

Multigrid Method for the Chan-Vese Model in Variational Segmentation

Noor Badshah and Ke Chen*

Department of Mathematical Sciences, University of Liverpool, Liverpool L69 3BX, UK.

Received 11 July 2007; Accepted (in revised version) 14 September 2007

Available online 18 March 2008

Abstract. The Chan-Vese method of active contours without edges [11] has been used successfully for segmentation of images. As a variational formulation, it involves the solution of a fully nonlinear partial differential equation which is usually solved by using time marching methods with semi-implicit schemes for a parabolic equation; the recent method of additive operator splitting [19, 36] provides an effective acceleration of such schemes for images of moderate size. However to process images of large size, urgent need exists in developing fast multilevel methods. Here we present a multigrid method to solve the Chan-Vese nonlinear elliptic partial differential equation, and demonstrate the fast convergence. We also analyze the smoothing rates of the associated smoothers. Based on our numerical tests, a surprising observation is that our multigrid method is more likely to converge to the global minimizer of the particular non-convex problem than previously unilevel methods which may get stuck at local minimizers. Numerical examples are given to show the expected gain in CPU time and the added advantage of global solutions.

AMS subject classifications: 62H35, 65N22, 65N55, 74G65, 74G75

Key words: Active contours, energy minimization, partial differential equations, segmentation, level sets, total variation, multigrid.

1 Introduction

Image segmentation is a central problem among image processing applications. The aim is to distinguish objects from background and to systematically select specific features out of an image that has many features [2, 10, 22]. For intensity-based images, the

*Corresponding author. *Email addresses:* noor.badshah@liv.ac.uk (N. Badshah), k.chen@liv.ac.uk (K. Chen). <http://www.liv.ac.uk/math>

non-equation-based methods are the popular approaches: threshold techniques, edge-based methods, region-based techniques, and connectivity-preserving relaxation method among others. One may also view the task of distinguishing objects of interest from “the rest” as one to identify the feature boundaries. In recent years, a class of variational formulations offer us the ability to work out features with sharp boundaries — these are the new nonlinear approaches which require more sophisticated solution techniques [10,22].

Let Ω be a bounded open subset of \mathbb{R}^2 with $\partial\Omega$ its boundary and let z be the initially given image, which may be a clean image or contain Gaussian noise. Our aim is to extract a desirable image u which represents features within z — more specifically u is piecewise smooth inside each extracted feature.

The purpose of this paper is to present a working multigrid algorithm for implementing the Chan-Vese variational model [11] and to highlight the algorithm’s practical advantages.

The rest of the paper is organized in the following way: Section 2 first reviews related variational models and then describes the active contour without edges model by Chan and Vese [11], including a discussion of unilevel solution methods of semi-implicit and additive operator splitting. Section 3 first reviews the nonlinear multigrid framework and then describes our choice of smoothers as well as the multigrid algorithm for solving the underlying differential equation [11]. Section 4 gives some local Fourier analysis of the smoothers used, which forms a basis for multigrid convergence. We end the paper in Section 5 with some numerical results and in Section 6 conclusions.

2 The model of active contour without edges and solution methods

Variational segmentation methods aim to find edges (denoted by the index set Γ below) of features in the image z by directly minimizing some objective functional in order to find the piecewise smooth u function separated by Γ . Different methods choose such functionals differently [10,22]. Two early and related methods are the following.

Firstly, the Mumford and Shah segmentation model [23] finds the desired piecewise smooth (so-called cartoon) image u and the edge set Γ from

$$\min_{u,\Gamma} F_1(u,\Gamma) = \alpha \int_{\Omega \setminus \Gamma} |\nabla u|^2 dx dy + \beta \int_{\Gamma} d\sigma + \gamma \int_{\Omega} (u-z)^2 dx dy, \quad (2.1)$$

where α, β, γ are nonnegative constants, the set $\Gamma \subset \Omega$ is also the set of discontinuities, and $\int_{\Gamma} d\sigma$ is the length of Γ . This minimization is clearly stated but is difficult to implement. Various attempts of approximating this formulation exist.

