+\frac{1}{2}})u_{i+1} - (K(x_{i+\frac{1}{2}}) + K(x_{i-\frac{1}{2}}))u_i + K(x_{i-\frac{1}{2}})u_{i-1}}{\Delta x^2} \tag{34}$$

Naturally, for constant diffusion $K(x) = K$ this reduces to

$$(K(x)u_x)_x \approx K\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} \tag{35}$$

For two dimensions, the spatially varying heat equation is

$$u_t = (K(x,y)u_x)_x + (K(x,y)u_y)_y. \tag{36}$$

and this can be discretized in a similar fashion to give

$$(K(x,y)u_x)_x \approx \frac{K(x_{i+\frac{1}{2}}, y_j)u_{i+1,j} - (K(x_{i+\frac{1}{2}}, y_j) + K(x_{i-\frac{1}{2}}, y_j))u_{i,j} + K(x_{i-\frac{1}{2}}, y_j)u_{i-1,j}}{\Delta x^2} \tag{37}$$

$$(K(x,y)u_y)_y \approx \frac{K(x_i, y_{j+\frac{1}{2}})u_{i,j+1} - (K(x_i, y_{j+\frac{1}{2}}) + K(x_i, y_{j-\frac{1}{2}}))u_{i,j} + K(x_i, y_{j-\frac{1}{2}})u_{i,j-1}}{\Delta y^2} \tag{38}$$

$$u_t \approx \frac{u^{n+1} - u^n}{\Delta t}. \tag{39}$$

If only the nodal values for $K(x,y)$ are known, then a reasonable approximation is to use the average of two neighbouring grid points

$$K(x_i, y_{j+\frac{1}{2}}) = \frac{1}{2}(K_{i,j+1} + K_{i,j}) \tag{40}$$

$$K(x_{i+\frac{1}{2}}, y_j) = \frac{1}{2}(K_{i+1,j} + K_{i,j}) \tag{41}$$

Which results in

$$(K(x,y)u_x)_x \approx \frac{(K_{i+1,j} + K_{i,j})u_{i+1,j} - ((K_{i+1,j} + 2K_{i,j} + K_{i-1,j}))u_{i,j} + (K_{i-1,j} + K_{i,j})u_{i-1,j}}{2\Delta x^2} \tag{42}$$

$$(K(x,y)u_y)_y \approx \frac{(K_{i,j+1} + K_{i,j})u_{i,j+1} - ((K_{i,j+1} + 2K_{i,j} + K_{i,j-1}))u_{i,j} + (K_{i,j-1} + K_{i,j})u_{i,j-1}}{2\Delta y^2} \tag{43}$$

$$u_t \approx \frac{u^{n+1} - u^n}{\Delta t}. \tag{44}$$

For $\Delta x = \Delta y = 1$ this simplifies to

$$u_{i,j}^{n+1} = u_{i,j}^n + \frac{1}{2}\Delta t(\tag{45}$$

$$(K_{i+1,j} + K_{i,j})u_{i+1,j} + (K_{i-1,j} + K_{i,j})u_{i-1,j} \tag{46}$$

$$(K_{i,j+1} + K_{i,j})u_{i,j+1} + (K_{i,j-1} + K_{i,j})u_{i,j-1} \tag{47}$$

$$- (K_{i+1,j} + K_{i-1,j} + K_{i,j+1} + K_{i,j-1} + 4K_{i,j})u_{i,j}) \tag{48}$$