TLM for vision systems

Richard Harvey and Ciarán Kenny
School of Information Systems, University of East Anglia,
Norwich, NR4 7TJ, UK.
Tel: +44 (1603) 59-3257, FAX: -3345, Email: rwh@sys.uea.ac.uk

Abstract

Anisotropic diffusion is an increasingly important technique in primary vision systems. It requires the solution of a nonlinear diffusion equation. The transmission line matrix (TLM) technique is used to solve the anisotropic diffusion equation explicitly. Results are presented to compare the effect of two possible nonlinearities.

Introduction

Processing an image for recognition, classification or coding is a complicated and computationally expensive task. One of the main goals of image processing is to reduce the complexity of the image in a controlled way, so that the high-level vision system can perform its task in a reasonable time. The machine that simplifies the image is called a *primary vision system.*

Since edges are important in recognising objects it is important that the primary vision system should not introduce new edges as the image is simplified. It has been suggested that a suitable method for this task would be to use the image intensity (the pixel values) as the initial conditions for a diffusion equation $u_{xx} + u_{yy} = cu_t$ where c is the conduction coefficient. The time, t, is associated with *scale*: the longer the algorithm runs, the more that small-scale features are blurred and large-scale features emphasized [1].

Care is needed when discretising the diffusion equation since it is required that no new maxima or minima are introduced [2] (this is the discrete two-dimensional equivalent of requiring no new edges). One such discretisation is

$$\begin{aligned} u^{k+1}_{x,y} &= (1-(2-\gamma)\,\Delta t)\,u^k_{x,y} + \frac{1}{2}(1-\gamma)\Delta t\left(u^k_{x-1,y} + u^k_{x+1,y} + u^k_{x,y-1} + u^k_{x,y+1}\right) \\ &\quad + \frac{1}{4}\gamma\Delta t\left(u^k_{x-1,y-1} + u^k_{x-1,y+1} + u^k_{x+1,y-1} + u^k_{x+1,y+1}\right) \end{aligned} \tag{1}$$

where $u^k_{x,y}$ is the pixel at discrete position (x, y) and at time $k\Delta t$. The parameter $\gamma \in [0, 1]$ determines whether the Laplacian is approximated by a five-point discrete Laplacian on a square grid $(u_{-1,0} + u_{1,0} + u_{0,-1} + u_{0,1} - 4u_{0,0})$ or a diagonal grid $(\frac{1}{2}(u_{-1,-1} + u_{-1,1} + u_{1,-1} + u_{1,1} - 4u_{0,0}))$. So long as $0 \leq \gamma \leq \frac{1}{2}$ the discrete equation does not introduce new extrema. If $\gamma = 0$ then the Laplacian is restricted to a square grid and it is possible, as an alternative to the explicit finite difference scheme, to convolve along the x and y directions with the discrete kernel $T(\zeta, t) = e^{-t}I_\zeta(t)$ where $I_\zeta(t)$ is the modified Bessel function of order ζ.

Figure 1 shows the results of using a 480 by 640 pixel image (Figure 1(a)) as the initial condition for a TLM solution of the diffusion equation [3].

The original image had 256 grey levels but the ones shown here have been histogram equalised to 64 levels to improve the reproduction quality. The same equalisation has been applied to all images.

The idea is that an intelligent vision system might, at each scale, convert the greyscale to a binary image and then track the resulting *blobs* back through scale space. In Figure 1(d) for example, the large white blob is associated with the person on the right of the picture.

Figure 1: Original image, (a), after, (b) 100, (c) 500, and (d) 1000 iterations

Anisotropic diffusion

Figure 1 reveals a well known problem. If edges are important then it is annoying that the vision system, which was designed not to introduce new edges, achieves this by blurring existing ones. What is needed is a system that reduces detail (preserves *scale-space causality*) but doesn't corrupt original edges. Several systems have been proposed, but one of the most widely used is a modification to the diffusion equation called *anisotropic diffusion* [4]. The usual equation is

$$u_t = \nabla \cdot (c(x,y,t)\nabla u) = c(x,y,t)\nabla^2 u + \nabla c(x,y,t) \cdot \nabla u \tag{2}$$

The conductivity, c, is adjusted so that diffusion is minimised at positions where the gradient of the intensity is large and is usually chosen as some nonlinear function of the local gradient, $c(x,y,t) = g\left(|\nabla u|^2\right)$. Some popular choices are $g(\xi) = \exp\{-(\xi/K)^2\}$ or $g(\xi) = \left(1+(\xi/K)^2\right)^{-1}$. Since the algorithm is nonlinear it is not possible to use linear filtering and the diffusion equation has to be solved directly. In this paper the following TLM scheme has been used:

$$\begin{aligned}
u(x,y) &= \frac{2\,(R_s+Z_s)}{1+R+4\,(R_s+Z_s)}\sum_{j=1}^{4} v_j^{(i)}(x,y) + \frac{2\,(1+R)}{1+R+4\,(R_s+Z_s)} v_5^{(i)}(x,y) \\
v_j^{(r)}(x,y) &= g_j(x,y)\frac{5\lambda}{1+R}u(x,y) + \frac{R-1}{R+1}v_j^{(i)}(x,y), \qquad j=1,\ldots,4 \\
v_5^{(r)}(x,y) &= \left(1-\lambda\sum_{j=1}^{4} g_j(x,y)\right)\frac{5Z_s}{R_s+Z_s}u(x,y) + \frac{R_s-Z_s}{R_s+Z_s}v_5^{(i)}(x,y)
\end{aligned} \tag{3}$$

where $v_j^{(i)} (j=1,\ldots,4)$ are the incident pulses on the connecting lines and $j=5$ indicates the stub line. λ is the discretisation parameter ($\lambda = 1/4$ in these simulations). The connection matrix is not shown but is conventional. The time index, which is not shown in (3), changes at the connection stage. The line parameters, R, Z, R_s, Z_s have been chosen equal to unity and the nonlinearity g has been chosen dependent on the first difference; for example, on line 1, $g_1(x,y) = g(|u(x,y) - u(x-1,y)|)$.

