SCI INSTITUTE
TECHNICAL REPORT

An Efficient, Geometric Multigrid Solver for the Anisotropic Diffusion Equation in Two and Three Dimensions

Tolga Tasdizen, Ross Whitaker

UUSCI-2004-002

Scientific Computing and Imaging Institute
University of Utah
Salt Lake City, UT 84112 USA

June 14, 2004

Abstract:

We propose a new geometric multigrid solver for anisotropic image diffusion. Anisotropic diffusion in image processing has been widely accepted as a denoising method; however, the large computation times for large volumes are prohibitive for interactive exploration of the parameter space. Our approach is able to reduce computation times via a new method for restricting the anisotropic diffusion operator to coarser grids for the multigrid solver. This operator restriction is based on computing equivalent conductance between two nodes in an electrical circuit.

THE UNIVERSITY OF UTAH

An Efficient, Geometric Multigrid Solver for the Anisotropic Diffusion Equation in Two and Three Dimensions

Tolga Tasdizen Ross Whitaker

UUSCI-2004-2

Chapter 1

Introduction

Anisotropic diffusion has been widely accepted as a preprocessing method for reducing noise while preserving boundaries and edges between different regions of an image. The original form of the anisotropic diffusion partial differential equation (PDE) was proposed by Perona and Malik [1]. Since then, a considerable amount of research has been done on anisotropic diffusion and various modifications to the original equation have been made for different applications; for a more in depth study of anisotropic diffusion, we refer the reader to [2, 3].

One of the main drawbacks of PDE based image processing methods, including anisotropic diffusion, has been their relatively large computational burden. With the ever increasing speed of modern CPUs, the computational cost for typical 2D images has come down to reasonable levels. With reasonable levels, we mean that a user can, in a single session, set the parameters of the PDE, let it run for a short amount of time (typically less than a few minutes), look at the results, adjust the parameters and repeat this as necessary. However, the same can not be said of the much larger 3 dimensional images that are becoming commonplace.

One of the important application areas of anisotropic diffusion has traditionally been medical image analysis, such as the filtering of magnetic resonance imaging (MRI) data [4]. MRI and other medical images are typically large 3D volumes that require significant computational expense to filter with PDE based methods. In this technical report, we introduce a novel multigrid solver for obtaining very fast solutions to anisotropic diffusion PDEs.

Multigrid methods [5] have been used commonly in many areas. In the field of image processing, they have been used to solve relaxation problems [6, 7] and the Perona and Malik (P&M) anisotropic diffusion equation [8]. Acton was able to provide significant computational savings over the regular solution methods [8]. Our approach differs from Acton's in one important aspect: Acton solves the nonlinear P&M anisotropic diffusion PDE using a full approximation scheme while we linearize the equation in time and solve the resulting semi-implicit formulation. Our method provides significant computational savings over Acton's approach due to the linearization. Furthermore, our approach is applicable to several other anisotropic diffusion equations which can not be solved with the methods described in [8].

Chapter 2

Anisotropic Diffusion

Let $\mathbf{f} : \Re^n \rightarrow \Re$ be an image, then the general form of the anisotropic diffusion PDE on f can be written as

$$\frac{\partial f}{\partial t} = \nabla \cdot (c \nabla f) \,, \tag{2.1}$$

where $c : \Re^n \rightarrow \Re$ denotes the spatially varying diffusion coefficients. In general, the diffusion coefficients can be arbitrary. In the P&M anisotropic diffusion PDE the diffusion coefficients are a function of the gradient of f. more specifically, Perona and Malik proposed

$$c_k(\mathbf{x}) = e^{-\left(\frac{\|\nabla f(\mathbf{x})\|}{k}\right)^2} \tag{2.2}$$

in their original paper [1]. The parameter k controls the extent of edge preservation. Many variations to this choice of the diffusion coefficient function have been proposed, but all of these are functions of the gradient of f. The methods proposed in this work are not restricted to any specific choice of the diffusion coefficients, they apply to the more general case described in (2.1). In the rest of this report, we will consider diffusion coefficients given by (2.2). More general cases will be discussed in future research papers.

