Anisotropic Diffusion

Martin Robinson

August 11, 2015

Image Domain

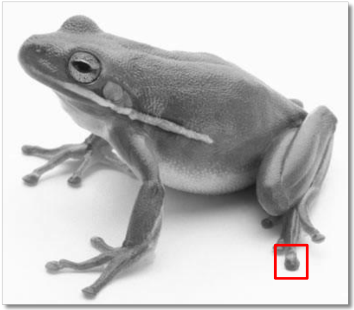
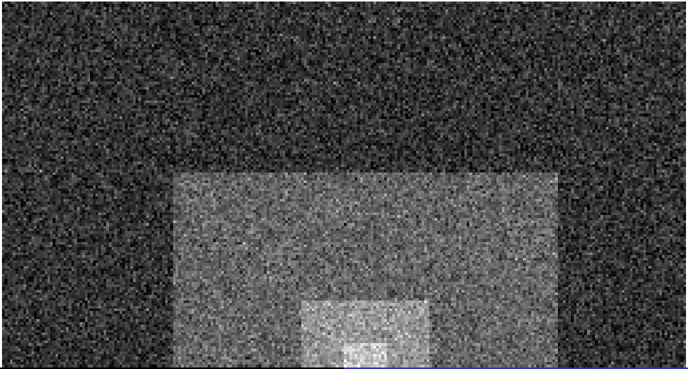


Figure : Digital image $u_{i,j}$ is a discretised and quantised version of "real" scene $u(x, y)$

Anisotropic Diffusion

- Motivation: diffusion parameter D is constant over the entire domain. While this smooths out noise, it also smooths out image features, most notably edges
- Goal: we wish to reduce diffusion across (or perpendicular) to the edge while keeping normal diffusion along (tangential) to the edge.



New coordinate system near edge

- The unit vector N normal to Γ and the tangential vector T are given by

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

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

$$\begin{aligned} u_N &= N \cdot \nabla u, \\ u_{NN} &= N \cdot H(u) N, \\ u_T &= T \cdot \nabla u, \\ u_{TT} &= T \cdot H(u) T. \end{aligned}$$

Anisotropic diffusion tensor

- Define new diffusion constant to be used in the coordinate system of N and T

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

- Defining $\nabla u = \begin{pmatrix} u_N \\ u_T \end{pmatrix}$, the heat equation becomes

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

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

- (**exercise**) The above in terms of x and y derivatives is

$$u_t = \frac{Au_{xx} + Bu_{xy} + Cu_{yy}}{u_x^2 + u_y^2},$$

where

$$\begin{aligned} A &= \alpha u_x^2 + \beta u_y^2 \\ B &= (\alpha - \beta) u_x u_y \\ C &= \beta u_x^2 + \alpha u_y^2 \end{aligned}$$

Finite Difference

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

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

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

$$u_{xy} = (u_x)_y \approx D_c^y(D_c^x u) \quad = \frac{u_{i+1,j+1} - u_{i-1,j+1} - u_{i+1,j-1} + u_{i-1,j-1}}{4\Delta x \Delta y}.$$

- Note: can derive u_{xx} and u_{yy} using a combination of *forward difference* D_+^x and *backwards difference* D_-^x operators.

$$u_{xx} \approx D_-^x(D_+^x u) = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} \quad u_{yy} \approx D_-^y(D_+^y u) = \frac{u_{i,j+1} - 2u}{\Delta y}$$

MATLAB code

Spatially varying diffusion

- Diffusion flux at point (x, y)

$$J(x, y) = K(x, y)\nabla u$$

- Heat equation becomes

$$\begin{aligned} u_t &= \nabla \cdot J(x, y), \\ &= \nabla \cdot (K(x, y)\nabla u). \end{aligned}$$

Spatially varying diffusion - 1D

- In one dimension, this would be

$$u_t = (K(x)u_x)_x. \tag{1}$$

- A common approach to discretizing this is to use a *forward difference* $D^x_+ \approx u_x$ on the u_x term, evaluate $K(x)$ at the midpoint: $K(x_{i+\frac{1}{2}})$, and then use a *backwards difference* to calculate the outside gradient. i.e.

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

$$\approx \frac{K(x_{i+\frac{1}{2}})u_{i+1} - (K(x_{i+\frac{1}{2}}) + K(x_{i-\frac{1}{2}}))u_i + K(x_{i-\frac{1}{2}})u_{i-1}}{\Delta x^2} \tag{3}$$

Spatially varying diffusion - 2D

For two dimensions, the spatially varying heat equation is

$$u_t = (K(x,y)u_x)_x + (K(x,y)u_y)_y.$$

- this can be discretized in a similar fashion to give

$$(K(x,y)u_x)_x \approx \frac{K(x_{i+\frac{1}{2}}, y_j)u_{i+1,j} - (K(x_{i+\frac{1}{2}}, y_j) + K(x_{i-\frac{1}{2}}, y_j))u_{i,j} + K(x}{\Delta x^2} \quad (4)$$

$$(K(x,y)u_y)_y \approx \frac{K(x_i, y_{j+\frac{1}{2}})u_{i,j+1} - (K(x_i, y_{j+\frac{1}{2}}) + K(x_i, y_{j-\frac{1}{2}}))u_{i,j} + K(x}{\Delta y^2} \quad (5)$$

$$u_t \approx \frac{u^{n+1} - u^n}{\Delta t}. \quad (6)$$

- 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

$$K(x_i, y_{j+\frac{1}{2}}) = \frac{1}{2}(K_{i,j+1} + K_{i,j}) \tag{7}$$

$$K(x_{i+\frac{1}{2}}, y_j) = \frac{1}{2}(K_{i+1,j} + K_{i,j}) \tag{8}$$

- Which results in

$$(K(x,y)u_x)_x \approx \frac{(K_{i+1,j}+K_{i,j})u_{i+1,j} - ((K_{i+1,j}+2K_{i,j}+K_{i-1,j}))u_{i,j} + (K}{2\Delta x^2} \quad (9)$$

$$(K(x,y)u_y)_y \approx \frac{(K_{i,j+1}+K_{i,j})u_{i,j+1} - ((K_{i,j+1}+2K_{i,j}+K_{i,j-1}))u_{i,j} + (K}{2\Delta y^2} \quad (10)$$

$$u_t \approx \frac{u^{n+1}-u^n}{\Delta t}. \quad (11)$$

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

$$
\begin{aligned}
u_{i,j}^{n+1} = & u_{i,j}^{n} + \frac{1}{2}\Delta t(&& (12)\\
& (K_{i+1,j} + K_{i,j})u_{i+1,j} + (K_{i-1,j} + K_{i,j})u_{i-1,j} && (13)\\
& (K_{i,j+1} + K_{i,j})u_{i,j+1} + (K_{i,j-1} + K_{i,j})u_{i,j-1} && (14)\\
& -(K_{i+1,j} + K_{i-1,j} + K_{i,j+1} + K_{i,j-1} + 4K_{i,j})u_{i,j}) && (15)
\end{aligned}
$$