ELSEVIER
Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational and Applied Mathematics 193 (2006) 460–473
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
www.elsevier.com/locate/cam

Domain decomposition methods in image denoising using Gaussian curvature[☆]

D. Firsov, S.H. Lui*

Department of Mathematics, University of Manitoba, Winnipeg, Manitoba, Canada R3T 2N2

Received 12 February 2005

Abstract

A domain decomposition method is used to solve a Gaussian curvature driven flow to denoise digital images. The algorithm is embarrassingly parallel and is ideal for parallel computers. We use the scheme of Lee–Keun [Noise removal based on nonlinear diffusion with Gauss curvature conductance, 2003, Preprint] in which the diffusion coefficient contains a term involving the Gaussian curvature. This minimizes the occurrence of patches and tips of cones are not flattened.

MSC: 35Q80; 35R05; 65M

Keywords: Domain decomposition; Image restoration; Nonlinear PDEs; Gaussian curvature

1. Introduction

Perona–Malik [17], Osher–Rudin [16], and Rudin–Osher–Fatemi [19] introduced nonlinear partial differential equations (PDEs)) into the arsenal of tools for image processing. See Weickert [23], Sapiro [20] and Aubert–Kornprobst [2] for three recent monographs and Chan–Shen–Vese [7] for a review article on this fascinating topic. While nonlinear PDEs have been rather successful in denoising and enhancing digital images, they suffer from some serious drawbacks. First, they are not as fast as traditional filters (for example, median filters). Second, nonlinear diffusion PDEs typically destroy some features in images.

[☆] This work was supported in part by a Grant from NSERC of Canada.

* Corresponding author. Tel.: +1 204 474 9578; fax: +1 204 474 7611.
E-mail addresses: firsovd@cc.umanitoba.ca (D. Firsov), luish@cc.umanitoba.ca (S.H. Lui).

0377-0427/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2005.05.032

For instance, tips of cones are flattened. Third, there is no simple automatic procedure to determine the stopping time of the PDE. Typically, the user solves the PDE to some time, visually inspects the quality of the image before deciding to either stop or continue the process.

Progress on the second and third points was made by Lee–Keun [14] who proposed the use of Gaussian curvature in noise removal. Because cones have zero Gaussian curvature, diffusion-driven PDEs based on Gaussian curvature would not flatten the top of cones. Furthermore, objects with corners retain the corners under this flow, contrary to curvature flows which round off all corners. Lee–Keun also demonstrate numerically that degradation of the image over time is much slower than other flows, permitting a simple stopping criterion. Their paper is a significant contribution to the literature.

In this paper, we employ domain decomposition techniques to accelerate the solution of PDEs on parallel machines. We employ the Gaussian curvature scheme of [14] to denoise digital images and found that it is one of the best methods for image restoration.

The intensity of an image can be considered as the surface $z = u(x, y)$ in $\mathbf{R}^3$ where u is a non-negative function defined on a rectangle. In practice, u is defined only at discrete points (pixels) on the rectangle and takes on discrete values, say integers between 0 and 255. A digital image can inherit noise due to turbulence in the atmosphere, interference in the channel used for transmission, small imperfections in the equipment, etc. The goal of image restoration is to automatically clean up the image. This is a very difficult problem due to the endless variety of images, from smooth textures to artificial objects with edges.

The first PDE method used for image denoising was to evolve the noisy image by the (linear) heat equation [13] with homogeneous Neumann boundary conditions. Since noise usually has high frequencies, the heat equation is quite suitable for damping high-frequency signals. Unfortunately, edges are also characterized by the presence of high-frequency terms due to discontinuities in the intensity function. The application of the heat equation to such images results in noticeable blurring near edges.

The heat equation is a diffusion equation in which diffusion is isotropic, the same in all direction. One breakthrough by Perona–Malik [17] was to diffuse only in direction parallel to edges—the so-called anisotropic diffusion. The simplest form of their nonlinear PDE is

$$u_t = \nabla \cdot \left(\frac{\nabla u}{C + |\nabla u|^2} \right) = \frac{1}{C + |\nabla u|^2} \left(u_{\mathrm{TT}} + \frac{C - |\nabla u|^2}{C + |\nabla u|^2} u_{\mathrm{NN}} \right),$$

where C is a constant, $|v|$ denotes the 2-norm of the vector v and u_{TT}, u_{NN} stand for the second tangential and normal derivatives, respectively. In regions where the image is smooth, $|\nabla u|$ is small and above equation is similar to the linear heat equation. However, at an edge, $|\nabla u|$ is large and anti-diffusion occurs in direction perpendicular to the edge. The effect is that the edge is sharpened.

The papers [4,1] made a significant improvement of the Perona–Malik scheme. They proposed the PDE

$$u_t = \nabla \cdot \left(\frac{\nabla u}{C + |\nabla G_\sigma * u|^2} \right),$$

(as well as much more general ones) where G_σ is the Gaussian kernel defined by

$$G_\sigma = c\sigma^{-1} \mathrm{e}^{-(x^2+y^2)/4\sigma}$$

for a normalization constant c and $*$ denotes convolution. The incorporation of this smoothing process means that noise will no longer be amplified as in the original PDE and that a rigorous mathematical theory of existence and uniqueness result can be obtained by the viscosity solution method [8]. In [1], they also proposed another nonlinear PDE for denoising:

$$u_t = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) |\nabla u| = u_{\mathrm{TT}}. \tag{1.1}$$

This has the interpretation that each level set $u=$ constant evolves in its normal direction at speed equal to its curvature $\nabla \cdot (\nabla u / |\nabla u|)$ as well as diffuses only in the tangential direction. This is known as curvature flow. Earlier, Gage and Grayson [10,12] proved rigorously that, roughly speaking, arbitrary closed curves in the plane become convex in finite time and then shrink to a point. This was pure mathematics which found applications only a few years later. It gives yet another view why curvature flow is effective in denoising images. Noise can be modelled by sharp spikes in the surface $z = u(x, y, t)$. Each level set of the spike has a very small cross-section and thus it quickly shrinks to a point under curvature flow. One drawback of curvature flow is that noise as well as all features in the image disappear over time. Another unpleasant by-product is that corners are rounded off.

