On multigrids for solving a class of improved total variation based PDE models*

Joseph Savage† and **Ke Chen**‡

Abstract

Total variation regularization is well established as a denoising procedure with excellent edge capturing properties, however images denoised using total variation regularization suffer from the staircasing effect. Many models to reduce this effect have been proposed in the literature but not all models can be solved effectively. Our interest is in the fast iterative solution of the nonlinear partial differential equations arising from these models, specifically the use of nonlinear multigrid methods.

In this paper we first survey a class of staircasing reduction models and then focus on using effective solution as a criterion to find the most suitable model in this class of models that maintains edges by compromising in some way between Total Variation and H^1 regularization. Our experimental results compare the performance of nonlinear multigrid solvers, the fixed point iteration method using preconditioned conjugate gradient inner solvers and the explicit time marching (gradient descent) approaches.

AMS subject class: 68U10, 65F10, 65K10.

Keywords: Image restoration, denoising, regularization, nonlinear solvers, multilevel methods, staircasing reduction.

Contents

*To appear: Proc. 1st Int. Conf. PDE-Based Image Processing and Related Inverse Problems, eds. T. F. Chan, X.-C. Tai, K. A. Lie and S. Osher, 2006, Springer-Verlag.

†Support of the UK EPSRC DTA grant is gratefully received.

‡Department of Mathematical Sciences, The University of Liverpool, Peach Street, Liverpool L69 7ZL, UK. E-mail: k.chen@liverpool.ac.uk, URL: http://www.liv.ac.uk/∼cmchenke (for correspondence).

1 Introduction

During recording and transmission an image will often become contaminated with random Gaussian type noise; this is modeled by the equation

$$z(x,y) = u(x,y) + n(x,y), \qquad x,y \in \Omega$$

where Ω is a bounded and open domain of $\mathbb{R}^2$ (usually a rectangle). Here z is a real function representing the observed (known) image, which in practice will be a discrete quantity (given in the form of $n \times m$ pixel values), u is the true image (unknown) and n is an additive (unknown) noise term. The problem of recovering u from z is an ill-conditioned inverse problem.

Image denoising methods use regularization techniques based on a priori knowledge of the image properties to approximate u. An early approach was H^1 regularization given by the following minimization problem:

$$\min_u J_{H^1}(u), \qquad J_{H^1}(u) = \int_\Omega \alpha|\nabla u|^2 + \frac{1}{2}(u-z)^2 dxdy$$

The convex functional $J_{H^1}(u)$ is made up of a regularization functional $\int |\nabla u|^2$,which penalizes against non-smooth images, and a fit to data functional $\int \frac{1}{2}(u-z)^2$, balanced by a regularization parameter α. The resulting Euler-Lagrange equation for this problem is:

$$-\alpha\Delta u + u = z$$

with homogenous Neumann boundary condition $\frac{\partial u}{\partial n} = 0$, which can be solved efficiently using, for example, a multigrid method (see [8]). The problem with this approach is that although smooth regions in the image are recovered well, edges present in the original image are blurred in the reconstruction.

To overcome the poor edge-capturing properties of H^1 regularization, Rudin, Osher and Fatemi (ROF) [37] proposed replacing the $\int |\nabla u|^2$ regularization term with the so-called total-variation (TV) semi-norm $\int |\nabla u|$ which will allow piecewise smooth images. The resulting minimization problem is:

$$\min_u J_{TV}(u), \qquad J_{TV}(u) = \int_\Omega \alpha\sqrt{|\nabla u|^2 + \beta} + \frac{1}{2}(u-z)^2 dxdy \tag{1}$$

where β is a small perturbing parameter that prevents degeneracy of the Euler-Lagrange equation when $|\nabla u| = 0$. The Euler-Lagrange equation for this problem is

$$-\alpha\nabla.\left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}}\right) + u = z \tag{2}$$

with homogenous Neumann boundary condition $\frac{\partial u}{\partial n} = 0$. Unlike in the H^1 case this equation is highly nonlinear and the fast solution of this equation has been an active area of research over the last decade or so.

The simplest approach is the artificial time marching (or gradient descent) method used by ROF [37]. In this method the parabolic equation

$$u_t = \alpha\nabla.\left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}}\right) + (z-u) \tag{3}$$

is solved to steady state using an explicit time marching (forward Euler) scheme on the discrete equation. A steepest descent type method with a line search on the discretization of $J_{TV}(u)$ can be viewed as an explicit time marching scheme with variable time step. The problem with the time marching approach is that due to stability restrictions the time step must be taken to be very small, resulting in very slow convergence. Marquina and Osher [34] reduce the stability constraints on the time step by multiplying (3) by $|\nabla u|$.

Vogel and Oman [43] proposed a 'lagged diffusivity' fixed-point iterative method (see also [42]) for solving (2) in which the following linear equation is solved on each step

$$u^{k+1} - \alpha \nabla . \left(\frac{\nabla u^{k+1}}{\sqrt{|\nabla u^k|^2 + \beta}} \right) = z$$

to update u. This method is equivalent to a semi-implicit time marching method with infinite time step and is globally convergent with linear convergence. The linear system to be solved on each step is symmetric positive definite and several different methods have been used in the literature to solve it, these include preconditioned conjugate gradient (pcg) with incomplete Cholesky preconditioner [14], geometric multigrid [41] (either on its own or as a preconditioner for preconditioned conjugate gradient) and Algebraic multigrid [17], which is more robust with respect to small values of β than geometric multigrid. In practice accurate solution of the linear equation is not necessary, and reducing the linear residual by a factor of 10 is usually enough to give a method which is optimal in terms of time taken.

Chan, Zhou and Chan [16] recognized that Newton's Method has a small domain of convergence for this problem particularly with respect to small values of β and proposed a continuation procedure on β. To overcome this in a more fundamental way Chan, Golub and Mulet [14] replace the original problem with an equivalent (u, w) system

$$\begin{aligned} &-\alpha \nabla . w + u - z = 0 \\ &w\sqrt{|\nabla u|^2 + \beta} - \nabla u = 0, \qquad \|w\|_\infty \leq 1 \end{aligned}$$

by introducing a new variable $w = \nabla u / \sqrt{|\nabla u|^2 + \beta}$. Alternatively this system can be seen as the conditions needed to bridge the duality gap between the primal and dual problems. The new system is better behaved with respect to Newton's method due to its quasi-linearity, and the cost of each step is only slightly more than for the primal problem (2). The linear solve on each step is done using a preconditioned conjugate gradient method with incomplete Cholesky preconditioner. The method appears globally convergent with quadratic convergence. The primal-dual method incorporates the primal and the dual variable, other authors have worked directly with the dual formulation of the TV problem see Carter [5] and more recently Chambolle [6], this avoids the use of the β parameter. Incidentally, one may eliminate u in (1) to derive the dual method [6].

In [38] we proposed using a nonlinear multigrid (FAS) method (see, for example, [39, 23]) to solve (2). Our method which used a smoother based on the fixed point method but using just 3 steps of Gauss-Seidel on the linear system on each step performed well in comparison with the fixed point and primal-dual methods provided the parameter β was not too small. Nonlinear multigrid methods for total variation denoising have also been studied by Frohn-Schauf, Henn and Witsch in [22]. Chan and Chen [9, 10] and Chen and Tai [19] have multilevel methods working directly with the minimization problem (not the Euler-Lagrange equation). Finally another approach to solving (1) is the active set methods of Karkkainen and Majava [27] and Ito and Kunisch [24].

Although TV regularization is very good at recovering edges and blocky images it does suffer from the 'staircasing effect' in which smooth regions (ramps) in the original image are recovered as piecewise smooth regions (staircases) in the reconstruction. In the literature there have been many attempts to devise image denoising methods which reduce the staircasing effect seen in images denoised using TV regularization with no one approach gaining universal appeal. In most cases the emphasis is on designing new suitable regularization functionals which reduce staircasing as well as recover edges via retaining some form of the TV regularization. However no particular attention has been paid to the fast efficient solution of the resulting equations; in fact, some of these new models cannot be solved efficiently. This paper is thus motivated to study how effectively a class of staircasing reduction models can be solved by three selected iterative methods.

The underlying Euler-Lagrange equation, to be studied here, is of the form

$$-\alpha \nabla . \left(D(\sqrt{|\nabla u|^2 + \beta}) \nabla u \right) + u = z \tag{4}$$

with various choices of $D(t)$, where $D(t) = 1/t$ reproduces the standard ROF model. Our three selected iterative methods are: (i). the fixed point method; (ii) the time-marching methods; (iii) the nonlinear multigrid method as proposed in [38]. Therefore, our objective is to find the most reliable model by two criteria: quality of restoration results and efficiency of a fast iterative solution. It turns out that a modified model out of the tested models fits our criteria.