2.1 Finite Difference Discretization

A typical spatial discretization of (2.1) uses $2n$-point neighborhood stencil where n is the dimensionality of the image. The PDE at point $\mathbf{x} \in \Re^n$ is discretized as

$$\frac{\partial f(\mathbf{x})}{\partial t} = \frac{1}{2n} \sum_{i=1}^{2n} c_i(\mathbf{x}) \left(f(\mathbf{x}_i) - f(\mathbf{x})\right), \tag{2.3}$$

where $\mathbf{x}_i$ enumerate the $2n$ neighbors of $\mathbf{x}$. Also, c_i denotes the diffusion coefficient of the connection between $\mathbf{x}$ and $\mathbf{x}_i$. Then, a simple choice for c_i in accordance with (2.2) is

$$c_i(\mathbf{x}) = e^{-\left(\frac{|f(\mathbf{x}) - f(\mathbf{x}_i)|}{k}\right)^2}. \tag{2.4}$$

2.2 Explicit vs. Implicit Solutions

After the spatial discretization discussed in Section 2.1, the most straightforward way to solve (2.1) is to employ a time step discretization. Let f_k denote the image function f after the k'th time step. Starting with the inital image function f_o, we define

$$f_{k+1}(\mathbf{x}) = f_k(\mathbf{x}) + dt \frac{1}{2n} \sum_{i=1}^{2n} c_{k,i}(\mathbf{x}) \left(f_k(\mathbf{x}_i) - f_k(\mathbf{x})\right), \tag{2.5}$$

where dt is the time increment per step. A maximum allowed value for $dt = 1/2n$ can be derived from XXX conditions []. Then, to obtain the solution after time Δt we repeat (2.5) $k = \Delta t/dt$ times. We use the notation $c_{k,i}$ to clarify that the values of the diffusion coefficients are recomputed at each time step from f_k.

The formulation given by (2.5) is effective for small values of Δt where only a few time steps need to be taken; however, it becomes computationally costly for finding solutions to larger Δt. A computationally attractive alternative is to linearize the PDE and use an implicit formulation. First, given the diffusion coefficients at time k, the right hand side of (2.4) can be rewritten as a linear operator $\mathbf{L}_k$ acting on f_k

$$f_{k+1} = f_k + dt\ \mathbf{L}_k f_k. \tag{2.6}$$

As shown in [9], we can transform the explicit time step equation (2.5) into an implicit equation using a backward Euler approximation

$$\begin{aligned} f_{k+1} &= f_k + dt\ \mathbf{L}_k f_{k+1} && (2.7)\\ f_{k+1} - dt\ \mathbf{L}_k f_{k+1} &= f_k && (2.8)\\ (\mathbf{I} - dt\mathbf{L}_k)\ f_{k+1} &= f_k && (2.9)\\ f_{k+1} &= (\mathbf{I} - dt\mathbf{L}_k)^{-1}\ f_k. && (2.10) \end{aligned}$$

This implicit formulation allows for much larger time step dt. Of course, directly inverting the very large linear system in (2.10) is not feasible. Therefore, other methods such as a conjugate gradient solver or a multigrid solver must be used. The solution of this equation with multigrid methods is the topic of the next chapter.

Chapter 3

Geometric Multigrid Solver for Anisotropic Diffusion

Multigrid methods provide a computationally efficient way to solve linear systems of the form (2.9). This equation can also be solved at the finest level using methods such as conjugate gradient or Jacobi iterations. These methods indirectly obtain the solution (2.10) by minimizing reducing the residual

$$r = f_k - (\mathbf{I} - dt\mathbf{L}_k)\, f_{k+1} \tag{3.1}$$

via iterative updating of f_{k+1}. The motivation for multigrid methods comes from the following observation Jacobi and Gauss-Seidel iterations quickly kill the high spatial frequency portions of the residual image r while lower frequency components take much longer to reduce. A low frequency signal at the finest level will appear to have higher frequency if the image is downsampled. Then, the basic premise of multigrid methods is to move the problem to coarser grids to take advantage of this property. For a in depth review of multigrid methods we refer the reader to [5].