At around the same time as the Perona–Malik paper, Osher–Rudin [16] used ideas from nonlinear hyperbolic PDEs to enhance images (sharpen edges). They treated edges as shocks. One form of their PDE is

$$u_t = -|\nabla u| \operatorname{sign} u_{\mathrm{NN}}.$$

That this PDE sharpens edges is apparent by examining the one-dimensional version of the above PDE.

Soon afterward, Osher–Rudin–Fatemi came up with the idea of minimizing the L^1 norm of the gradient of the image, known as the BV norm. This penalizes noise without blurring edges. One simple form of their PDE is

$$u_t = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right). \tag{1.2}$$

In most of the diffusion-based PDEs above, there is serious loss of information over time. Lee–Keun [14] recently derived a scheme based on Gaussian curvature K:

$$u_t = \nabla \cdot (|K| \nabla u), \quad K = \frac{\det D^2 u}{(1 + |\nabla u|^2)^2} = \frac{u_{xx} u_{yy} - u_{xy}^2}{(1 + |\nabla u|^2)^2}. \tag{1.3}$$

Here, $D^2 u$ is the 2×2 Hessian matrix of second derivatives. The beauty of this scheme is that cones have a zero Gaussian curvature and thus remain unchanged by the PDE. (Strictly speaking, the Gaussian curvature at the tip of the cone is infinite but this is not relevant in a discrete setting.) Furthermore, they find that the decay of image over time is much slower than other schemes and this allows a simple criterion to stop the process. Note that the above form of the PDE can be difficult to implement because of the second derivative terms inside the divergence term. Lee–Keun recommended some modifications:

$$u_t = |K| \Delta u \quad \text{or} \quad u_t = |K| \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right). \tag{1.4}$$

Another variation is

$$u_t = |K| \nabla \cdot \left(\frac{\nabla u}{C + |\nabla u|^2} \right), \tag{1.5}$$

where C is some positive constant. This scheme both denoises and enhances the image. However, in our numerical experiments, (1.3) yields the best results.

In the next section, we discuss how domain decomposition methods can be used. In Section 3, we describe in detail our finite difference scheme. Because of the presence of discontinuities, simple second order finite difference scheme do not yield ideal results. Numerical results are given in Section 4.

2. Domain decomposition algorithm

Domain decomposition methods have become essential tools in modern large-scale computation. One version, known as the Schwarz alternating method, subdivides the domain of the PDE into several overlapping subdomains. Independent processors can work on the subdomain problems with an exchange of subdomain boundary information at the end of each iteration. See the monographs [21,18] and review article [6] on this method and references.

The original Schwarz method was invented more than 130 years ago. It has become quite a popular method over the past 20 years because of parallel computers. To explain a version of this method, known as the additive Schwarz method, consider the Poisson equation $-\Delta u = f$ on some domain Ω with homogeneous Dirichlet boundary conditions. Suppose that Ω is the union of two overlapping subdomains Ω_1 and Ω_2. Starting with any initial guess $u^{(0)}$, define the following sequence:

$$u^{n+1} = u^n + \omega(u_1^{n+1} - u^n) + \omega(u_2^{n+1} - u^n),$$

where ω is some positive constant and

$$-\Delta u_i^{n+1} = f \quad \text{on } \Omega_i, \qquad u_i^{n+1} = u^n \quad \text{on } \partial\Omega_i. \tag{2.1}$$

Thus, the new iterate is the old iterate plus a correction from each subdomain. Note that the subdomain PDEs can be solved in parallel.

It can be shown that when $0 < \omega < \frac{1}{2}$, the sequence of iterates converges to the exact solution. In practice, the additive Schwarz method is used as a preconditioner for the conjugate gradient (CG) method rather than as the solver. Under certain conditions, it can be shown that the rate of convergence of this preconditioned CG method is independent of the discretization parameter.

There are two approaches for time-dependent PDEs. If an implicit method is used, then each time iteration requires the solution of a linear elliptic PDE. The CG method with additive Schwarz preconditioner can be used [3]. See also [22]. One drawback of this technique is that there is communication cost involved with exchanging boundary data and that all subdomain problems must be synchronized at each time iteration. The other alternative is known as waveform relaxation (see [11,15]). Here, we solve the time-dependent subdomain problems independently until the final time T. Then boundary data for all intermediate times are exchanged. We integrate again from $t = 0$ to T with the new boundary data and repeat the integration until some tolerance criterion has been satisfied. The advantage here is that communication costs are minimized. However, the performance of this algorithm clearly deteriorates as T increases.

We now define more precisely the first iteration of the waveform relaxation for the time-dependent PDE $u_t = f(u)$ on the spatial domain $(0, 1)$ with boundary conditions $u = 0$ at $x = 0,\ 1$ and initial condition $u = u_0$ at $t = 0$. Suppose the spatial domain is subdivided into two overlapping subdomains $(0, b),\ (a, 1)$

where $a<b$. (Generalization to an arbitrary number of subdomains is straightforward.) The algorithm generates two subdomain sequences as follows:

$$\begin{aligned}
&\frac{\partial u_1^{n+1}}{\partial t}=f(u_1^{n+1}), \quad x\in(0,b),\\
&u_1^{n+1}(t=0)=u_0, \quad x\in(0,b),\\
&u_1^{n+1}(x=0)=0, \quad t>0,\\
&u_1^{n+1}(x=b)=u_2^n(x=b), \quad t>0,
\end{aligned}$$

and

$$\begin{aligned}
&\frac{\partial u_2^{n+1}}{\partial t}=f(u_2^{n+1}), \quad x\in(a,1),\\
&u_2^{n+1}(t=0)=u_0, \quad x\in(a,1),\\
&u_2^{n+1}(x=1)=0, \quad t>0,\\
&u_2^{n+1}(x=a)=u_1^n(x=a), \quad t>0.
\end{aligned}$$

We define $u_1^0=u_2^0=u_0$ at $x=a,b$ for $t>0$. Throughout the time integration, the subdomain problems can be solved in parallel without communication.