The rest of the paper is organized as follows. In Section 2 we review the various staircase reducing models that have been proposed in the literature including the class of models that we study in this paper. In Section 3 we discuss numerical solution of 4 particular PDE models in this class: discretization and iterative algorithms as well as the implementation of the iterative methods for each of the models and some numerical results. Section 4 focuses on the model which out of those tested we feel is best both in terms of reconstructed image quality and robustness of solvers and considers an effective modification to the best approach found from Section 3. Finally in Section 5 some conclusions are drawn.

2 An overview of staircasing reduction models

In this section we review various models to reduce the staircasing effect that have been proposed in the literature. In the next section we shall focus on the first class of the models.

2.1 Combining TV and H^1

A popular approach to reducing staircasing is to try and combine the ability of TV denoising to preserve edges with the ability of H^1 denoising to preserve smooth regions. In this paper we consider 4 such approaches for which we will attempt to use nonlinear multigrid to solve the resulting PDEs, they all involve a minimisation problem of the form

$$\min_u \int_\Omega \alpha\Phi(|\nabla u|) + \frac{1}{2}(u-z)^2 dxdy$$

which has the Euler-Lagrange equation

$$-\alpha\nabla.\left(\Phi'(\sqrt{|\nabla u|^2+\beta})\frac{\nabla u}{|\nabla u|}\right) + (u-z) = 0$$

where a small parameter $\beta > 0$ is added (as in the TV case with $\Phi(g) = g$) to avoid $|\nabla u| = 0$.

Model 1. Noting that the H^1 case corresponds to $\Phi(g) = g^2$, one can propose as in [2, 29]:

$$\Phi(|\nabla u|) = \frac{1}{p}|\nabla u|^p \tag{5}$$

we then have

$$\Phi'(|\nabla u|) = |\nabla u|^{p-1} \text{ and } \frac{\Phi'(|\nabla u|)}{|\nabla u|} = \frac{1}{|\nabla u|^{2-p}}$$

where $1 < p < 2$ and in order to recover edges reasonably well p should be close to 1, say 1.1.

Model 2. A more sophisticated approach would be to choose p in some way adapting to the behavior of $|\nabla u|$. To this end, Blomgren [1] and Blomgren, Chan, Mulet [2] proposed the following general choice

$$\Phi(|\nabla u|) = |\nabla u|^{p(|\nabla u|)}, \qquad \text{with } \lim_{g\to 0} p(g) = 2 \text{ and } \lim_{g\to\infty} p(g) = 1 \tag{6}$$

which results in a non-convex minimisation problem, where $p(g)$ is a monotonically decreasing function i.e TV-like regularization ($p = 1$) is used at edges, H^1-like regularization ($p = 2$) is used in flat regions and in between $p \in (1, 2)$. We have

$$\Phi'(|\nabla u|) = p(|\nabla u|)|\nabla u|^{p(|\nabla u|)-1} + p'(|\nabla u|)|\nabla u|^{p(|\nabla u|)}\log(|\nabla u|).$$

Blomgren [1] suggests the following particular choice for p :

$$p(g) = \begin{cases} 2, & g = 0 \\ ag^3 + bg^2 + cg + d, & 0 < g < sg_{max} \\ 1, & g \geq sg_{max} \end{cases} \tag{7}$$

where the third order polynomial is chosen so that $p'(0) = 0$ and $p'(sg_{max}) = 0$, g_{max} is the maximum realizable gradient and $0 < s \leq 1$. Resolving the conditions on p gives $a = \frac{2}{(sg_{max})^3}$, $b = \frac{-3}{(sg_{max})^2}$, $c = 0$ and $d = 2$. If we assume that our image is a square $n \times n$ image with values in the range $[0, 255]$, then $g_{max} = 255\sqrt{2}(1/h)$ where h is the grid spacing (see later). We note here that in a later paper, Chan, Esedoglu, Park and Yip [13] suggested taking p to be a monotonically decreasing function from 2 to 0 e.g $p(g) = \frac{2}{1+2g}$, here we focus on the case that p takes values between 2 and 1.

Model 3. A simplified alternative to (6) would be to replace $p = p(|\nabla u|)$ by $p = p(|\nabla u^*|)$ for some "known" quantity u^* approximating u (and thus 'less' nonlinear), i.e. take

$$\Phi(|\nabla u|) = \frac{1}{p(|\nabla u^*|)}|\nabla u|^{p(|\nabla u^*|)} \tag{8}$$

$$\Phi'(|\nabla u|) = |\nabla u|^{p(|\nabla u^*|)-1} \text{ and } \frac{\Phi'(|\nabla u|)}{|\nabla u|} = \frac{1}{|\nabla u|^{2-p(|\nabla u^*|)}}.$$

This choice ensures that the new minimization problem is convex. Blomgren [1] suggests $u^* = G * z$ where G is a Gaussian used to smooth the noisy image z. More recently this approach was used by Karkkainen and Majava [28] who suggest $u^* = u_{TV}$ and

$$p(|\nabla u_{TV}|) = \begin{cases} 2 & |\nabla u_{TV}| < g_1 \\ 1.5 & |\nabla u_{TV}| = g_1 \\ p_1(|\nabla u_{TV}|) & g_1 < |\nabla u_{TV}| < g_2 \\ 1 & |\nabla u_{TV}| \geq g_2 \end{cases} \tag{9}$$

where $p_1(g)$ is a second order polynomial satisfying $p_1(g_1) = 1.5$, $p_1(g_2) = 1$ and $p'(g_2) = 0$. The idea here is that a value of 1.5 is enough to recover smooth regions effectively with larger values possibly oversmoothing the image. In order that a nonlinear CG solver can be implemented effectively p takes values 2 for $|\nabla u_{TV}| < g_1$ where g_1 is small, p then jumps to a value of 1.5 and then decreases smoothly as $|\nabla u_{TV}|$ increases until $|\nabla u_{TV}| = g_2$, g_2 being small enough so that $p = 1$ at all edges in the image. The values of g_1 and g_2 are chosen using a histogram of $|\nabla u_{TV}|$ values.

Another similar, but slightly different, approach is used by Chen, Levine and Rao [20]:

$$\Phi(|\nabla u|) = \begin{cases} \frac{1}{p(|\nabla u^*)}|\nabla u|^{p(|\nabla u^*|)} & |\nabla u| \leq \epsilon \\ |\nabla u| - \frac{\epsilon p(|\nabla u^*|) - \epsilon^{p(|\nabla u^*|)}}{p(|\nabla u^*|)} & |\nabla u| > \epsilon \end{cases}$$

$$p(|\nabla u^*|) = 1 + \frac{1}{1 + k|\nabla u^*|^2}$$

where $u^* = G * z$. The difference here is that the threshold for a switch to pure TV regularization is based on the gradient of u rather than u^* (so implicitly 'nonlinear'). The function p is a monotonically decreasing rational function which is 2 at $|\nabla u^*| = 0$ and tends to 1 as $|\nabla u^*|$ tends to infinity. Some theoretical study of this model was conducted in [20].

Model 4. The approach proposed in [2, 1] tries to combine TV and H^1 in a convex combination:

$$\Phi(|\nabla u|) = \pi(|\nabla u|)|\nabla u| + (1 - \pi(|\nabla u|))|\nabla u|^2 \tag{10}$$

with $\lim_{g\to 0} \pi(g) = 0$ and $\lim_{g\to\infty} \pi(g) = 1$. In this case

$$\Phi'(|\nabla u|) = \pi'(|\nabla u|)(|\nabla u| - |\nabla u|^2) + \pi(|\nabla u|)(1 - 2|\nabla u|) + 2|\nabla u|.$$

It is suggested in [1] to take $\pi(g) = 2 - p(g)$ where p is the polynomial outlined in (7).

Finally we mention several other (less representative) methods which also compromise between TV and H^1 regularization. The first is the inf-convolution of the TV and H^1 regularization functionals proposed in [7] the resulting minimization problem is equivalent to:

$$\min_u \int_{|\nabla u| \geq \epsilon} |\nabla u| dxdy + \frac{\epsilon}{2} \int_{|\nabla u| < \epsilon} |\nabla u|^2 dxdy + \frac{\lambda}{2} \int_\Omega (u - z)^2 dxdy.$$

Another approach proposed in [25] by Ito and Kunisch is to minimize the functional

$$\int_\Omega \alpha \Phi(|\nabla u|) + \frac{1}{2}(u - z)^2 dxdy$$

where Φ is chosen so that it behaves like $|\nabla u|$ for large values of $|\nabla u|$ and (in contrast to other models seen above) for small values of $|\nabla u|$ and behaves like $|\nabla u|^2$ for mid range values of $|\nabla u|$.