In the rest of this report, we will drop the time step subscript k; f_o will be used to denote the initial image, L will denote the operator, and f will be used to denote the solution. We rewrite (2.9) at a specific grid level (spacing) h as

$$\left(\mathbf{I} - \Delta t\mathbf{L}^h\right) f^h = f_o^h. \tag{3.2}$$

Higher values of h correspond to coarser grids. A multigrid solver consists of one or more *v-cycles* shown in Figure 3.1. Each *v-cycle* starts at the finest grid h. For the first *v-cycle* the initial solution f is set to be equal to f_o. Then, v_1 Gauss-Seidel relaxation operations are performed to iteratively reduce the residual. We have found that Gauss-Seidel relaxation outperforms Jacobi relaxation in our problem. Details on relaxation methods can be found in [5]. The relaxation iterations result in some f at the finest grid level h. The residual is computed at this grid level accoring to the equation

$$r^h = f^h - \left(\mathbf{I} - \Delta t\mathbf{L}^h\right) f_o^h. \tag{3.3}$$

Then, the residual is restricted to grid level 2h. The residual restriction $r^h \xrightarrow{\mathbf{R}} r^{2h}$ is discuseed in Section 3.1. We also need to restrict the operator $\mathbf{L}^h$ to the coarser grid. The operator restriction

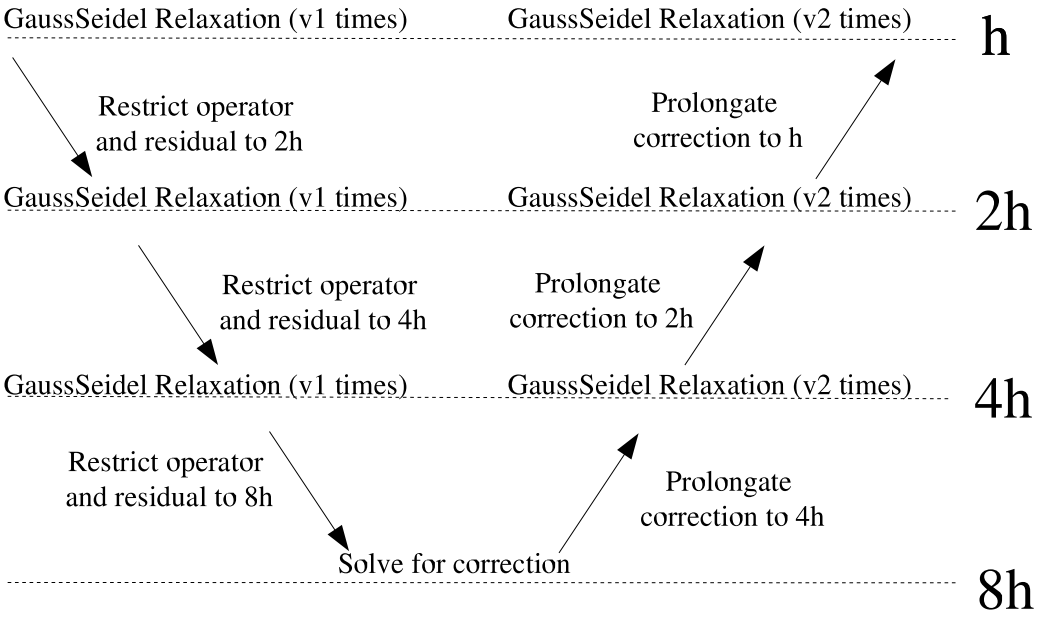


Figure 3.1: Multigrid *v-cycle*

is different than the residual restriction. in this report, we introduce a new approach to restricting the anisotropic diffusion operator that improves performance over standard restriction methods, see Section 3.2.

A two level multigrid method would work by computing a correction term f^{2h} such that

$$\left(\mathbf{I} - \Delta t \mathbf{L}^{2h}\right) f^{2h} = r^{2h}. \tag{3.4}$$

If we have an exact correction term, we can simply upsample it to the finest grid level and add it back onto f^h; however, in practice, the grid level $2h$ is still too complex to solve analytically. Multigrid methods overcome this problem by getting an approximate correction using relaxation operations and repeating the residual restriction and relaxation operations until a grid level simple enough to allow an exact solution is reached. In Figure 3.1, this level is $8h$.

Once a level where an exact solution can be obtained in a computationally feasible manner is reach, we start send the corrections back up the grid hierarchy. In the example shown in Figure 3.1, we start by prolongating the correction term f^{8h}. Let $f^{8h\overset{\mathbf{P}}{\rightarrow}4h}$ denote the prolongation of the correction term. The prolongated correction from $8h$ is then added to the correction term that was obtained at lebel $4h$:

$$f^{4h} \leftarrow f^{4h} + f^{8h\overset{\mathbf{P}}{\rightarrow}4h}. \tag{3.5}$$