For image processing, T, the final time of integration, is typically small. Furthermore, noise is a local feature and thus information exchange between adjacent subdomains is not crucial. Hence the second approach described above seems more suitable for image processing. In our algorithm, we split up the domain into $m\geqslant 2$ overlapping vertical strips. We found that it is sufficient to carry out *one* iteration of the waveform relaxation. The final solution is pieced together from the subdomain solutions. In the overlapping regions, we take a convex combination. For instance, in the two-subdomain case with subdomains $[0,b]\times[0,1]$ and $[a,1]\times[0,1]$ where $a<b$. Call the subdomain solutions (after one pass of the waveform relaxation) u_1 and u_2, respectively. Then define

$$u(x,y)=\begin{cases} u_1(x,y), & 0\leqslant x\leqslant a,\\ \left(1-\dfrac{x-a}{b-a}\right)u_1(x,y)+\dfrac{x-a}{b-a}u_2(x,y), & a\leqslant x\leqslant b,\\ u_2(x,y), & b\leqslant x\leqslant 1,\end{cases}$$

as the final solution.

In practice we padded each subdomain with extra data. For instance, for the subdomain one above, we actually took the width to be $[0,b+\delta]$ for some positive δ. This was done to decrease any artificial effects along the artificial boundary, for instance, an edge occurring along $x=b$.

In our numerical experiments, we found that the domain decomposition algorithm with Dirichlet data ($u_1^{n+1}(b)=u_0(b)$ and $u_2^{n+1}(a)=u_0(a)$) is slightly inferior to that with Neumann data ($(u_1^{n+1})'(b)=u_0'(b)$ and $(u_2^{n+1})'(a)=u_0'(a)$). Hence, we use Neumann boundary conditions in our numerical results.

We remark that the above domain decomposition algorithm is applicable to any diffusion scheme and is *embarrassingly parallel*. Since only one waveform relaxation iteration is performed, there is actually no communication cost involved. This algorithm thus should attain near perfect speedup and is scalable.

Note that PDEs are good schemes for denoising additive Gaussian noise but are not suitable for salt-and-pepper noise. For the latter, median filters provide excellent denoising capability. See, for instance, [5].

3. Finite difference schemes

We use an implicit finite difference scheme for the PDE

$$u_t - |K(G_\sigma * u)|^\alpha \nabla \cdot (\mu(G_\sigma * u)\nabla u) = 0$$

with homogeneous Neumann boundary conditions. Here,

$$K(w) = \frac{D^2 w}{1 + |\nabla w|^2}, \quad \mu(w) = \frac{\gamma}{C + |\nabla w|^2} + (1-\gamma)|K(w)|.$$

When $\alpha = 0$, $\gamma = 1$, we recover the Perona–Malik scheme while if $\alpha = 0$ and $\gamma = 0$, we obtain (1.3).

Let $A(w)$ denote a discrete approximation of $|K(w)|^\alpha \nabla \cdot (\mu(w)\nabla u)$. The scheme is a Crank–Nicholson scheme

$$u^{n+1} - \frac{\tau}{2} A(G_\sigma * u^n) u^{n+1} = u^n + \frac{\tau}{2} A(G_\sigma * u^n) u^n.$$

Here, τ is the time step and u^n denotes the discrete solution at time step n.

More specifically, for internal grid point (i_1, i_2),

$$|K(w)|^\alpha_{i_1,i_2} [\nabla \cdot (\mu(w)\nabla u)]_{i_1,i_2} = \sum_{j=-1}^{1} \sum_{k=-1}^{1} a^{j,k}_{i_1,i_2} u_{i_1+j,i_2+k},$$

where

$$a^{j,k}_{i_1,i_2} = \frac{\lambda_{|j|+|k|}}{h^2 \sqrt{|j|+|k|}} |K(w)|^\alpha_{i_1,i_2} \; \mu(w)^{j,k}_{i_1,i_2}$$

if not both j and k are zero and

$$a^{0,0}_{i_1,i_2} = \sum_{j,k=-1;\, j,k \neq 0}^{1} \frac{a^{j,k}_{i_1,i_2}}{h^2},$$

$\lambda_1 = \sqrt{2} - 1, \;\; \lambda_2 = 1 - \lambda_1,$

$$\mu(w)^{j,k}_{i_1,i_2} = \begin{cases} \frac{1}{2}|K(w)_{i_1,i_2} + K(w)_{i_1+j,i_2+k}|, & \gamma = 0, \\ (C + h^{-2}\mathrm{Grad}^2(w)^{j,k}_{i_1,i_2})^{-1}, & \gamma = 1, \end{cases}$$

and

$$\mathrm{Grad}^2(w)^{j,k}_{i_1,i_2} = \begin{cases} \frac{1}{2}(w_{i_1+j,i_2+k} - w_{i_1,i_2})^2 + \frac{1}{2}(w_{i_1+j,i_2} - w_{i_1,i_2+k})^2, & j,k \neq 0, \\ (w_{i_1+j,i_2} - w_{i_1,i_2})^2 & \\ \quad +(w_{i_1+j,i_2+1} - w_{i_1+j,i_2-1} + w_{i_1,i_2+1} - w_{i_1,i_2-1})^2/16, & k = 0, \\ (w_{i_1,i_2+k} - w_{i_1,i_2})^2 & \\ +(w_{i_1+1,i_2+k} - w_{i_1-1,i_2+k} + w_{i_1+1,i_2} - w_{i_1+1,i_2})^2/16, & j = 0. \end{cases}$$

Note that the divergence term is calculated by a rotationally invariant scheme. See [2], for instance, for an introduction to rotationally invariant finite difference schemes. In the approximation of μ, we take an average of the Gaussian curvature at a centre point and its neighbour. Another possibility is to take the max of the value at the centre point and at a neighbouring point. This alternate scheme gives slightly better results for complex images but is inferior for simple images (small oscillations are observed at flat places).