2.2 Higher Order Models

Another popular way to reduce staircasing is to introduce in some way higher order derivatives into the regularization term. In [7] Chambolle and Lions do this by minimizing the inf-convolution of the TV norm and a second order functional

$$\min_{u_1, u_2} \int_\Omega |\nabla u_1| + \mu |\nabla (\nabla u_2)| + \frac{\lambda}{2}(u_1 + u_2 - z)^2 dxdy. \tag{11}$$

Here u is decomposed into a smooth function u_2 and a function containing the discontinuities u_1. Another way to use higher order derivatives is introduced by Chan et al [15] in which the non-convex functional

$$\int_\Omega \left[\alpha \sqrt{|\nabla u|^2 + \beta} + \mu \frac{(\Delta u)^2}{(\sqrt{|\nabla u|^2 + 1})^3} + \frac{1}{2}(u - z)^2 \right] dxdy$$

is minimized. Here the $(|\nabla u|^2 + 1)^{-3/2}$ term multiplying the higher order term ensures that true edges (with very large gradient) are not penalized while staircasing is reduced. Instead of combining the TV norm and second order derivatives within one regularization functional Lysaker and Tai [33] use two regularization functionals:

$$E_1(u) = \int_\Omega |\nabla u| + \frac{\lambda_1}{2}(u - z)^2 dxdy$$

$$E_2(v) = \int_\Omega (v_{xx}^2 + v_{xy}^2 + v_{yx}^2 + v_{yy}^2)^{1/2} + \frac{\lambda_2}{2}(v - z)^2 dxdy.$$

Their approach is to use an iterative procedure in which they simultaneously apply an explicit time marching method to the Euler-Lagrange equation of each functional. After each step the current iterates u^k and v^k are combined in a convex combination to give $w = \theta^k u^k + (1 - \theta^k) v^k$; u^k and v^k are then overwritten with w in preparation for the next step. Here θ^k is chosen to be 1 only at the largest jumps (edges) allowing smaller jumps due to staircasing to be suppressed by the higher order PDE. In an earlier paper the same authors with Lundervold [32] considered E_2 on its own and another functional $\int_\Omega |u_{xx}| + |u_{yy}| + \lambda/2(u - z)^2 dxdy$ which was not rotationally invariant.

2.3 Other ways to reduce staircasing

Marquina and Osher [34] preconditioned the right hand side of the parabolic equation (3) with $|\nabla u|$ which had a staircase reducing effect. This is because the inclusion of β only in the $|\nabla u|$ term multiplying the first term on the right hand side of (3) and the use of an upwind difference scheme for the $|\nabla u|$ multiplying the

second term leads to a different numerical steady state which is less staircased than the TV problem. In a similar vein is the algebraic scaling approach used in [26] which is equivalent to using

$$u_t = \min\left(\frac{a_{max}}{2}(|\nabla u|^2+\beta)^{1/2}, 1\right)\nabla.\left(\frac{\nabla u}{\sqrt{|\nabla u|^2+\beta}}\right) + \lambda(z-u)$$

where a_{max} is a parameter to be chosen. We also mention the Gauss-curvature driven diffusion approach (not related to any optimization problem) proposed in [30] which has several desirable properties including staircase reduction:

$$u_t = \nabla.\left(\left|\frac{u_{xx}u_{yy}-u_{xy}^2}{(1+u_x^2+u_y^2)^2}\right|\nabla u\right).$$

See also [4, 35], [11], [12, 36, 40, 46] for the iterated TV model, the TV L^1 model and the texture models respectively.

3 Algorithms for solving the combined TV and H^1 models

Our aim in this paper is first to implement and compare 3 numerical algorithms for solving the the above listed 4 combined TV and H^1 models, and then to propose a modified staircasing reduction model which can be efficiently solvable. The selected algorithms are: (i) explicit time marching methods; (ii) fixed point type methods; (iii) the nonlinear multigrid method [38]. We now outline our discretization scheme, introduce the iterative methods and give details of implementation and numerical results.

Remark 1 *As mentioned earlier less focus has been given to the efficient solution of the models of the previous section than their effectiveness in reducing staircasing. In [2] a fixed point type method is proposed to solve model 2 and model 4 but no numerical results are given. In [28] a nonlinear conjugate gradient method is used to solve model 3 with the particular choice of p outlined above. In the case of model 1 and model 3, the choice of D in* (4) *is similar to the TV case with the added advantage in model 3 that when* $|\nabla u|$ *is small* $p(|\nabla u^*|)$ *should be close to 2, preventing jumps in the diffusion coefficient as large as in the TV case. For models 2 and 4 the Euler-Lagrange equation is more nonlinear than in the TV case. We also note that many iterative methods can benefit from using the separate acceleration technique of [44].*

Discretization. Below we outline the discretization scheme used. Given that the image data will be given in the form of $n \times m$ pixel values, each representing average light intensity over a small rectangular portion of the image, we use a cell-centered discretization of our domain and a cell-centered finite difference scheme to discretize (2). From now on we assume that $\Omega = [0,n]\times[0,m]$. We discretize the domain Ω into Ω^h with $n \times m$ rectangular cells of size $h \times k$ where $h = k = 1$, with grid points placed at the center of the cells so grid point (i,j) is located at

$$(x_i, y_j) = ((2i-1)h/2,\ (2j-1)k/2)\,.$$

Denoting the discrete version of equation (4) by $N_h(u_h) = z_h$, we have:

$$(N_h(u_h))_{i,j} = u_{i,j} - \alpha_h\left[\delta_x^-\left(D_{ij}(g_{ij})\delta_x^+ u_{i,j}\right) + \gamma\delta_y^-\left(D_{ij}(g_{ij})\gamma\delta_y^+ u_{i,j}\right)\right] = z_{ij} \tag{12}$$

where u_h and z_h are grid functions on Ω^h,

$$g_{i,j} = \frac{1}{h}\sqrt{(\delta_x^+ u_{i,j})^2 + (\gamma\delta_y^+ u_{i,j})^2 + \beta_h}$$

$$D_{ij}(g_{ij}) = \begin{cases} \frac{1}{h}(g_{ij}^{-(2-p)}) & \text{Model 1} \\ \frac{1}{h}(p(g_{ij})g_{ij}^{p(g_{ij})-1} + p'(g_{ij})g_{ij}^{p(g_{ij})}\log(g_{ij}))g_{ij}^{-1} & \text{Model 2} \\ \frac{1}{h}(g_{ij}^{-(2-p_{i,j})}) & \text{Model 3} \\ \frac{1}{h}(\pi'(g_{ij})(g_{ij}-g_{ij}^2) + \pi(g_{ij})(1-2g_{ij}) + 2g_{ij})g_{ij}^{-1} & \text{Model 4} \end{cases} \tag{13}$$

$$\alpha_h = \alpha/h,\ \beta_h = h^2\beta \text{ and } \gamma = h/k = 1$$

and

$$\delta_x^{\pm} u_{i,j} = \pm\left(u_{i\pm1,j} - u_{i,j}\right) \qquad \delta_y^{\pm} u_{i,j} = \pm\left(u_{i,j\pm1} - u_{i,j}\right).$$

Note that D is actually only dependant on (i,j) in the case of model 3. We also have boundary condition:

$$u_{i,0} = u_{i,1}, u_{i,m+1} = u_{i,m}, u_{0,j} = u_{1,j}, u_{n+1,j} = u_{n,j}. \tag{14}$$

Remark 2 *Unlike in the TV case where the choice of Ω is not important provided α_h and β_h are chosen to be the same, whatever the value of h, there is not in all cases here a straightforward relationship (the exception is model 1) between the case $\Omega = [0,n] \times [0,m]$ i.e $(h,k) = (1,1)$ and the case $\Omega = [0,1] \times [0,1]$ i.e $(h,k) = (1/n, 1/m)$. We have chosen the former to be consistent with the majority of papers.*

We now introduce the 3 algorithms to be used.

Algorithm 1 (Time Marching)
Choose initial guess u_h^0
Set $k = 0$.
While $\|vec(z_h - N_h(u_h^k))\|_2 > tol$
$u_h^{k+1} = u_h^k + \Delta t\left[z_h - N_h(u_h^k)\right]$
$k = k + 1$
end

The time step Δt is determined by experiment as the largest value which gives stability of the algorithm. Here vec denotes the stacking of a grid function into a vector. *tol* is typically $10^{-4}\|\text{vec}(z_h - N_h(z_h))\|_2$, where $\|.\|_2$ is the Euclidean norm.