Secondly, the Ambrosio and Tortorelli model [1] finds u and Γ (via a phase quantity p) from

$$\min_{u,p} F_2(u,p) = \alpha \int_{\Omega} p^2 |\nabla u|^2 dx dy + \beta \int_{\Omega} \left(\epsilon |\nabla p|^2 + \frac{(1-p)^2}{4\epsilon} \right) dx dy + \gamma \int_{\Omega} (u-z)^2 dx dy \quad (2.2)$$

based on the Modica and Mortola [20, 21]'s Γ -convergence theory of representing a two dimensional curve Γ (that has 1 inside the curve and 0 outside it) by solving for the phase field function p ($0 \leq p \leq 1$)

$$\min_p L_\epsilon(p) = \int_\Omega \epsilon |\nabla p|^2 + \frac{(1-p)^2}{4\epsilon} dx dy.$$

Here $L_\epsilon(p) \approx \int_\Gamma d\sigma$ from (2.1) when ϵ is small [10, 29]. Clearly model (2.2) appears more amenable than (2.1) to numerical implementation.

It is of interest to mention two other variational models. The snake model of [18] aims to find the segmentation curve C (a parameterized version of Γ with $C(s) : [0, 1] \rightarrow \mathbb{R}^2$) by solving the problem

$$\min_C F_3(C) = \int_0^1 \alpha |C'(s)|^2 + \beta |C''(s)| - \lambda |\nabla z(C(s))|^2 ds. \quad (2.3)$$

The geodesic contour model of [6] proposes to find C by solving

$$\min_C F_4(C) = \int_0^1 |C'(s)| g(|\nabla z(C(s))|) ds, \quad (2.4)$$

where g is an edge detection function e.g. for some $p \geq 1$ and a Gaussian $G_\sigma(x, y)$

$$g(|\nabla z(x, y)|) = \frac{1}{1 + |\nabla G_\sigma(x, y) * z(x, y)|^p}.$$

It should be noted that model (2.4) was solved in [6] by a level-set formulation [25] using time-marching methods.

Below we review the robust model of Chan-Vese [11] which provides a different approximation to the Mumford and Shah model using the idea of level set functions [25], before we discuss its solution methods.

2.1 The Chan-Vese model

Chan and Vese [11] proposed the method of active contour without edges based on (2.1), which differs from previous methods of active contours and snake models in not relying on directly detecting edges (gradients based). The basic idea of the new model is as follows: assume that the given image z is formed by two regions of approximately piecewise constant intensities, of distinct values c_1 and c_2 and that the object to be detected is represented by the region with the value c_1 . Let Γ denote the boundary that separates the two regions. Then the fitting energy is given by:

$$E_1(\Gamma) + E_2(\Gamma) = \int_{\text{inside}(\Gamma)} |z - c_1|^2 dx dy + \int_{\text{outside}(\Gamma)} |z - c_2|^2 dx dy, \quad (2.5)$$

where Γ denotes any curve, and the constants c_1, c_2 , depending on Γ , are the average values of z inside and outside Γ respectively.

Adding a regularization term, Chan and Vese [11] proposed the minimization problem

$$\inf_{c_1, c_2, \Gamma} F(\Gamma, c_1, c_2) \tag{2.6}$$

for the segmentation of image z where

$$F(\Gamma, c_1, c_2) = \mu \text{length}(\Gamma) + \lambda_1 \int_{\text{inside}(\Gamma)} |z - c_1|^2 dx dy + \lambda_2 \int_{\text{outside}(\Gamma)} |z - c_2|^2 dx dy, \tag{2.7}$$

where c_1 and c_2 are the average values of z inside and outside of the variable contour Γ . Here μ, λ_1 and λ_2 are non-negative but fixed parameters to be specified. Both the integrand and the limits of integration in Eq. (2.7) are unknown. To overcome this problem, the level set formulation is used.