For the approximation of K,

$$K(w)_{i_1,i_2} = \left.\frac{\operatorname{maxmod}(D_1^2 w,\, D_2^2 w)}{h^4(1+\max(\operatorname{Grad}_1^2 w,\, \operatorname{Grad}_2^2 w)h^{-2})^2}\right|_{i_1,i_2},$$

where

$$\operatorname{maxmod}(a,b) = \begin{cases} a & \text{if } |a| \geqslant |b|, \\ b & \text{otherwise} \end{cases}$$

and

$$\begin{aligned}(D_1^2 w)_{i_1,i_2} &= (w_{i_1+1,i_2} + w_{i_1-1,i_2} - 2w_{i_1,i_2})(w_{i_1,i_2+1} + w_{i_1,i_2-1} - 2w_{i_1,i_2}) \\ &\quad - \frac{(w_{i_1+1,i_2+1} + w_{i_1-1,i_2-1} - w_{i_1-1,i_2+1} - w_{i_1+1,i_2-1})^2}{16},\end{aligned}$$

$$\begin{aligned}(D_2^2 w)_{i_1,i_2} &= \frac{(w_{i_1+1,i_2+1} + w_{i_1-1,i_2-1} - 2w_{i_1,i_2})(w_{i_1-1,i_2+1} + w_{i_1+1,i_2-1} - 2w_{i_1,i_2})}{4} \\ &\quad - \frac{(w_{i_1+1,i_2} + w_{i_1-1,i_2} - w_{i_1,i_2+1} - w_{i_1,i_2-1})^2}{4},\end{aligned}$$

$$\begin{aligned}(\operatorname{Grad}_1^2 w)_{i_1,i_2} = {} & \max((w_{i_1+1,i_2} - w_{i_1,i_2})^2,\ (w_{i_1-1,i_2} - w_{i_1,i_2})^2) \\ & + \max((w_{i_1,i_2+1} - w_{i_1,i_2})^2,\ (w_{i_1,i_2-1} - w_{i_1,i_2})^2),\end{aligned}$$

$$\begin{aligned}&(\operatorname{Grad}_2^2 w)_{i_1,i_2} \\ &= \frac{\max((w_{i_1+1,i_2+1}-w_{i_1,i_2})^2(w_{i_1-1,i_2-1}-w_{i_1,i_2})^2)+\max(w_{i_1-1,i_2+1}-w_{i_1,i_2})^2,\, (w_{i_1+1,i_2-1}-w_{i_1,i_2})^2)}{2}.\end{aligned}$$

Note that $D_1^2 w$ is the usual second-order finite difference approximation of $h^4 \ \det\ D^2 w$, while $D_2^2 w$ is a second-order finite difference approximation on a grid that is rotated by 45°. By taking the maxmod of these two approximations, we obtain a more representative estimation of noise and edges. Experimentally, we find that taking the maxmod of these two determinants yield higher quality images. Also $\operatorname{Grad}_1^2 w$ is a combination of one-sided difference approximation of $h^2|\nabla w|^2$, while $\operatorname{Grad}_2^2 w$ achieves a different approximation on a grid rotated by 45°. Along an edge, it is possible for a one-sided difference scheme to fail to detect it if the finite difference happens to be taken in the direction along the edge. By taking one-sided difference along a rotated grid as well, the scheme will detect the edge and return a large value of the gradient. We have tried other difference schemes including a second-order ENO scheme resulting in slightly inferior results.

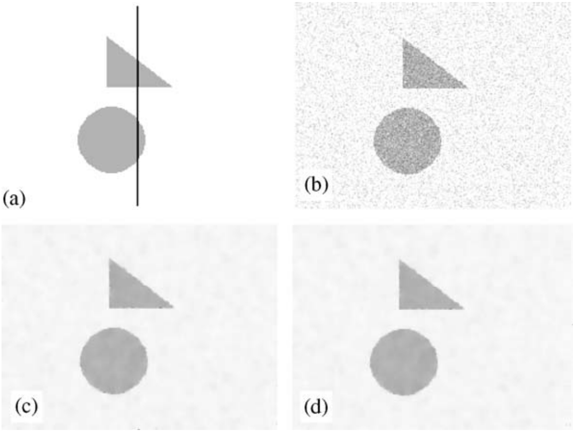

Fig. 1. (a) Original picture, (b) noisy picture, (c) denoised on 1 domain, (d) denoised on 2 domains.

As result of the discretization, we have a system of linear equations with a 9-diagonal M-matrix. The stopping criterion for the scheme is

$$\frac{|u^n - u^{n-1}|}{|u^n|} < \varepsilon.$$

As we shall see in the next section, the Gaussian curvature (1.3) yields rather impressive results. The parameters in the algorithm are the space and time steps, and the value of ε in the stopping criterion. The quality of the picture is not sensitive to the value of ε when the noise is small. In the presence of large noise, for the best quality, some experimentation in selecting the value of ε is essential. For the Perona–Malik scheme, termination requires human intervention regardless of the noise level.