Algorithm 2 (Fixed Point Method)
Choose initial guess u_h^0 and Set $k = 0$.
While $\|vec(z_h - N_h(u_h^k))\|_2 > tol$
Set u_h^{k+1} to be the result of applying some iterative method to:
$L_h(u_h^k)w_h = z_h$
$k = k + 1$
end

The linear operator $L_h(u_h^k)$ on step $k+1$ is given by the stencil:

$$\begin{bmatrix} 0 & -\alpha\lambda D_{ij}(g_{ij}^k) & 0 \\ -\alpha D_{i-1,j}(g_{i-1,j}^k) & 1 + \alpha\Pi_{ij} & -\alpha D_{ij}(g_{ij}^k) \\ 0 & -\alpha\lambda D_{i,j-1}(g_{i,j-1}^k) & 0 \end{bmatrix}$$

where $\Pi_{ij} = (1+\lambda)D_{ij}(g_{ij}^k) + D_{i-1,j}(g_{i-1,j}^k) + \lambda D_{i,j-1}(g_{i,j-1}^k)$. The linear solver used in most cases is a geometric multigrid method with red-black Gauss-Seidel pre-correction and black-red Gauss-Seidel post correction as smoother (c.f. [41, 43]). We only require a relatively small decrease in the linear residual (typically a halving) as this seems to give the best results in terms of overall cpu time. We may also require the use of methods such as preconditioned conjugate gradient and minimum residual, we stack the grid functions along rows of pixels into vectors $\mathbf{u}_h^k = (u_{1,1}^k, u_{2,1}^k, ..., u_{n,1}^k, u_{1,2}^k, ..., u_{n,m}^k)^T$ and $\mathbf{z}_h$, the resulting system is of the form $A(\mathbf{u}_h^k)\mathbf{w}_h = \mathbf{z}_h$ where A is symmetric.

Nonlinear Multigrid. Multigrid methods based on the recursive application of smoothing relaxation and coarse grid correction are efficient solvers for a wide range of linear and nonlinear elliptic partial differential equations. Below we give a brief introduction to the full approximation scheme (FAS) nonlinear multigrid

scheme and review the smoother used in [38] for the TV problem before giving the algorithm for a similar scheme to be used in this paper. For a more comprehensive treatment of multigrid see for example [3, 18, 39, 45] and references therein.

Denote by $N_h u_h = z_h$ the nonlinear system (12) and by Ω^{2h} the $n/2 \times m/2$ cell-centered grid which results from standard coarsening of Ω^h i.e the cell-centered grid with grid spacing $(2h, 2k)$. If v_h is an approximation to the solution u_h define the error in v_h by $e_h = u_h - v_h$ and the residual by $r_h = z_h - N_h v_h$ recall also that these quantities are related by the nonlinear residual equation:

$$N_h(v_h + e_h) - N_h v_h = r_h.$$

If e_h is 'smooth' it can be well approximated on Ω^{2h}. To describe a multigrid cycle, we define the transfer and smoothing operators.

The **Restriction** operator is

$$I_h^{2h} v_h = v_{2h}$$

where

$$(v_{2h})_{i,j} = \frac{1}{4}[(v_h)_{2i-1,2j-1} + (v_h)_{2i-1,2j} + (v_h)_{2i,2j-1} + (v_h)_{2i,2j}]$$
$$1 \le i \le n/2, 1 \le j \le m/2.$$

The **Interpolation** operator is defined by

$$I_{2h}^h v_{2h} = v_h$$

where

$$\begin{gathered}(v_h)_{2i,2j} = \tfrac{1}{16}[9(v_{2h})_{i,j} + 3[(v_{2h})_{i+1,j} + (v_{2h})_{i,j+1}] + (v_{2h})_{i+1,j+1}] \\ \tfrac{1}{16}[9(v_{2h})_{i,j} + 3[(v_{2h})_{i-1,j} + (v_{2h})_{i,j+1}] + (v_{2h})_{i-1,j+1}] \\ \tfrac{1}{16}[9(v_{2h})_{i,j} + 3[(v_{2h})_{i+1,j} + (v_{2h})_{i,j-1}] + (v_{2h})_{i+1,j-1}] \\ \tfrac{1}{16}[9(v_{2h})_{i,j} + 3[(v_{2h})_{i-1,j} + (v_{2h})_{i,j-1}] + (v_{2h})_{i-1,j-1}] \\ 1 \le i \le n/2, 1 \le j \le m/2.\end{gathered}$$

Local smoothers. At grid point (i, j) the Euler-Lagrange equation is

$$\begin{aligned}&u_{i,j} - \alpha_h \left(D_{i,j}(g_{i,j})(u_{i+1,j} - u_{i,j}) - D_{i-1,j}(g_{i-1,j})(u_{i,j} - u_{i-1,j})\right. \\ &\left.+\gamma^2[D_{i,j}(g_{i,j})(u_{i,j+1} - u_{i,j}) - D_{i,j-1}(g_{i,j-1})u_{i,j} - u_{i,j-1})]\right) = z_{i,j}\end{aligned} \tag{15}$$

where g_{ij} depends on $u_{i+1,j}$, $u_{i,j+1}$ and $u_{i,j}$. If we freeze all non (i, j) terms at the current approximation then we have a nonlinear equation in one variable to solve in order to update the approximation at (i, j), which can be done using a step of Newton's method. This type of local nonlinear relaxation scheme is known as Gauss-Seidel Newton. In our investigations into the TV problem we found that this type of method only converged with heavy under-relaxation and was not useful as a smoother for the nonlinear multigrid method. An alternative approach would be to freeze also the g terms in (15) rather than just the $u_{i,j}$ terms in g. In this case we have a linear equation in one variable to solve at each grid point. This type of approach is more stable than Gauss-Seidel Newton and can be speeded up in the TV case by the application of nonlinear multigrid, however we found in [38] that a better option is a smoother in which the Euler-Lagrange equation is linearized globally as in the fixed point method before a few (3 seems to be optimal) steps of linear Gauss-Seidel relaxation are applied to the linear system i.e $D_{ij}(g_{ij})$ is evaluated for all (i, j) at the beginning of the smoothing step using the value of the current iterate before linear Gauss-Seidel is used to update. We call this smoother FPGS. For clarity the algorithm for one step of the FPGS smoother is given below

$$v_h \leftarrow FPGS(v_h, N_h, z_h)$$

for $i = 1 : n$

for $j = 1 : m$

Evaluate $g_{i,j} = ((\delta_x^+ v_{i,j})^2 + (\gamma\delta_y^+ v_{i,j})^2 + \beta_h)^{-1/2}$
and $D_{i,j}(g_{i,j})$ according to N_h using (13).
end
end
Perform Gauss-Seidel steps on linear system (start from $w = v_h$)
for $iter = 1 : it$
for $j = 1 : m$
for $i = 1 : n$
$\bar{w} \leftarrow w$, $T_0 = D(g_{i,j})_{i,j}$, $T_1 = D(g_{i-1,j})_{i-1,j}$, $T_2 = D(g_{i,j-1})_{i,j-1}$,

$$w_{i,j} \leftarrow \frac{z_{i,j} + \alpha_h(T_0(\bar{w}_{i+1,j} + \gamma^2\bar{w}_{i,j+1}) + T_1\bar{w}_{i-1,j} + \gamma^2 T_2\bar{w}_{i,j-1})}{1 + \alpha_h((1+\gamma^2)D(g_{i,j})_{i,j} + T_1 + \gamma^2 T_2)}$$

or an appropriate modification if (i, j) is a boundary point.
end
end
end
$v_h \leftarrow w_h$
We take $it = 3$ unless otherwise stated.

Any iterative method which smooths the error on the fine grid i.e damps high frequency Fourier components of the error while not necessarily reducing its size greatly can be improved by the use of coarse grid correction, in which a coarse grid analogue of the residual equation is solved (solution on the coarse grid being less expensive than on the fine grid) to obtain a coarse grid approximation of the error, which is then transferred back to the fine grid to correct the approximation v_h.

The nonlinear multigrid method. We are ready to state the algorithm for the FAS multigrid method with FPGS smoother that we use in this paper. The method is a V-cycle method, which means that just one recursive call to the algorithm is made on each level to approximately solve the coarse grid problem, we have found that using the more expensive W-cycle (performing two cycles to solve the coarse grid problem on each level) does not give a significant improvement in convergence and therefore is not pursued.