Next, Gauss-Seidel relaxation is applied v_2 times on the equation

$$\left(\mathbf{I} - \Delta t \mathbf{L}^{4h}\right) f^{4h} = r^{4h}. \tag{3.6}$$

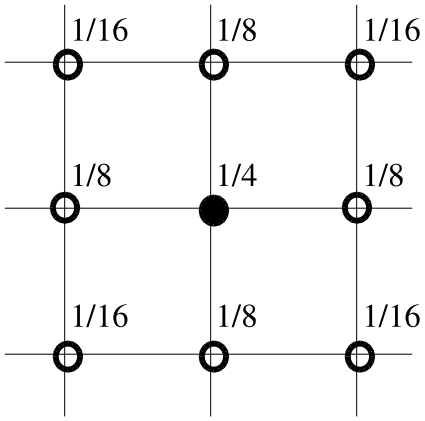


Figure 3.2: Full weight coarsening.

This prolongation/addition/relaxatiom procedure is repeated until the solution at the finest grid level f^h has been updated, which completes the *v-cycle*. Typically, a single *v-cycle* is not sufficient to lower the norm of the residual r^h to acceptable leveles, and it is necessary to perform multiple *v-cycles*. In this case, the result f^h of a *v-cycle* becomes the initialization for the next, and everything else (the operator $\mathbf{L}$ and the right-hand side f_o) remain the same.

The number of relaxation iterations on the down and up legs of the *v-cycle*, v_1 and v_2, are parameters of the algorithm. In [5], the authors suggest using $v_1 = 8$ and $v_2 = 3$ for most applications.

3.1 Residual Restriction

We consider two options for residual restriction: injection and full-weight averaging. In either case, the image at the coarser resolution has pixel locations that correspond to every other pixel in every axis direction of the finer resolution image. With the injection method, the residual image at the coarser resolution is simply defined as $r^{2h}[x, y] = r^h[2x, 2y]$. On the other hand, the full weighting approach uses the averaging kernel shown in Figure 3.2 to compute the value of the pixel at coarse resolution (the center black pixel in the figure) from the pixels at the fine resolution. Both methods easily extend to n dimensions. The relative advantages of the two methods will be examined in Chapter 4.

3.2 Conductance Restriction in 2D

The multigrid approach requires solving linear systems defined on different grids as in (3.4). Therefore, the operator $\mathbf{L}$ has to be defined for all grids. A simple approach is to use the same methods as

residual restriction (Section 3.1) for the operator restriction as well. In this work, we propose a novel approach that will be shown to speed up convergence of the multigrid algorithm to the solution, see Chapter 4.

The attractiveness of the Perona & Malik anisotropic diffusion PDE is its ability to denoise images while preserving edges. If the anisotropic diffusion operator $\mathbf{L}^h$ is coarsened with injection or full weighting restriction, the edges in the image are inevitably weakened at the coarser grid. Boundaries between regions of an image appear as a thin layer of low conductance values separating high conductance regions. The full weighting kernel, Figure 3.2, will average the low conductance values with higher values, resulting in progressively weaker edges as we move down the multigrid hierarchy. Similarly, injection methods can completely lose the low conductance boundary values. Even though, both of these operator coarsening strategies still work as will be shown in Chapter 4, they waste computational effort by solving for systems at coarser grids that do not correspond well to the system on the fine grid.

The multigrid solution to the Laplace equation with spatially varying weights requires non-standard restriction and prolongation operators. The suggested approach in geometric multigrid literature is to design the restriction operations to downsample within strongly coupled nodes, and to avoid downsampling across loosely coupled ones [5]. This approach assumes that node connections can be labeled as strong or loose as a preprocessing step and that they vary smoothly over the computationally domain. In our segmentation problem, labeling loose connections corresponds to edge detection. These edges can not be labeled until we solve the equation, and we can not solve the equation until we label the edges.