4. Numerical results

All numerical simulations involving the Gaussian curvature scheme are run using the waveform relaxation domain decomposition algorithm with four overlapping subdomains. The width of each overlapping region is 10 pixels with an additional buffer region of 5 pixels.

We also compare our scheme with the version of Perona–Malik scheme described in [4]. For brevity, we still refer to this as the Perona–Malik scheme. For both schemes, we use $h=10^{-2}$, $\tau=10^{-3}$, $\sigma=0.5$. We choose $C=1$ in the Perona–Malik scheme. The intensity function u is normalized to be between zero and one.

We first test the performance of the Gaussian curvature algorithm when there is an edge in the overlapping region. See Fig. 1. The line in the first image (a) depicts the overlapping region. The noisy image has an signal-to-noise ratio (SNR) of 25.268 while the SNR values of the denoised image using one domain and two subdomains are 30.572 and 30.570, respectively. This shows that domain decomposition improves computational speed (on a parallel machine) without any detrimental effect on the quality of the image, even in case an edge crosses the subdomain boundaries.

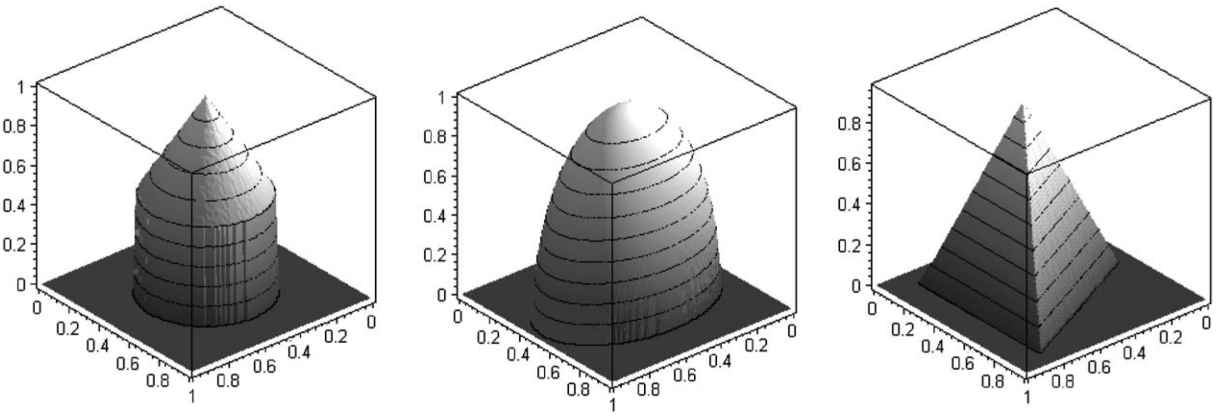

Fig. 2. Clean cone, ellipsoid and pyramid.

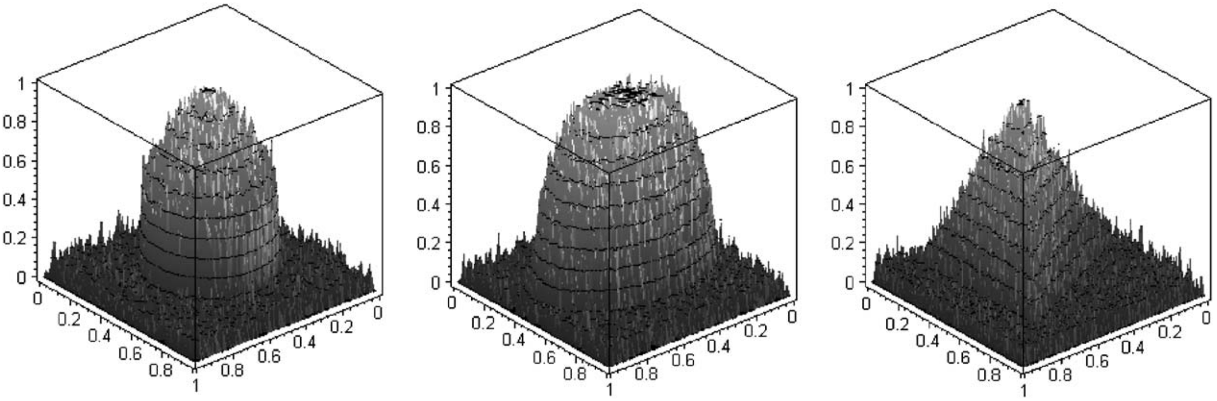

Fig. 3. Noisy cone, ellipsoid and pyramid.

Next we compare the Gaussian curvature (GC) (1.3) with the Perona–Malik (PM) scheme on some simple geometric objects. The original picture is shown in Fig. 2 while the noisy one is shown in Fig. 3. The denoised images are shown in Figs. 4 and 5. The GC scheme nicely recovers the original objects while the PM scheme inflicts some damage to the shapes. The stopping criterion for the GC scheme is $\varepsilon = 10^{-5}$ while by visual inspection, four PM iterations appeared the best.

We also used the GC algorithm but employing central differences instead of our more elaborate difference scheme to illustrate the effects of not using appropriate finite differences. See Fig. 6. Observe the presence of waves on the surfaces of the solids as well as on the planes.

Next, consider a complex image containing both edges and texture in Fig. 7 with and without noise and the denoised pictures in Fig. 8. For the GC scheme, the stopping criterion was $\varepsilon = 5 \times 10^{-3}$, resulting in 21 iterations while for the PM scheme, 10 iterations appeared the best.

In Fig. 9, we zoom in to parts of the picture. It can be seen that the GC scheme gives better details on the wall paper. At other parts of the image, however, it is difficult to determine which scheme is better.