Algorithm 3 (Nonlinear Multigrid Method)
Set v_h to be some initial guess.
While $\|vec(z_h - N_h(v_h))\|_2 > tol$
$v_h \leftarrow NLMG^h(v_h, N_h, z_h, \nu_1, \nu_2)$
end
where $NLMG^h$ is defined recursively as $v^h \leftarrow NLMG^h(v_h, N_h, z_h, \nu_1, \nu_2)$ as follows:

1. *If Ω^h =coarsest grid, solve $N_h u_h = z_h$ using Fixed Point Method and stop.*
   Else For $l = 1, .., \nu_1$ $v_h \leftarrow FPGS(v_h, N_h, z_h)$
2. $v_{2h} = I_h^{2h} v_h$, $\bar{v}_{2h} = v_{2h}$, $z_{2h} = I_h^{2h}(z_h - N_h v_h) + N_{2h} v_{2h}$
3. $v_{2h} \leftarrow NLMG^{2h}(v_{2h}, N_{2h}, z_{2h}, \nu_1, \nu_2)$
4. $v_h \leftarrow v_h + I_{2h}^h(v_{2h} - \bar{v}_{2h})$
5. *For $l = 1, .., \nu_2$ $v_h \leftarrow FPGS(v_h, N_h, z_h)$*

Here $v_h \leftarrow FPGS(v_h, N_h, z_h)$ denotes the updating of v_h via one step of the FPGS smoother. N_{2h} is the coarse grid analogue of N_h which results from standard coarsening i.e the nonlinear operator which results from discretizing the Euler-Lagrange equation using a cell-centered grid with grid spacing $(2h, 2k)$. The number of

pre and post-correction smoothing steps (ν_1 and ν_2) we use depends on the model under consideration, details will be given below. We use standard cell-centered interpolation and restriction operators outlined earlier, and take the coarsest grid as 4×4.

Numerical Results. Now we present some numerical results and give details of some of the issues regarding our implementation of iterative methods for each of the four models. It should be remarked that although Algorithms 1-2 have been used for solving some of these equations it is up to now unclear whether Algorithm 3 would work for the models considered.

Tests are run on the test hump image seen in Figure 1, which has both smooth regions, high intensity edges and low intensity edges and the more realistic Lenna image shown in Figure 2. In each case we have tried to choose parameters which give the optimal reconstruction, focusing on the need to reduce staircasing. What the optimal reconstruction is, is somewhat subjective, as a guide we have used mesh and image plots as well as Peak signal to noise ratio (PSNR) defined by

$$PSNR = 20\log_{10}(\frac{255}{RMSE(u,u^0)}), \quad RMSE(u,u^0) = \sqrt{\frac{\sum_{(i,j)}(u_{i,j} - u^0_{i,j})^2}{nm}}$$

where u is the reconstructed image and u^0 is the true image.

The PSNR does not always give a clear guide as to whether one image is less staircased than another as can be seen in the hypothetical 1D example in Figure 3, so we also take into account the value of $PSNR_{grad}$ which we define as $1/2(PSNR(u_x, u^0_x) + PSNR(u_y, u^0_y))$ this should measure how well the derivatives of the reconstruction match those of the true image. All methods were implemented in MATLAB on a Sun Fire 880.

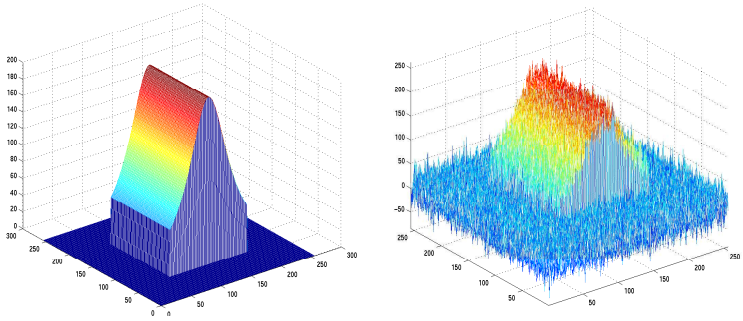

Figure 1: Mesh plots of true (left) and noisy (right) Hump image

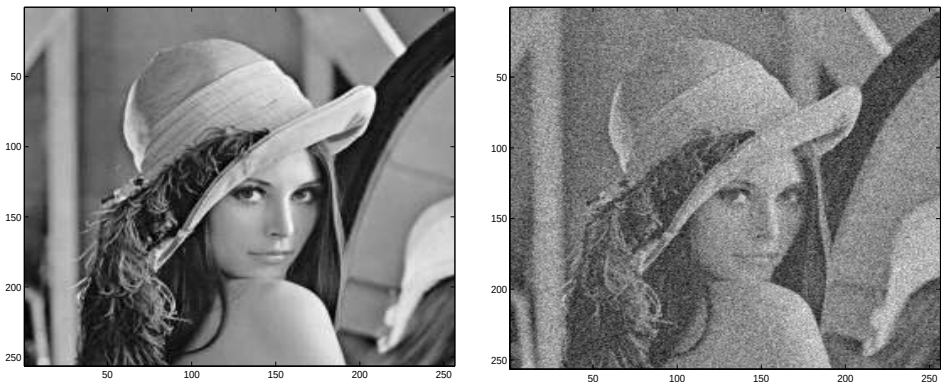

Figure 2: True (left) and noisy (right) Lenna image

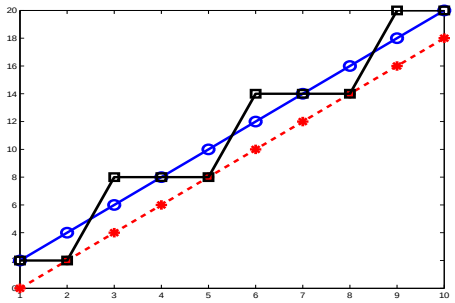

Figure 3: A simple 1D example of a staircased reconstruction (squares) which will have a higher PSNR than the smooth reconstruction (stars), the smooth reconstruction in this case has exactly the same gradient as the true solution (circles)

In Figure 4, we present some plots showing the results of applying each of the four models to the test hump image, we also show the results of applying TV and H^1 regularization. We remark that it is not our intention in this paper to carry out a detailed comparison of the various staircase reducing methods in terms of the quality of the reconstructed images, however we make a few general comments. To some extent all the models can recover better the smooth regions of the image than the original TV model (1) but in our experience models 2 and 3 seem to give better overall results than model 1 (as would be expected) and model 4 in which there is some over-smoothing of the edges (particularly the low intensity edges), this is predicted in [1]. With models 2 and 3 for the test image shown we have been able (with suitable choices of parameters) to reduce the staircasing present in the TV reconstructed image while still recovering well the high and low intensity edges in the image.

Model 1. For this model we consider three choices of p, $p = 1.1$, $p = 1.5$ and $p = 1.9$ mainly to highlight the effect the value of p has on the convergence of the various methods (the latter two choices will of course over-smooth the edges). A suitable value of α_h to remove the noise is chosen for each value, the larger p is the smaller α_h needs to be. The effect that the parameter β_h has on convergence is also studied.

In Table 1 we show results (number of steps required for convergence and cpu time in seconds) for the Fixed Point method (FP), Nonlinear multigrid method (NLMG) and the explicit time marching method (TM) run on model 1 for the hump image with 3 different values of p, 1.1, 1.5 and 1.9 the corresponding values of α_h are 52, 24 and 15. Also shown are results for the smoother (FPGS) run on its own. Shown are results for various values of β_h. In all cases the initial guess is taken to be the noisy image z and the stopping criteria is a reduction in the residual by a factor of 10^{-4}. As linear solver in the fixed point method we use a linear multigrid method with 2 pre and 2 post correction smoothing steps of Gauss-Seidel relaxation and stop the iterations when the linear residual has been reduced by a factor of 0.5. Shown in the table are the choices of ν_1 and ν_2 which give the optimal nonlinear multigrid method for each case, also shown is the value of the time step in the time marching method.