The unknown curve Γ can be represented by the zero level set of Lipschitz function $\phi: \Omega \rightarrow \mathbb{R}$ such that

$$\begin{aligned} \Gamma &= \{(x, y) \in \Omega : \phi(x, y) = 0\}, \quad \text{Inside}(\Gamma) = \{(x, y) \in \Omega : \phi(x, y) > 0\}, \\ \text{Outside}(\Gamma) &= \{(x, y) \in \Omega : \phi(x, y) < 0\}, \end{aligned}$$

following [25]. Denoting the Heaviside and the Dirac delta functions (in the sense of distributions) respectively by

$$H(w) = \begin{cases} 1, & \text{if } w \geq 0 \\ 0, & \text{if } w < 0 \end{cases} \quad \text{and} \quad \delta(w) = H'(w),$$

we use ϕ to express the terms in (2.7) in the following way [10]:

$$\begin{aligned} \text{length}\{\phi = 0\} &= \int_{\Omega} |\nabla H(\phi)| = \int_{\Omega} \delta(\phi) |\nabla \phi| dx dy, \\ \int_{\phi \geq 0} |z - c_1|^2 dx &= \int_{\Omega} |z - c_1|^2 H(\phi) dx dy, \\ \int_{\phi < 0} |z - c_2|^2 dx &= \int_{\Omega} |z - c_2|^2 (1 - H(\phi)) dx dy. \end{aligned}$$

Thus Eq. (2.7) becomes

$$\begin{aligned} F(\phi, c_1, c_2) &= \mu \int_{\Omega} \delta(\phi) |\nabla \phi| dx dy + \lambda_1 \int_{\Omega} |z(x, y) - c_1|^2 H(\phi) dx dy \\ &\quad + \lambda_2 \int_{\Omega} |z(x, y) - c_2|^2 (1 - H(\phi)) dx dy. \end{aligned} \tag{2.8}$$

Keeping ϕ fixed and minimizing $F(\phi, c_1, c_2)$ with respect to c_1 and c_2 , we have

$$c_1(\phi) = \frac{\int_{\Omega} z(x, y) H(\phi) dx dy}{\int_{\Omega} H(\phi) dx dy}, \quad c_2(\phi) = \frac{\int_{\Omega} z(x, y) (1 - H(\phi)) dx dy}{\int_{\Omega} (1 - H(\phi)) dx dy} \tag{2.9}$$

if $\int_{\Omega} H(\phi) dx dy > 0$ (i.e., the curve has a nonempty interior in Ω), and $\int_{\Omega} (1 - H(\phi)) dx dy > 0$ (i.e., the curve has a nonempty exterior in Ω).

To compute the Euler-Lagrange equation for the unknown function ϕ , consider slightly regularized versions of H and δ , denoted by H_{ϵ} and δ_{ϵ} , with $\delta_{\epsilon} = H'_{\epsilon}$, and in particular as used in [11] (see also [13])

$$H_{\epsilon}(x) = \frac{1}{2} \left(1 + \frac{2}{\pi} \arctan\left(\frac{x}{\epsilon}\right) \right), \quad \delta_{\epsilon}(x) = H'_{\epsilon}(x) = \frac{\epsilon}{\pi(\epsilon^2 + x^2)}.$$

Denote the regularized functional of $F(\phi, c_1, c_2)$ by $F_{\epsilon}(\phi, c_1, c_2)$, given by

$$F_{\epsilon}(\phi, c_1, c_2) = \mu \int_{\Omega} \delta_{\epsilon}(\phi) |\nabla \phi| + \lambda_1 \int_{\Omega} |z(x, y) - c_1|^2 H_{\epsilon}(\phi) dx dy + \lambda_2 \int_{\Omega} |z(x, y) - c_2|^2 (1 - H_{\epsilon}(\phi)) dx dy. \tag{2.10}$$