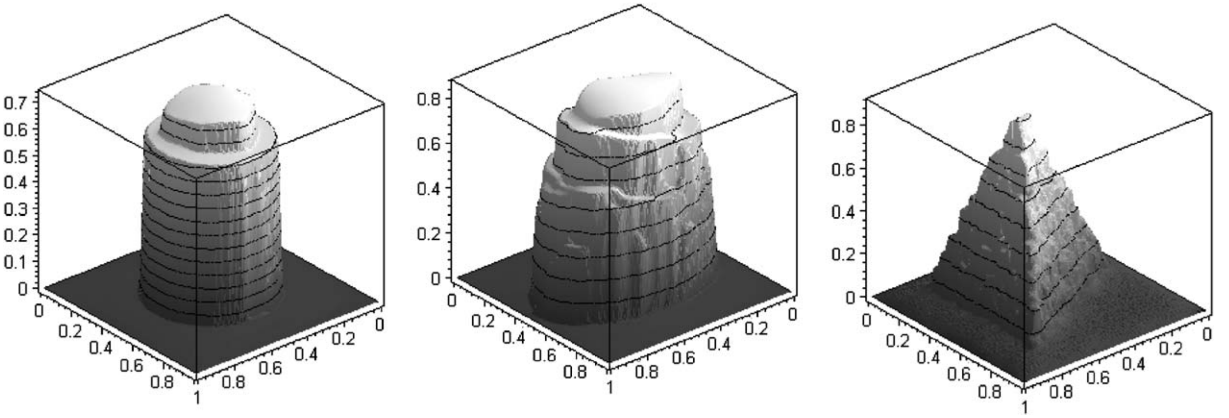

Fig. 4. Denoised by Perona–Malik scheme.

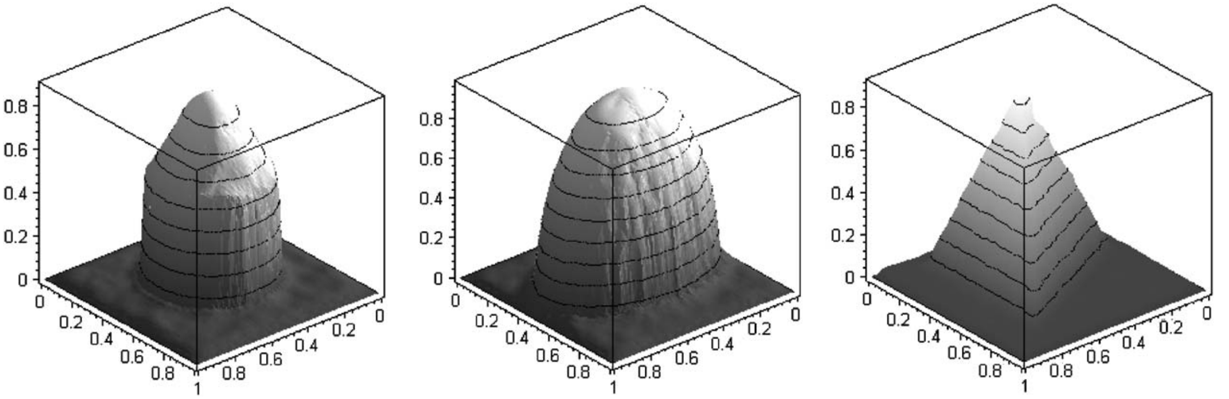

Fig. 5. Denoised by Gaussian curvature scheme.

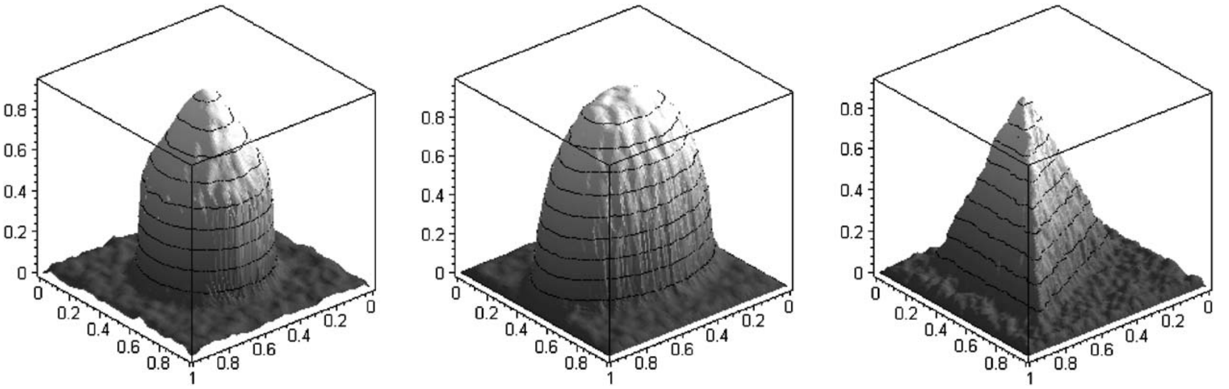

Fig. 6. Denoised by Gaussian curvature scheme using central differences.

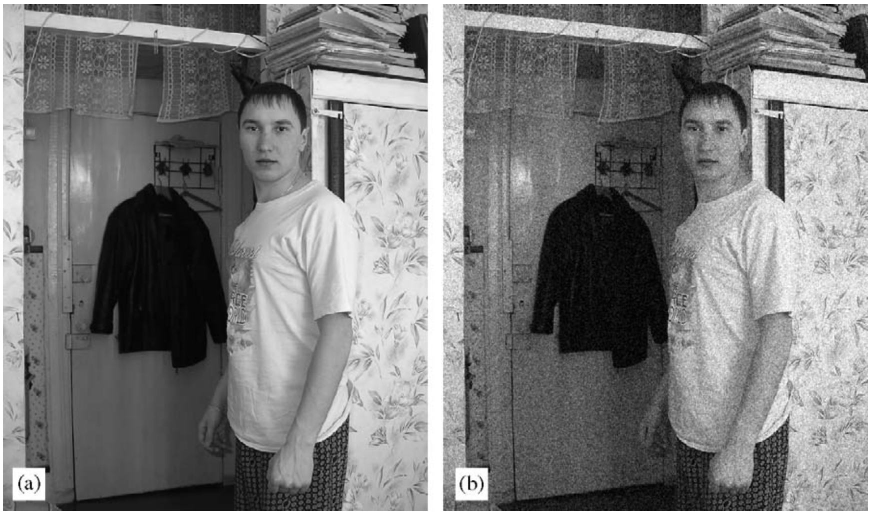

Fig. 7. Clean and noisy images.