We observe that the closer p is to 2 the easier the problem is to solve, less steps are required for each of the methods and less smoothing steps are required in the nonlinear multigrid method. We see that for $p = 1.9$ the convergence of the various methods is seemingly invariant to the value of β_h. For $p = 1.5$ decreasing the value of β_h has only a small effect on the FP method and the FPGS smoother and no effect on the nonlinear multigrid method. In the case that $p = 1.1$ the value of β_h has a significant effect on convergence. We see that as β_h is decreased from 10^{-2} to 10^{-4} the cost of the fixed point method increases by 3 times. The cost of the nonlinear multigrid method doubles and more pre and post correction steps are needed to ensure convergence. We found that the time marching method cannot converge in a reasonable number of steps. If β_h is reduced to 10^{-10} only the fixed point method converges in a reasonable number of steps (in this case a pcg linear solver with Cholesky preconditioner gives the best results). This breakdown of the nonlinear multigrid convergence

Table 1: Comparison of the 3 main Algorithms for Model 1 with various p and β

p	β_h	FP		TM		
		steps	cpu	Δt	steps	cpu
1.1	10^{-2}	43	73	5×10^{-4}	9502	2540
	10^{-4}	73	216			
1.5	10^{-2}	14	19	1×10^{-3}	4054	536
	10^{-4}	16	23	1×10^{-3}	4053	536
	10^{-10}	16	23	5×10^{-4}	8150	1131
1.9	10^{-2}	6	8.8	1×10^{-2}	303	56
	10^{-10}	6	8.8	1×10^{-2}	303	56

p	β_h	NLMG			FPGS	
		ν_1/ν_2	steps	cpu	steps	cpu
1.1	10^{-2}	5/5	4	34	748	680
	10^{-4}	10/10	4	66	4389	4036
1.5	10^{-2}	1/1	6	13	78	61
	10^{-4}	1/1	6	13	94	74
	10^{-10}	1/1	6	13	119	93
1.9	10^{-2}	1/1	3	6.9	29	23.9
	10^{-10}	1/1	3	6.9	29	23.9

for very small β_h was also observed in the TV ($p = 1$) case. Apart from this last case the nonlinear multigrid method significantly speeds up the smoother FPGS and is faster than the time marching and fixed point methods.

Model 2. For this model $p(|\nabla u|)$ is chosen to be the polynomial (7). There were several problems that occurred during the implementation of iterative solvers for this model. The first problem is that the functional is non-convex and the initial guess seems to have an effect on the quality of the final image. If we take the noisy image z as initial guess we appear to converge to a minimum which is still highly oscillatory. To achieve the reconstruction of the test image shown in Figure 4 we had to take the solution to the TV problem as initial guess, the following discussion relates to experiments run using this initial guess. The second problem is that unlike in the TV case the D_{ij} terms can take negative values, as a consequence the previous smoother FPGS is *no longer* adequate. We proposed a modification of this smoother (to be denoted by FPGS2). Instead of updating u^{k+1} by applying 3 Gauss-Seidel steps to the linear system $L(u_h^k)w_h = z_h$ we apply 3 Gauss-Seidel steps to the *new* linear system $(\lambda + L(u_h^k))w_h = z_h + \lambda u_h^k$ (essentially we add a λu term to both sides of the Euler-Lagrange equation and lag the right hand side term). Taking λ large enough will ensure diagonal dominance of the inner linear system and hence positive definiteness, which ensures convergence of the Gauss-Seidel steps. In addition we have also used this approach when implementing the fixed point method. We tried to implement the fixed point method in its original form but had problems finding a suitable inner solver (linear multigrid did not converge and pcg was not an option) we settled on the minimum residual method but found that the outer fixed point steps stagnated, this was also the case when we used a direct solver to solve the linear system. Using the modified fixed point method, we can use linear multigrid or pcg as the inner linear solver and the outer steps also converge.

We implemented the time marching method, the modified fixed point method and the nonlinear multigrid method with FPGS2 smoother on the test hump image using a value of $s = 0.2$, $\alpha_h = 10$ and $\lambda = 7$, in this case only 2 pre and 2 post correction smoothing steps were required in the nonlinear multigrid method which converged in 9 steps and was around 1.75 times as fast as the modified fixed point method and over 5 times as fast as the time marching method. However when we tried to implement this model for the Lenna image we could not achieve a reasonable quality reconstruction, the image tended to look too blurred or be contaminated with undesirable artifacts. In addition we found that the nonlinear multigrid method is not effective in that the convergence stagnates unless a large number (10 or more) of smoothing steps is used and the total number of smoothing steps in this case is more than if the smoother were run on its own. The convergence of the modified fixed point method also seems somewhat unstable and typically the number of steps required by the modified fixed point and time marching methods is considerably larger than the case of the hump image above. We note that some of the problems with the iterative methods described above also occur in the case of the hump image for larger values of s (although these do not produce good reconstructions). More work is needed on this model before we can draw any firm conclusions.

Finally we note that the value of β_h seems to have no effect on convergence for this model and so it is taken to be very small (10^{-10}) in the implementation.

Model 3. We have implemented model 3 with the choice of $p(|\nabla u^*|)$ described by (9). We have been able to implement a working nonlinear multigrid method (with the usual FPGS smoother) as well as the fixed point and time marching methods.

For the parameters g_1 and g_2 in (9) we take $g_1 = g^*_{max}/50$ (as in [28]) and $g_2 = sg^*_{max}$ where $0 < s < 1$ and is chosen to give the best visual results, g^*_{max} is the maximum value of $g^*_{i,j}$ over all (i, j) where the $g^*_{i,j}$ is the discretization of $|\nabla u^*|$ at grid point (i, j), u^* in this case being the TV solution u_{TV}.

In Table 2 (left) results of running FP, NLMG and TM on model 3 for the hump test image are shown. In this case we take $s = 0.3$ and $\alpha_h = 30$, β_h in this case appears to have no effect on convergence and is taken to be 10^{-10}. We take z as the initial guess and the same stopping criteria as above is used. One pre and one

Table 2: Comparison of Fixed Point, Time Marching and Nonlinear Multigrid for Model 3 (top) and Model 4 (bottom) on the hump image and the Lenna Image

Method	Model 3			
	Hump image		Lenna image	
	Steps	cpu(s)	Steps	cpu(s)
FP	8	11.8	10	13.8
FPGS	33	24.3	22	17.3
NLMG	4	8.4	5	10.5
TM	213	27.9	169	24.8

Method	Model 4			
	Hump image		Lenna image	
	Steps	cpu(s)	Steps	cpu(s)
FP	16	17.9	22	24.7
FPGS	140	31.3	78	17.5
NLMG	6	8.0	8	10.3
TM	378	34.2	245	21.8

post correction smoothing step is used in the nonlinear multigrid method, for the fixed point method linear multigrid is used as the linear solver again with the same stopping criteria as in model 1. The time step in the time marching method is $\Delta t = 8.0 \times 10^{-3}$.

We observe that the nonlinear multigrid method reduces the cost of the smoother alone by approximately 65%. Nonlinear multigrid is around 1.4 times faster than the fixed point method and around 3.3 times as fast as the time marching method.

In our second test, we compare the performance of fixed point, time marching and nonlinear multigrid on the more realistic Lenna image. In this case we take $s = 0.9$ and $\alpha_h = 11$. The implementation is as above, except that the time step $\Delta t = 2.2 \times 10^{-2}$ is used in the time marching method. The usual initial guess and stopping criteria are used, results are given in Table 2 (left). In this case the speed up in the smoother achieved by the nonlinear multigrid method is around 40%, the nonlinear multigrid method is around 1.3 times as fast as the fixed point and around 2.4 times faster than the time marching method.

Model 4. We consider (10) only for the case

$$\pi(x) = \frac{\epsilon x}{\epsilon x + q} \tag{16}$$

In this case the functional is convex (see [2] for the conditions on π required for a convex functional). Also

$$D(x) = \frac{\Phi'(x)}{x} = \frac{(\epsilon + q)(\epsilon x + 2q)}{(\epsilon x + q)^2}$$

which is positive for nonnegative x ensuring a positive definite linear system in the fixed point method. With this choice we have successfully implemented nonlinear multigrid fixed point and time marching methods. With other choices of $\pi(x)$ e.g $2 - p(x)$ where p is the third order polynomial, we may not have a convex functional and some of same issues as in the case of Model 2 may arise. We are not aware of the choice (16) being used before but in our experience it is easier to implement iterative solvers for this case.

We have found that the choice of ϵ is more important than the choice of q in obtaining a reasonable reconstruction. With our choice of π the Euler-Lagrange equation is not degenerate for $|\nabla u| = 0$ and so we take $\beta_h = 0$.

In Table 2 (right) we show some results for the FP, NLMG and TM methods run on model 4 for the hump image, with the particular choice of π outlined above. For the parameters ϵ and q in π we take values 0.001 and 0.005 respectively, the value of α_h is 9. We have found in this case that the fastest multigrid method was achieved if we took the parameter *it* in the FPGS smoother to be 1 rather than the usual 3. The initial guess, stopping criteria and linear solver for the fixed point method are the same as in the case of model 1 and model 3. In the nonlinear multigrid method we use 2 pre and 2 post correction smoothing steps and in the time marching method we use a time step $\Delta t = 1.3 \times 10^{-2}$.

We observe that the nonlinear multigrid method reduces the cost of the smoother alone by around 75%. The nonlinear multigrid method is ≈ 2.2 times as fast as the fixed point method and ≈ 4.3 times as fast as the time marching method.