Thus the minimization problem (2.6) becomes the following

$$\min_{\phi, c_1, c_2} F_{\epsilon}(\phi, c_1, c_2).$$

After iterating for c_1 and c_2 from (2.9), minimizing F_{ϵ} with respect to ϕ yields the following Euler-Lagrange equation for ϕ :

$$\begin{cases} \delta_{\epsilon}(\phi) \left[\mu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda_1 (z(x, y) - c_1)^2 + \lambda_2 (z(x, y) - c_2)^2 \right] = 0 & \text{in } \Omega, \\ \frac{\delta_{\epsilon}(\phi)}{|\nabla \phi|} \frac{\partial \phi}{\partial \vec{n}} = 0 & \text{on } \partial \Omega, \end{cases} \tag{2.11}$$

where \vec{n} denotes the unit normal exterior to the boundary $\partial \Omega$, and $\partial \phi / \partial \vec{n}$ is the normal derivative of ϕ at boundary [11]. Whenever ϕ is updated from (2.11), the segmented image is generated by

$$u(x, y) = H(\phi(x, y))c_1 + (1 - H(\phi(x, y)))c_2.$$

The above reviewed model divides an image into two regions. It is equally feasible to develop a model for multiple regions [10, 32].

2.2 Semi-Implicit and additive operator splitting methods

Explicitly time-marching schemes have been in wide use for solving variational image models, due to their simplicity. In this subsection we discuss the Semi-Implicit method (SI) and Additive Operator Splitting (AOS) method (which are more stable than the explicit schemes) to solve (2.11) with an artificial time step Δt i.e. the following problem:

$$\begin{cases} \frac{\partial \phi}{\partial t} = \delta_{\epsilon}(\phi) \left[\mu \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda_1 (z - c_1)^2 + \lambda_2 (z - c_2)^2 \right], & \text{in } \Omega, \\ \phi(0, x, y) = \phi_0(x, y) \text{ for } (x, y) \in \Omega \text{ and } \frac{\delta_{\epsilon}(\phi)}{|\nabla \phi|} \frac{\partial \phi}{\partial n} \Big|_{\partial \Omega} = 0, \end{cases} \tag{2.12}$$

where c_1, c_2 are as in (2.9) with H replaced by H_ϵ . Refer to [11, 36]. The AOS method was used in [36] to solve an equation arising from image processing and in [19] to solve a fluid dynamics equation.

Assume Ω is a rectangular domain say $[a, b] \times [c, d]$. For discretization, we use the cell centered finite difference method. Let z be of size $m_1 \times m_2$. Dividing Ω into $m_1 \times m_2$ cells of size $h_1 \times h_2$, and place the grid point at the center of each cell (pixel) with $h_1 = (b - a) / m_1$ and $h_2 = (d - c) / m_2$. The grid point (i, j) is located at

$$(x_i, y_j) = \left(a + (2i - 1)h_1 / 2, c + (2j - 1)h_2 / 2 \right), \quad 1 \leq i \leq m_1, \quad 1 \leq j \leq m_2.$$

Denote by $z_{i,j} = z(x_i, y_j)$.

2.2.1 Semi-implicit scheme

At time $t_n = n\Delta t$, denote by $\phi_{i,j}^n = \phi(t_n, x_i, y_j)$ an approximation of $\phi(t, x, y)$ and similarly

$$\Delta_{\pm}^x \phi_{i,j}^n = \pm(\phi_{i\pm 1,j}^n - \phi_{i,j}^n), \quad \Delta_{\pm}^y \phi_{i,j}^n = \pm(\phi_{i,j\pm 1}^n - \phi_{i,j}^n).$$

Thus we obtain the following linearized equation (through semi-implicitness) [11, 26]

$$\begin{aligned} \frac{\phi_{ij}^{n+1} - \phi_{ij}^n}{\Delta t} = & \delta_\epsilon(\phi_{ij}^n) \left[\frac{\mu}{h_1^2} \Delta_-^x \left(\frac{\Delta_+^x \phi_{ij}^{n+1}}{\sqrt{(\Delta_+^x \phi_{ij}^n / h_1)^2 + ((\phi_{i,j+1}^n - \phi_{i,j-1}^n) / 2h_2)^2}} \right) \right. \\ & + \frac{\mu}{h_2^2} \Delta_-^y \left(\frac{\Delta_+^y \phi_{ij}^{n+1}}{\sqrt{((\phi_{i+1,j}^n - \phi_{i-1,j}^n) / 2h_1)^2 + (\Delta_+^y \phi_{ij}^n / h_2)^2}} \right) \\ & \left. - \lambda_1 (z_{ij} - c_1(\phi^n))^2 + \lambda_2 (z_{ij} - c_2(\phi^n))^2 \right]. \end{aligned}$$