In our approach, to ensure that edges in the image are maintained at coarser resolutions, we propose a novel way of restricting the edge weights of the L operator to the coarser levels. Figure 3.3(a) illustrates a 3×3 part of the 2D image at the fine grid resolution. At the coarser grid level, the nodes marked as A and B will remain, while the rest of the nodes will be eliminated. The connections between nodes A and B (restricted to the 3×3 part of the domain) can be seen as an electrical circuit. Each edge is associated with a conductance value. Then, the problem is reduced to computing the effective conductance between A and B which can be assigned as an edge weight in the coarse grid. We can first eliminate the corner nodes to obtain the circuit illustrated in Figure 3.3(b). The conductances for each of the four diagonal corner connections are computed from the series combination of two conductances in the original circuit. Then, using Δ-Y conversions from electrical circuit theory, we eliminate node 1 from the circuit in Figure 3.3(b) resulting in the circuit shown in Figure 3.3(c). The conductances for the new connections are computed according to the Δ-Y rule. Repearting the same procedure for the bottom part of the circuit, we obtain the circuit illustrated in Figure 3.3(d). Finally, an effective conductance value between nodes A and B can be computed using standard series and parallel circuit computations. The extension of this method to 3D will be discussed in Section 3.4.

3.3 Prolongation of the correction image

Corrections computed at coarser levels have to be upsampled to the finer levels, this operation is known as prolongation. Similar to the restriction of the residual, we consider two possibilities for

prolongation: nearest neighbor and interpolation. The nearest label approach is defined simply as

$$f^h[x,y] = f^{2h}\left[\lfloor\frac{x}{2}\rfloor, \lfloor\frac{y}{2}\rfloor\right]. \tag{3.7}$$

This can be thought of as the counterpart to injection restriction of the residual.

The interpolation approach uses one of the averaging kernels shown in Figure 3.4 depending on the position of the pixel on the finer grid. Pixels positioned horizontally between two pixels of the coarser grid are interpolated using the kernel in Figure 3.4(b). Similarly pixels located vertically between two coarser grid pixels use the kernel in Figure 3.4(c). For pixel at the center of 4 coarse grid pixels the larger kernel shown in Figure 3.4(a) is used.

3.4 Conductance restriction for 3D images

Unfortunately, the circuit reduction technique introduced in Section 3.2 does not extend to 3D circuits in a trivial manner. However, we can compute an approximate equivalent conductance by enumerating the paths between nodes A and B (that will be kept at the coarser level) on the fine grid and treating them as parallel pathways. Then, the conductance for each pathway is the serial combination of the included conductances, and the overall conductance between A and B is approximated as the parallel combination of each pathway. However, the number of non self crossing paths from A to B in a $3 \times 3 \times 3$ circuit is too large to allow for efficient computation of the effective conductance in the manner described above. Therefore, we restrict the possible paths to those including 7 pixels or less. These paths are enumerated ahead of time and stored as a list.