We also applied model 4 to the Lenna image, results are shown in Table 2 (right). The value of q and ϵ are as above, but $\alpha_h = 5$. The implementation is as above, except that the time step in the time marching method is $\Delta t = 2.7 \times 10^{-2}$. In this case the FPGS smoother on its own performs quite well and is actually slightly faster than the fixed point method with linear multigrid inner solver. The nonlinear multigrid method is 1.7 times faster than FPGS. The time marching method is actually quite competitive in this case at around twice the cost of the nonlinear multigrid method.

Remark 3 *Although model 4 did not perform that well on the hump image with oversmoothing of some edges, we have observed for more realistic images like the Lenna image, where the intensity of edges is more uniform, this model does not perform that badly in comparison with model 3 as can be seen from the plots in Figure 5.*

To summarise we have successfully implemented the three iterative methods for both model 3 and model 4 with a specific choice of π. At the moment their are still some outstanding issues regarding both the robustness of iterative solvers and the quality of the reconstructed image for model 2 and model 4 with other possible π, we therefore favour model 3 with the nonlinear multigrid solver as a method which can achieve good quality reconstructions and can be solved simply and efficiently. In the next section we consider other possible choices of $p(|\nabla u^*|)$ and show that the nonlinear multigrid method is the most efficient solver.

4 A modified staircasing reduction model

We hope to improve on the above recommended model 3 further. To this end, we wish to simplify the specification of $p(v)$ in (9) while maintaining the smooth transition of $p(v) = 1$ to $p(v) = 1.5$. Our proposed modification is still of the general type (4)

$$-\alpha \nabla . \Big(\Phi(|\nabla u|, |\nabla u^*|) \nabla u \Big) + u = z \tag{17}$$

where

$$\begin{aligned} &\Phi(t, v) = \tfrac{1}{p(v)} t^{p(v)} \\ &p(v) = 1.5 \left(1 + \tfrac{2v}{g_2}\right) \left(\tfrac{v - g_2}{g_2}\right)^2 + \left(1 - \tfrac{2(v-g_2)}{g_2}\right) \left(\tfrac{v}{g_2}\right)^2 \end{aligned} \tag{18}$$

and $p(t)$ is a cubic Hermite polynomial satisfying $p(g_1) = 1.5$, $p(g_2) = 1$ and $p'(g_1) = p'(g_2) = 0$ (here we take $g_1 = 0$). An alternative choice of $p(v)$ is a cubic Hermite polynomial satisfying $p(g_1) = 2$, $p(g_2) = 1$ and $p'(g_1) = p'(g_2) = 0$ (which has been found to perform similarly). Here $u^* = u_{TV}$ is the numerical solution from the standard ROF model, which distinguishes smooth regions and edges in an image. As with other models, the idea is again to respect large gradients (edges) and to reduce the effect of TV for small gradients (smooth regions). First of all, as illustrated in Figure 6, one can observe that such a model is simpler than the 'three piece' choice for $p(|\nabla u^*|)$ used in [28]. It only remains to test, in comparison to previous models, how effectively the new model can reduce the staircasing effect and how efficiently it can be solved by our selected iterative methods.

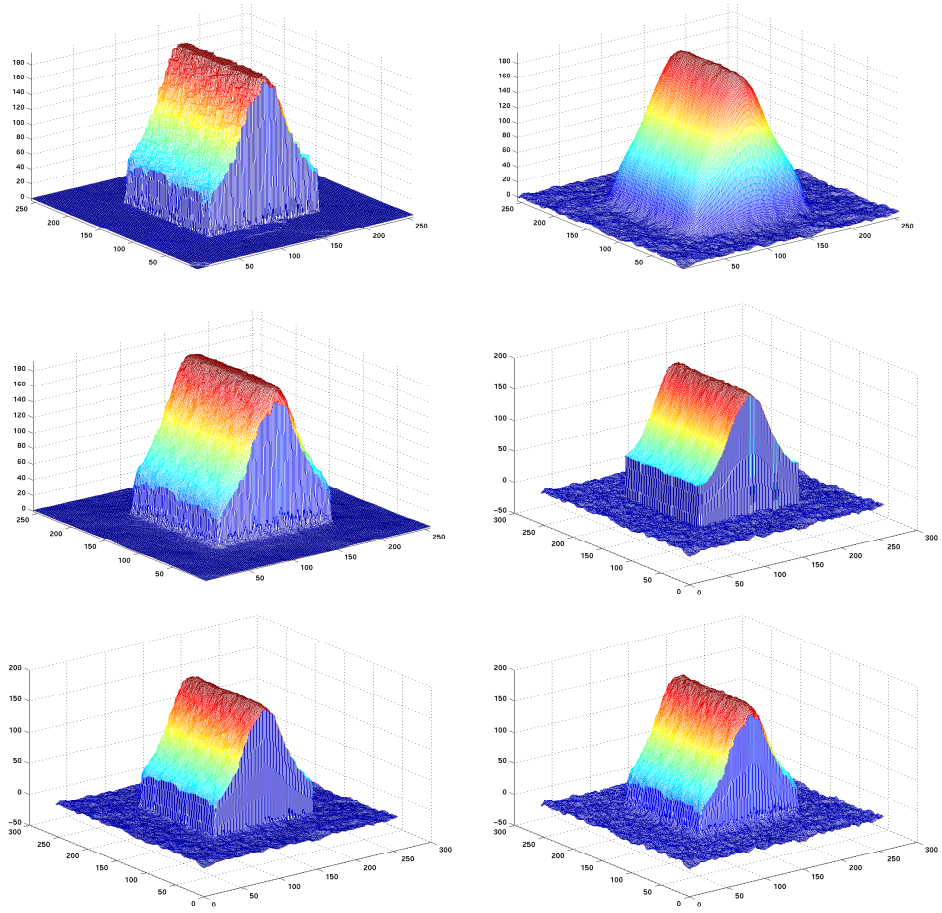

Figure 4: From top left to bottom right, the images recovered using TV, H^1, model 1 ($p = 1.1$), model 2, model 3 and model 4

Figure 5: Close up of Lenna Image recovered using model 3 (left) and model 4 (centre), with TV result (right) for comparison, notice the reduction in staircasing on the face and shoulder

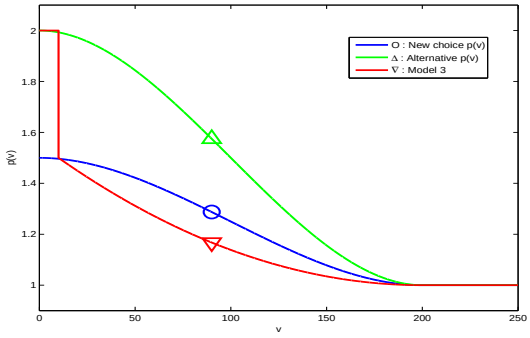

Figure 6: Comparison of the choice of the exponent of a modified TV function

Table 3: Comparison of solution quality (PSNR) and speed (by MG) of Model 3 and the new model

Image	Model 3		The new model	
	$PSNR$	$PSNR_{grad}$	$PSNR$	$PSNR_{grad}$
Hump	41.77	47.26	42.45	48.03
Lenna	28.73	28.31	28.53	28.51
(NLMG)	steps	CPU	steps	CPU
Hump	4	13.1	4	14.4
Lenna	4	13.7	4	14.4

Restored quality comparison. Instead of visualizing the restored images, we compare the $PSNR$ values of the new model with Model 3 in Table 3. The same values of g_2 and α_h are used for both model 3 and the new model. The results presented illustrate the general trend that we have observed; for the hump image both the $PSNR$ and $PSNR_{grad}$ values are higher for the new model than for model 3 while in the case of the Lenna image we can achieve slightly highr $PSNR$ values with model 3 but the $PSNR_{grad}$ values are higher with the new model, on visual inspection Lenna's face also looks a little smoother when using the new model.

Efficiency comparison. In terms of implementation, the new model can be solved similarly to model 3 as they are of the same model type. The cost of 4 nonlinear multigrid steps with 2 pre and 2 post correction smoothing steps is shown in Table 3. The cost of the new model per step is very slightly higher than for model 3, we think this is because it costs slightly less in terms of cpu to evaluate $|\nabla u|^{2-p}$ when $p = 1$ or 2 than it does when $1 < p < 2$ and model 3 takes $p = 2$ when $|\nabla u^*| < g_1$.

In general the advantage of the nonlinear multigrid method over the fixed point method and, in particular, the time marching method is greater for the new model (polynomial from 1.5 to 1) than it is for model 3, the nonlinear multigrid method has been observed to be over twice as fast as the fixed point method and up to 90 times as fast as the time marching, the advantage in the case of the other polynomial (2 to 1) is similar to that observed for model 3.