Denoting the coefficients of $\phi_{i+1,j}^{n+1}, \phi_{i-1,j}^{n+1}, \phi_{i,j+1}^{n+1}, \phi_{i,j-1}^{n+1}$ by A_1, A_2, A_3, A_4 respectively, we get the following system of linear equations

$$\begin{aligned} & \left[1 + \delta_\epsilon(\phi_{ij}^n)(A_1 + A_2 + A_3 + A_4) \right] \phi_{ij}^{n+1} \\ = & \phi_{ij}^n + \Delta t \delta_\epsilon(\phi_{ij}^n) \left[(A_1 \phi_{i+1,j}^{n+1} + A_2 \phi_{i-1,j}^{n+1} + A_3 \phi_{i,j+1}^{n+1} + A_4 \phi_{i,j-1}^{n+1}) \right. \\ & \left. - \lambda_1 (z_{ij} - c_1(\phi^n))^2 + \lambda_2 (z_{ij} - c_2(\phi^n))^2 \right], \end{aligned} \tag{2.13}$$

which may be solved by an iterative method [11, 13], as a direct solution can be expensive for images of large size.

2.2.2 Additive operator splitting scheme

The AOS scheme provides an equally accurate (in Δt) and yet more efficient semi-implicit scheme than (2.13) by splitting the 2-dimensional spatial operator into two separate one-dimensional space discretizations [17, 19, 36] and then applying one-dimensional semi-implicit scheme in turns. In linear algebra terms, two tridiagonal systems are solved per

iteration rather than a band-5 (non-diagonal) system. This is in the same spirit of the classical alternating directions implicit (ADI) methods.

3 A nonlinear multi-grid method

The aim of this paper is to present a multigrid algorithm for solving the nonlinear equation (2.11) rather than (2.12). We are not aware of similar work done for segmentation models in the level set formulation. For image restoration, there exist multigrid algorithms [5, 9, 15, 24, 28, 31].

Without t , we shall denote the approximation at (i, j) by $\phi_{i,j} = \phi(x_i, y_j)$ instead of $\phi_{i,j}^n$. Using finite difference scheme to discretize the Euler-Lagrange equation for ϕ , the equation at a grid point (i, j) is given by

$$\delta_\varepsilon(\phi_{i,j}) \left[\mu \left\{ \frac{\Delta_x^-}{h_1} \left(\frac{\Delta_+^x \phi_{i,j} / h_1}{\sqrt{(\Delta_+^x \phi_{i,j} / h_1)^2 + (\Delta_+^y \phi_{i,j} / h_2)^2 + \beta}} \right) + \frac{\Delta_y^-}{h_2} \left(\frac{\Delta_+^y \phi_{i,j} / h_2}{\sqrt{(\Delta_+^x \phi_{i,j} / h_1)^2 + (\Delta_+^y \phi_{i,j} / h_2)^2 + \beta}} \right) \right\} - \lambda_1 (z_{i,j} - c_1)^2 + \lambda_2 (z_{i,j} - c_2)^2 \right] = 0,$$

or

$$\begin{aligned} & \bar{\mu} \left\{ \Delta_x^- \left(\frac{\Delta_+^x \phi_{i,j}}{\sqrt{(\Delta_+^x \phi_{i,j})^2 + (\lambda \Delta_+^y \phi_{i,j})^2 + \bar{\beta}}} \right) + \lambda^2 \Delta_y^- \left(\frac{\Delta_+^y \phi_{i,j}}{\sqrt{(\Delta_+^x \phi_{i,j})^2 + (\lambda \Delta_+^y \phi_{i,j})^2 + \bar{\beta}}} \right) \right\} \\ & = \lambda_1 (z_{i,j} - c_1)^2 - \lambda_2 (z_{i,j} - c_2)^2, \end{aligned} \tag{3.1}$$