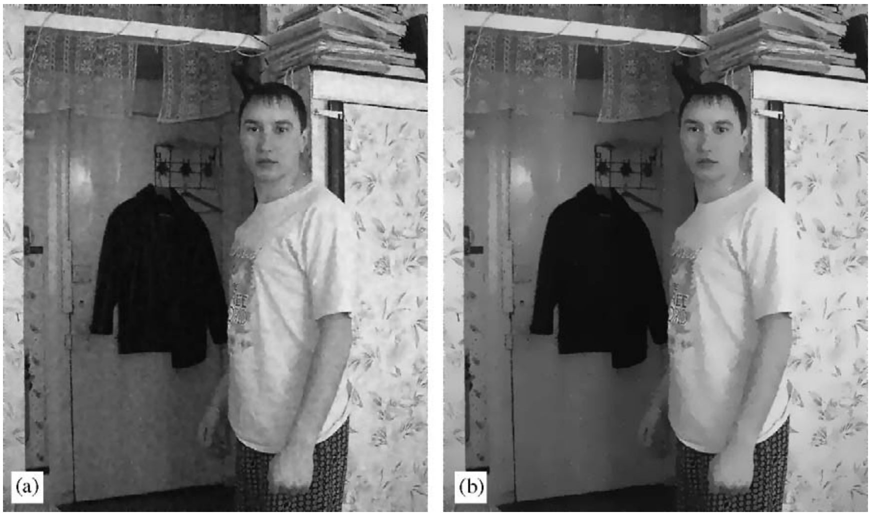

Fig. 8. Picture denoised by Gaussian curvature and Perona–Malik schemes.

We plot the SNR and the relative change ΔU as functions of the iteration number in Fig. 10. Here

$$\mathrm{SNR} = 10\log_{10}\frac{|u^n|}{|u^n - u^*|}, \quad \Delta U = \frac{|u^n - u^{n-1}|}{|u^n|}, \tag{4.1}$$

where u^* is the uncorrupted image. It can be observed that when the noise is small (variance of 0.001), then the quality of GC image is not sensitive to the stopping criterion. However, when the noise is large (variance of 0.01), the result is highly dependent on the stopping criterion, though much less so than for PM. Thus, our choice of final stopping time T depends on the noise level. Larger level of noise requires

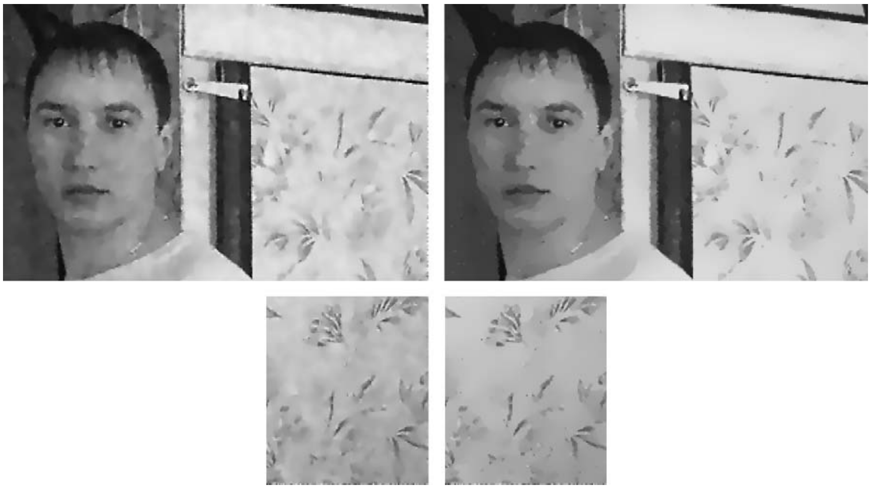

Fig. 9. Pictures enlarged near the wall. Left images are denoised by Gaussian curvature scheme while the images on the right are by Perona–Malik.