The effects of the two nonlinear conduction functions are illustrated in Figure 2 which is a one-dimensional slice through Figure 1(a) at line 380. The x-axis is the pixel coordinate, the y-axis is the image intensity normalised to lie in [0,1].

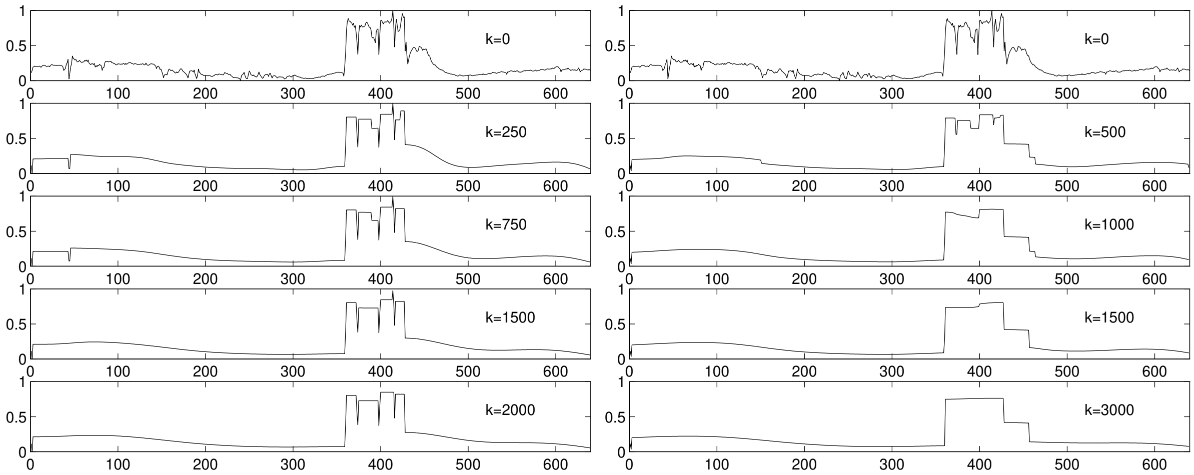

Figure 2: Nonlinear diffusion with, left, $g(x) = \exp\{-(x/K)^2\}, K = 0.05$ and, right, $g(x) = (1 + (x/K)^2)^{-1}, K = 0.0095$.

K is a constant chosen to allow a tradeoff between diffusion time and structure preservation. It can be chosen by eye, as we have done, or using the Canny noise estimator [5]. However a significant problem with anisotropic diffusion is that, without careful bookkeeping, it is easy to use more computation in the calculation of K than in the simplification of the images.

Referring to Figure 2, the effect of the exponential function is to preserve edges whereas the effect of the polynomial function is to preserve regions.

Results

Both of these conductivity functions have been evaluated on real images. Figure 3 shows the result of anisotropic diffusion with the edge-preserving conductivity function and Figure 4 shows a similar thing but with the region-preserving conductivity.

Figure 3: Anisotropic diffusion using edge-favouring conductivity. Original image (a), 50 iterations (b), 500 iterations (c), 1500 iterations (d), $K = 0.04$

Since this is a nonlinear algorithm, different levels of simplification are achieved at different times so the iteration counts in Figure 3 and Figure 4 differ. The boundary condition[1] for the image is $c = 0$.

Figure 4: Anisotropic diffusion using width-favouring conductivity. Original image (a), 50 iterations (b), 150 iterations (c), 500 iterations (d), $K = 0.04$

Remarks

TLM is an appropriate and efficient algorithm for solving the anisotropic diffusion equation but some care is needed with the discretisation. For example it is well known that the TLM algorithm for diffusion can introduce wave equation solutions [6] and sometimes this is done deliberately to reduce computation time but, in this application, these need to be avoided because the main purpose of a primary vision system is to simplify the image without introducing new extrema.

References

[1] J.J.Koenderink, *The structure of images*, Biological Cybernetics, 1984, 50, pp 363-370.

[2] T.Lindeberg, Scale-space theory in Computer Vision, Kluwer, 1994.

[3] P.B.Johns, *A simple explicit and unconditionally stable numerical routine for the solution of the diffusion equation*, Int. J. Num. Modell., 11, 1977, pp 1303-1328.

[4] P.Perona and J.Malik, *Scale-space and Edge Detection using Anisotropic Diffusion*, IEEE Trans. Patt. Anal. Mach. Intell., 12, (7), July 1990, pp 629 - 639.

[5] P.Perona, T.Shiota and J.Malik, *Anisotropic Diffusion* in Geometry-Driven Diffusion in Computer Vision, Ed B.M. ter Harr Romeny, Kluwer, 1994, pp 73-92.

[6] P.Enders and D. de Cogan, *The efficiency of Transmission Line Matrix modelling: a rigorous viewpoint*, Int. J. Num. Modell., 6, 1993, pp 109-126.

[1] There is some confusion about the correct boundary conditions. In [4] all the boundaries have $c = 0$ but in [5] the implementation of this algorithm appears to have two $c = 0$ boundaries and two $u_x, u_y = 0$ boundaries.