where $\bar{\mu} = \mu / h_1$, $\bar{\beta} = h_1^2 \beta$ and $\lambda = h_1 / h_2$, with Neumann's boundary conditions

$$\phi_{i,0} = \phi_{i,1}, \quad \phi_{i,m_2+1} = \phi_{i,m_2}, \quad \phi_{0,j} = \phi_{1,j}, \quad \phi_{m_1+1,j} = \phi_{m_2,j} \tag{3.2}$$

implying that the finite difference terms involving indexes outside the range $i = 1, \dots, m_1; j = 1, \dots, m_2$ are zero.

Here the first term in Eq. (3.1) resembles the denoising model by [26] using the total variation (TV) regularization. The parameter β should be a small quantity to avoid the gradient becoming 0 as in [26, 33, 34].

3.1 The full approximation scheme

We first give a brief discussion of the 3 main ingredients of a nonlinear multigrid method (MG) called the full approximation scheme (FAS) [9, 13, 16, 30] due to Brandt [3], and then concentrate on our choice of smoothers. Denote the system of non-linear equations described by Eqs. (3.1) and (3.2) by

$$N^h \phi^h = f^h, \tag{3.3}$$

where ϕ^h and f^h are grid functions on a $m_1 \times m_2$ cell centered rectangular grid Ω^h with spacing (h_1, h_2) . Let Ω^{2h} denote the $m_1/2 \times m_2/2$ cell centered grid which results from standard coarsening of Ω^h .

Let $e^h = \phi^h - \Phi^h$ be the algebraic error and $r^h = f^h - N^h \Phi^h$ be the residual, where Φ^h is a good approximation to solution of (3.3) in the sense that e^h is smooth. Such smoothness can only be archived by a careful choice of suitable smoothers — a major task in developing a working multigrid method. Then the non-linear residual equation will be:

$$N^h(\Phi^h + e^h) - N^h \Phi^h = r^h. \tag{3.4}$$

If e^h is smooth, it can be well approximated on Ω^{2h} . Therefore any iterative method which smooths the error on the fine grid can be improved by the use of the coarse grid correction, in which a coarse grid analogue of the residual equation is solved (solution on a coarse grid is less expensive than on a fine grid) to obtain a coarse grid approximation of the error, which is then transferred back to the fine grid to correct the approximation Φ^h . This is known as a two-grid cycle, and with recursive application it can be extended to a multigrid method. Let us define the restriction and interpolation operators for transferring grid functions between Ω^h and Ω^{2h} for cell-centered discretization:

Restriction

$$I_h^{2h} \Phi^h = \Phi^{2h}, \tag{3.5}$$

where for $1 \leq i \leq m_1/2, 1 \leq j \leq m_2/2$:

$$\Phi_{i,j}^{2h} = \frac{1}{4}(\Phi_{2i-1,2j-1}^h + \Phi_{2i-1,2j}^h + \Phi_{2i,2j-1}^h + \Phi_{2i,2j}^h)$$

is a full weighting operator [13,30].

Interpolation

$$I_{2h}^h \Phi^{2h} = \Phi^h, \tag{3.6}$$

where for $1 \leq i \leq m_1/2, 1 \leq j \leq m_2/2$:

$$\begin{aligned} \Phi_{2i,2j} &= \frac{1}{16}(9\Phi_{i,j}^{2h} + 3\Phi_{i+1,j}^{2h} + 3\Phi_{i,j+1}^{2h} + \Phi_{i+1,j+1}^{2h}), \\ \Phi_{2i-1,2j} &= \frac{1}{16}(9\Phi_{i,j}^{2h} + 3\Phi_{i-1,j}^{2h} + 3\Phi_{i,j+1}^{2h} + \Phi_{i-1,j+1}^{2h}), \\ \Phi_{2i,2j-1} &= \frac{1}{16}(9\Phi_{i,j}^{2h} + 3\Phi_{i+1,j}^{2h} + 3\Phi_{i,j-1}^{2h} + \Phi_{i+1,j-1}^{2h}), \\ \Phi_{2i-1,2j-1} &= \frac{1}{16}(9\Phi_{i,j}^{2h} + 3\Phi_{i-1,j}^{2h} + 3\Phi_{i,j-1}^{2h} + \Phi_{i-1,j-1}^{2h}) \end{aligned}$$

is known as a bilinear interpolation operator [13,30].