Remark 4 *We have considered several other choices of p and u^* which include general second and third order polynomials ranging between $2 < q < 1$ at 0 and 1 at sg^*_{max} and a rational similar to that used by [20, 21, 31] but with the threshold for TV regularization built into p, for both $u^* = u_{TV}$ and $u^* = G_\gamma * z$ where $G_\gamma = ce^{-\gamma(x^2+y^2)}$ is a Gaussian used to smooth the noisy image. Typically q should be between 1.75 and 1.5 to give the best results and 0.75 is a suitable choice for γ. For some realistic images $u^* = G_\gamma * z$ gives slightly*

superior results to $u^* = u_{TV}$ in our experience, although u_{TV} is better for the hump image. The nonlinear multigrid method is in all cases we have tested faster than the fixed point and time marching method.

5 Conclusion

We studied several staircasing-reducing regularization methods in this paper. Firstly we compared the efficiency of solving these models by 3 selected iterative solvers and the restored quality, and concluded that Model 3 is the most robust staircasing reduction model. Secondly we proposed a simpler model than Model 3. Numerical tests show that the new model offers better restored quality (in terms of higher PSNR values) and equally efficient solution.

6 Acknowledgements

The authors thank the anonymous referees for making helpful remarks and suggestions. This work is supported by an UK EPSRC DTA award and the Leverhulme Trust RF/9/RFG/2005/0482.

References

[1] Peter Blomgren. *Total Variation Methods for Restoration of Vector Valued Images*. PhD thesis, UCLA, 1998.

[2] Peter Blomgren, Tony Chan, and Pep Mulet. Extensions to total variation denoising. In *SPIE 97*, San Diego, 1997.

[3] William Briggs. *A Multigrid Tutorial*. SIAM, Philadelphia, 1987.

[4] Martin Burger, Stanley Osher, Jinjun Xu, and Guy Gilboa. Nonlinear inverse scale space methods for image restoration. Technical Report 05-34, UCLA, 2005.

[5] Jamylle Carter. *Dual Methods for Total Variation Based Image Restoration*. PhD thesis, UCLA, 2001.

[6] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision.*, 20:89–97, 2004.

[7] Antonin Chambolle and Pierre-Louis Lions. Image recovery via total variation minimization and related problems. *Numer. Math.*, 76:167–188, 1997.

[8] Raymond Chan, Tony Chan, and W. L. Wan. Multigrid for differential-convolution problems arising from image processing. In *Proceedings of the Workshop on Scientific Computing*, 1997.

[9] Tony Chan and Ke Chen. On a nonlinear multigrid algorithm with primal relaxation for the image total variation minimization. *Numer. Algor.*, 2006. To appear.

[10] Tony Chan and Ke Chen. An optimization-based multilevel algorithm for total variation image denoising. *SIAM J. MMS*, 2006. To appear.

[11] Tony Chan and Selim Esedoglu. Aspects of total variation regularized l^1 function approximation. *SIAM J. Appl. Math.*, 65:1817–1837, 2005 (see also CAM04-07).

[12] Tony Chan, Selim Esedoglu, and Frederick Park. Image decomposition combining staircase reduction and texture extraction. Technical Report 05-18, UCLA, 2005.

[13] Tony Chan, Selim Esedoglu, Frederick Park, and A. Yip. Recent developments in total variation image restoration. In *Mathematical Models in Computer Vision: The Handbook*. 2004.

[14] Tony Chan, Gene Golub, and Pep Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM J. Sci. Comput.*, 20:1964–1977, 1999.

[15] Tony Chan, Antonio Marquina, and Pep Mulet. Second order differential functionals in total variation-based image restoration. Technical Report 98-35, UCLA, 1998.

[16] Tony Chan, H. M. Zhou, and Raymond Chan. Continuation method for total variation denoising problems. Technical Report 95-28, UCLA, 1995.

[17] Qianshun Chang and I-Liang Chern. Acceleration methods for total variation based image denoising. *SIAM J. Sci. Comput.*, 25:982–994, 2003.

[18] Ke Chen. *Matrix Preconditioning Techniques and Applications*, volume 19 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, UK, 2005.

[19] Ke Chen and Xue-Cheng Tai. A nonlinear multigrid method for curvature equations related to total variation minimization. Technical Report 05-26, UCLA, 2005.

[20] Yumnei Chen, Stacey Levine, and Murali Rao. Variable exponent, linear growth functionals in image restoration. *SIAM J. Appl. Math. (to appear)*.

[21] Yumnei Chen, Stacey Levine, and Jon Stanich. Image restoration via nonstandard diffusion. Technical report, Duquesne Univ. Dept. of Math and Comp. Sci.

[22] Claudia Frohn-Schauf, Stefan Henn, and Kristian Witsch. Nonlinear multigrid methods for total variation denoising. *Comput. Vision. Sci.*, 7:199–206, 2004.

[23] Van-Emden Henson. Multigrid methods for nonlinear problems: An overview. Technical report, Center for Applied Scientific Computing Lawrence Livermore Laboratory.

[24] Kazufumi Ito and Karl Kunisch. An active-set strategy based on the augmented lagrangian formulation for image restoration. *M2AN Mathematical Modelling and Numerical Analysis*, 33:1–21, 1999.

[25] Kazufumi Ito and Karl Kunisch. BV-type regularization methods for convoluted objects with edge flat and grey scales. *Inverse Problems*, 16:909–928, 2000.

[26] K. Joo and S. Kim. Pde-based image restoration, i: Anti-staircasing and anti-diffusion. Technical report, University of Kentucky, 2003.

[27] Tommi Karkkainen and Kirsi Majava. Nonmonotone and monotone active-set methods for image restoration. *Journal of Optimization Theory and Applications*, 106(1):61–105, 2000.

[28] Tommi Karkkainen and Kirsi Majava. Semi-adaptive optimization methodology for image denoising. *IEE Proc. Vis. Image Signal Process.*, 152(1):553–560, 2005.

[29] Tommi Karkkainen, Kirsi Majava, and Marko Makela. Comparisons of formulations and solution methods for image restoration problems. Technical Report B 14/2000, Department of Mathematical Information Technology University of Jyvaskyla, 2000.

[30] S-H. Lee and J. K. Seo. Noise removal with gauss curvature driven diffusion. *IEEE transactions on Image Processing*, 2005.

[31] Stacey Levine, M. Ramsey, T. Misner, and S. Schwab. An adaptive model for image decomposition. Technical Report 05-01, Duquesne Univ. Dept. of Math and Comp. Sci., 2005.

[32] Marius Lysarker, Andreas Lundervold, and Xue-Cheng Tai. Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time. *IEEE Trans. Image Processing*, 12, 2003.

[33] Marius Lysarker and Xue-Cheng Tai. Interactive image restoration combining total variation minimization and a second order functional. *Int. J. Comp Vision.*

[34] Antonio Marquina and Stanley Osher. Explicit algorithms for a new time dependant model based on level set motion for nonlinear deblurring and noise removal. *SIAM J. Sci. Comput.*, 22:387–405, 2000.

[35] Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin. An iterative regularization method for total variation based image restoration. *Multiscale Model. and Simul.*, 4:460–489, 2005.

[36] Stanley Osher, Andreas Sole, and Luminita Vese. Image decomposition and restoration using total variation minimization and the h^{-1} norm. *Multiscale Model. and Simul.*, 1:349–370, 2003.

[37] Leonard Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[38] Joseph Savage and Ke Chen. An improved and accelerated nonlinear multigrid method for total-variation denoising. *International Journal of Computer Mathematics*, 82:1001–1015, 2005.

[39] Ulrich Trottenberg, Cornelis Oosterlee, and Anton Schuller. *Multigrid*. Academic Press, London, 2001.

[40] Luminita Vese and Stanley Osher. Modelling textures and with total variation minimization and oscillating patterns in image processing. Technical Report 02-19, UCLA, 2002.

[41] Curtis Vogel. A multigrid method for total variation based image denoising. In *Computaion and Control IV*. Birkhauser, 1995.

[42] Curtis Vogel. *Computational Methods for Inverse Problems*. SIAM, Philadelphia, 2002.

[43] Curtis Vogel and Mary Oman. Iterative methods for total variation denoising. *SIAM J. Sci. Comput.*, 17:227–238, 1996.

[44] T. Washio and C. Oosterlee. Krylov subspace acceleration for nonlinear multigrid schemes. *Electronic Transactions on Numerical Analysis.*, 6:271–290, 1997.

[45] P. Wesseling. *An Introduction to Multigrid Methods*. Wiley, Chichester, 1992.

[46] Wotao Yin, Donald Goldfarb, and Stanley Osher. Image cartoon-texture decomposition and feature selection using the total variation regularized l^1 functional. Technical Report 05-47, UCLA, 2005.