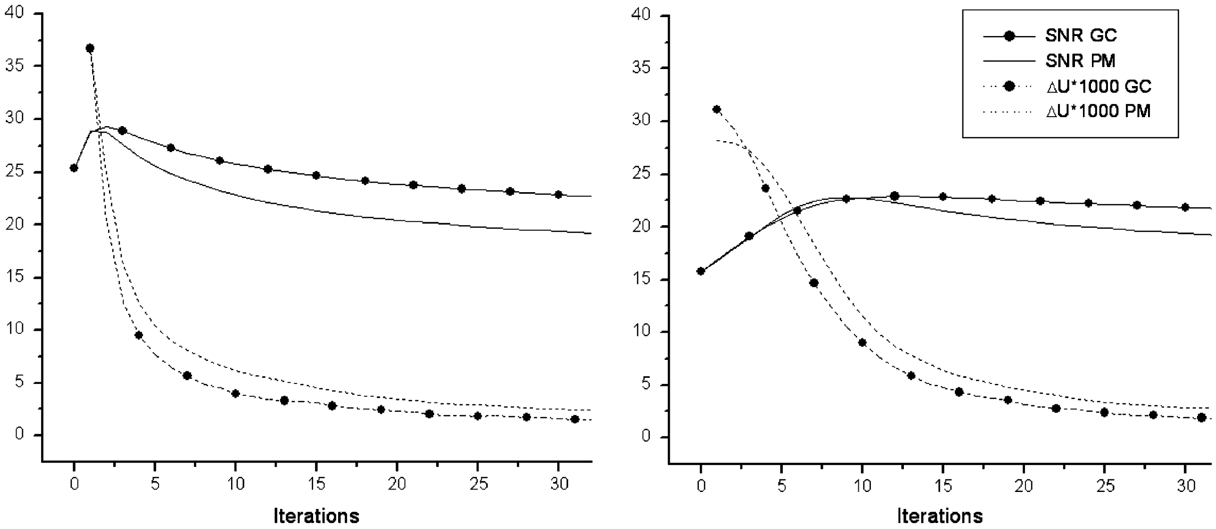

Fig. 10. Comparison of some statistics for the Gaussian curvature (GC) scheme and the Perona–Malik (PM) scheme. Left picture is for an initial image with small noise (variance of 10^{-3}) while the right one is for one with large noise (variance of 10^{-2}). See (4.1) for the definition of the relative difference ΔU.

a larger value of T but the quality of the image does not degrade quickly as a function of (larger than necessary values of) T for GC.

We note that the SNR difference between the domain decomposition algorithm (using up to 10 subdomains) and the algorithm without domain decomposition is less than 10^{-3} for the GC scheme.

For simple geometric objects, it seems that the GC algorithm (1.3) does a much better job of denoising than the PM scheme. For complex images, it is much more difficult to compare the algorithms. However, the GC scheme does seem to be slightly better at certain places and it attains a higher value of SNR. The (1.3) appears to be superior to the other GC (1.4) and (1.5). We have also tried the curvature flow (1.1) and TVD (1.2) and both give inferior results to the Gaussian curvature scheme. We do not display these other results due to space limitation.

In conclusion, we have proposed a simple domain decomposition algorithm based on the waveform relaxation of nonlinear diffusion equations for denoising images. The algorithm is embarrassingly parallel. The Gaussian curvature-driven scheme is found to be one of the best PDE methods for image restoration. It is found to be less sensitive to deviation in the final time of integration than other schemes. The time complexity of the algorithm is linear in the number of pixels. We note that there is another approach to image denoising which is based on wavelet transform followed by thresholding [9] which also has linear time complexity.

Acknowledgements

We thank an anonymous referee for several useful comments.

References

[1] L. Alvarez, P.L. Lions, J.M. Morel, Image selective smoothing and edge detection by nonlinear diffusion. ii, SIAM J. Numer. Anal. 29 (1992) 845–866.

[2] G. Aubert, P. Kornprobst, Mathematical Problems in Image Processing, Springer, Berlin, 2002.

[3] X.C. Cai, Additive Schwarz algorithms for parabolic convection–diffusion equations, Numer. Math. 60 (1991) 41–61.

[4] F. Catte, P.L. Lions, J.M. Morel, T. Coll, Image selective smoothing and edge detection by nonlinear diffusion, SIAM J. Numer. Anal. 29 (1992) 182–193.

[5] R.H. Chan, C.W. Ho, M. Nikolova, Impulse noise removal by median-type noise detector and edge-preserving regularization, 2003, preprint.

[6] T.F. Chan, T.P. Mathew, Domain decomposition algorithms, Acta Numerica (1994) 61–143.

[7] T.F. Chan, J. Shen, L. Vese, Variational pde models in image processing, Notices AMS 50 (2003) 14–26.

[8] M.G. Crandall, H. Ishii, P.L. Lions, User's guide to viscosity solutions of second order partial differential equations, Bull. AMS 27 (1992) 1–67.

[9] D. Donoho, De-noising by soft thresholding, IEEE Trans. Inform. Theory 41 (1995) 613–627.

[10] M. Gage, Curve shortening makes convex curves circular, Invent. Math. 76 (1984) 357–364.

[11] M.J. Gander, A.M. Stuart, Space-time continuous analysis of waveform relaxation for the heat equation, SIAM J. Sci. Comput. 19 (1998) 2014–2031.

[12] M. Grayson, The heat equation shrinks embedded plane curves to round points, J. Differential Geom. 26 (1987) 285–314.

[13] J.J. Koenderink, The structure of images, Biol. Cybernetics 50 (1984) 363–370.

[14] S.H. Lee, S.J. Keun, Noise removal based on nonlinear diffusion with Gauss curvature conductance, 2003, Preprint.

[15] S.H. Lui, On monotone iteration and Schwarz methods for nonlinear parabolic PDEs, J. Comput. Appl. Math. 130 (2003) 309–321.

[16] S. Osher, L. Rudin, Feature-oriented image enhancement using shock filters, SIAM J. Numer. Anal. 27 (1990) 919–940.

[17] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Trans. Pattern Anal. Mach. Intell. 12 (1990) 629–639.

[18] A. Quarteroni, A. Valli, Domain Decomposition Methods for Partial Differential Equations, Oxford University Press, Oxford, 1999.

[19] L. Rudin, S. Osher, D. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D 60 (1992) 259–268.

[20] G. Sapiro, Geometeric Partial Differential Equations and Image Analysis, Cambridge University Press, Cambridge, 2001.

[21] B.F. Smith, P. Bjorstad, W.D. Gropp, Domain Decomposition: Parallel Multilevel Algorithms for Elliptic Partial Differential Equations, Cambridge University Press, New York, 1996.

[22] X.C. Tai, A subspace decomposition method for parabolic problems, Numer. Methods Partial Differential Equations 14 (1998) 27–46.

[23] J. Weickert, Anisotropic Diffusion in Image Processing, Teubner, Stuttgart, 1998.