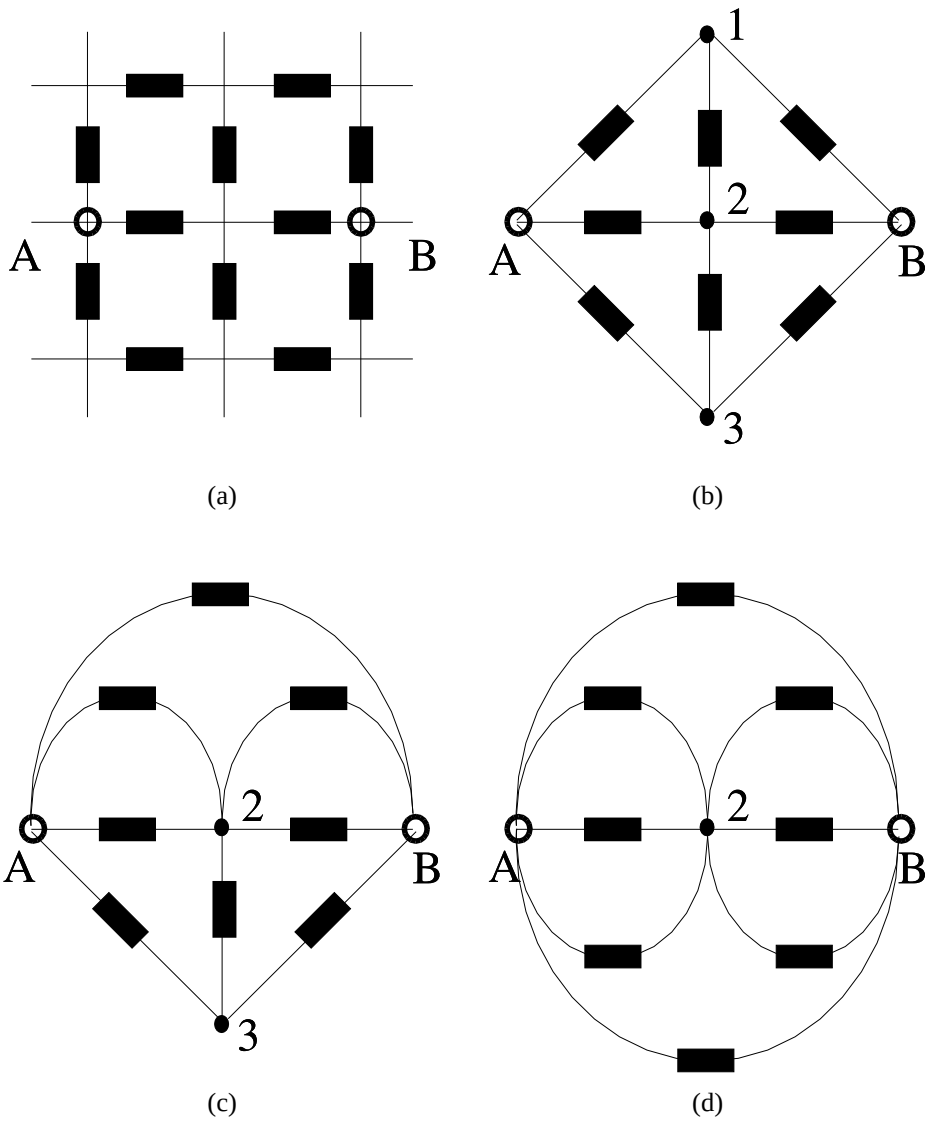


Figure 3.3: Circuit-reduction based conductance restriction.

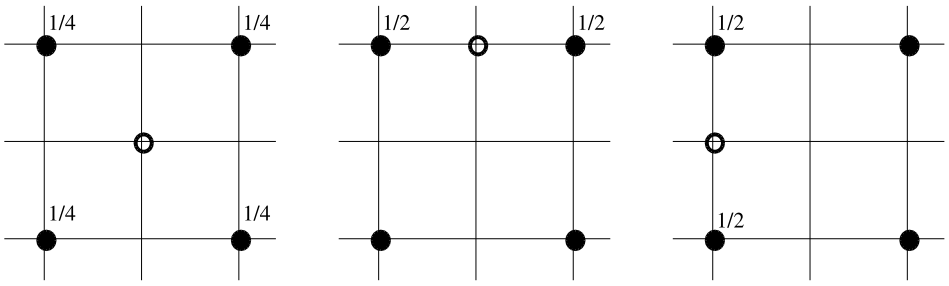


Figure 3.4: Prolongation kernels.

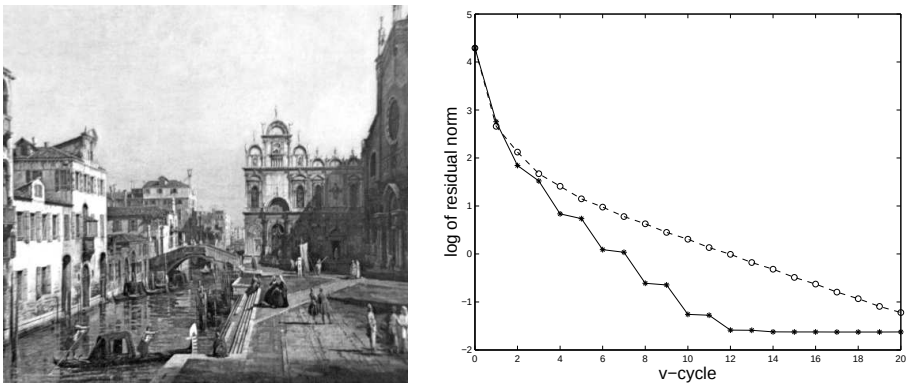
Chapter 4

Experiments

We begin this chapter by examining the convergence properties of our multigrid approach. Figure 4.1(a) shows a typical image used in image processing experiments. Lets assume that our goal is to obtain a cartoon-like image from this input image. This can be achieved by running Perona & Malik diffusion for a large amount of time and relatively small values of k in (2.2). Figure 4.1(b) plots the logarithm of the average quadratic norm of the residual given in (3.1) at the finest grid level against the number of *v-cycles* performed. We compare out proposed conductance based operator restriction to the more standard multigrid methods. The result of using injection restriction for the residual image, nearest neighbor prolongation for the correction image and the proposed conductance restriction for the anisotropic diffusion operator is shown with the solid line in Figure 4.1(b). The result of using the same residual restriction and correction prolongation, but a standard averaging restriction for the anisotropic diffusion operator is shown with the dashed line. We observe that the proposed approach requires fewer *v-cycles* to reach an acceptable level of residual. The vertical axis is the logarithm of the norm of the residual, and the original image had a range $[0, 255]$. Therefore, an acceptable level of residual can be taken as less than 0.1. This corresponds to the -1 level in the figure. Our approach requires about 8 *v-cycles* to reach this level, whereas the standard approach requires about 18. This corresponds to significant computational savings.

Next, we compare the multigrid and regular approaches to anisotropic diffusion. Figure 4.2 shows the computation time vs. Δt in (3.2). The regular, forward time stepping approach outlined in Section 2.2 requires a computational effort that scales linearly with Δt. On the other hand, the multigrid approach has a sub-linear scaling. Hence, the advantages of the multigrid approach become stronger as we increase Δt.

As Δt is increased to large values, the approximation to Perona & Malik diffusion given by (3.2) becomes less accurate. This is due to fixed conductance values used in the linearization (3.2). Let assume we'd like run Perona & Malik anisotropic diffusion for $\Delta t = 1000$. By choosing $\Delta t = 1000$ in (3.2) and obtaining the solution from a single multigrid step, we can obtain the largest computational savings. However, the accuracy of the solution will be poor. Instead, a tradeoff can be obtained by choosing a moderately large intermediate Δt value, say 100, which gives a good approximation to the original non-linear PDE, and repeating the multigrid solver 10 times



(a) (b)

Figure 4.1: Convergence of Multigrid approaches.

(recomputing the conductance values before each of the multigrid solvers).

In conclusion, we have introduced a new multigrid solver for Perona & Malik anisotropic diffusion that can provide significant computational savings. For future work, we plan to apply the solver to other PDEs with anisotropic coefficients.

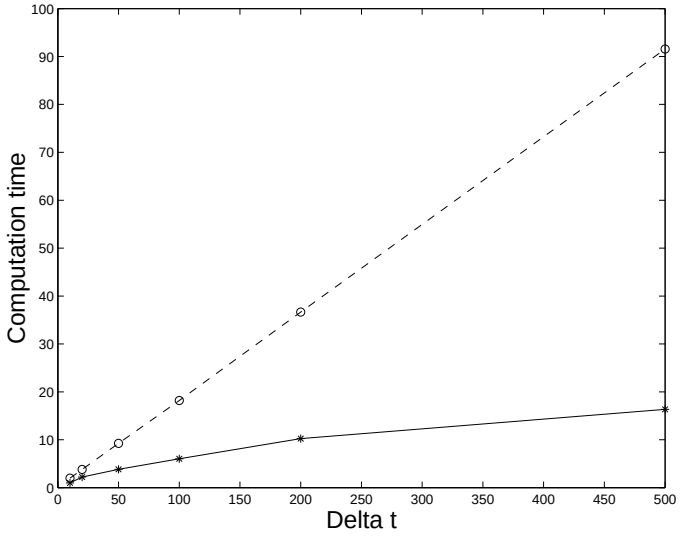


Figure 4.2: Computation time: Multigrid vs. Regular approaches.

Bibliography

[1] P. Perona and J. Malik, "Scale space and edge detection using anisotropic diffusion," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 629–639, July 1990.

[2] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Transactions on Image Processing*, vol. 7, March 1998.

[3] G. Sapiro, *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2001.

[4] G. Gerig, O. Kubler, R. Kikinis, and F. A. Jolesz, "Nonlinear anisotropic filtering of mri data," *IEEE Transactions on Medical Imaging*, vol. 11, pp. 221–232, 1992.

[5] U. Trottenberg, A. Schuller, and C. Oosterlee, *Multigrid*. Academic Press, 2000.

[6] D. Terzopoulos, "Image analysis using multigrid relaxation methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 129–139, 1986.

[7] P. Saint-Marc, J. Chen, and G. Medioni, "Adaptive smoothing: A general tool for early vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 514–529, 1991.

[8] S. T. Acton, "Multigrid anisotropic diffusion," *IEEE Transactions on Image Processing*, vol. 7, pp. 280–291, March 1998.

[9] M. Desbrun, M. Meyer, P. Schroder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *Proceedings of 26th annual conference on computer graphics and interactive techniques*, pp. 317–324, August 1999.