It remains to discuss the most important ingredient of a MG: smoothing. We will first present a local nonlinear smoother and then review the smoother introduced by [28] (for a TV denoising model [26]) with a view to comparison. Other smoothers are mentioned next.

3.2 Smoother I

In this method the system of nonlinear equations is linearized locally, by using an idea related to fixed-point iterations (see [33]) or computing $D(\phi)$ on each grid (i, j) locally. However this method is different from that of fixed-point iterations [33] which is a global linearization method (refer to Smoother II below). Then we solve a linear equation to update the local quantity ϕ_{ij} and this is repeated for a small number of smoothing steps. Overall we use the Gauss-Seidel idea to smooth the error over all pixels. In detail, Eq. (3.1) can be written as

$$\begin{aligned} & \bar{\mu} \left\{ \left[\frac{\Delta_+^x \phi_{i,j}}{\sqrt{(\Delta_+^x \phi_{i,j})^2 + (\lambda \Delta_+^y \phi_{i,j})^2 + \bar{\beta}}} - \frac{\Delta_+^x \phi_{i-1,j}}{\sqrt{(\Delta_+^x \phi_{i-1,j})^2 + (\lambda \Delta_+^y \phi_{i-1,j})^2 + \bar{\beta}}} \right] \right. \\ & \left. + \lambda^2 \left[\frac{\Delta_+^y \phi_{i,j}}{\sqrt{(\Delta_+^x \phi_{i,j})^2 + (\lambda \Delta_+^y \phi_{i,j})^2 + \bar{\beta}}} - \frac{\Delta_+^y \phi_{i,j-1}}{\sqrt{(\Delta_+^x \phi_{i,j-1})^2 + (\lambda \Delta_+^y \phi_{i,j-1})^2 + \bar{\beta}}} \right] \right\} \\ & = \lambda_1 (z_{i,j} - c_1)^2 - \lambda_2 (z_{i,j} - c_2)^2. \end{aligned}$$

Let the coefficients (intended below to be frozen in local linearization) be denoted by

$$\begin{aligned} D(\phi)_{i,j} &= \frac{1}{\sqrt{(\Delta_+^x \phi_{i,j})^2 + (\lambda \Delta_+^y \phi_{i,j})^2 + \bar{\beta}}}, \\ D(\phi)_{i-1,j} &= \frac{1}{\sqrt{(\Delta_+^x \phi_{i-1,j})^2 + (\lambda \Delta_+^y \phi_{i-1,j})^2 + \bar{\beta}}}, \\ D(\phi)_{i,j-1} &= \frac{1}{\sqrt{(\Delta_+^x \phi_{i,j-1})^2 + (\lambda \Delta_+^y \phi_{i,j-1})^2 + \bar{\beta}}}. \end{aligned}$$

So we have

$$\begin{aligned} & \bar{\mu} \left\{ \left[D(\phi)_{i,j} (\phi_{i+1,j} - \phi_{i,j}) - D(\phi)_{i-1,j} (\phi_{i,j} - \phi_{i-1,j}) \right] + \lambda^2 \left[D(\phi)_{i,j} (\phi_{i,j+1} - \phi_{i,j}) \right. \right. \\ & \left. \left. - D(\phi)_{i,j-1} (\phi_{i,j} - \phi_{i,j-1}) \right] \right\} = \lambda_1 (z_{i,j} - c_1)^2 - \lambda_2 (z_{i,j} - c_2)^2. \end{aligned} \quad (3.7)$$

Let $f_{i,j}$ denote the right hand side of (3.7) and $\tilde{\phi}$ be the approximation to ϕ at a previous iteration. Then Eq. (3.7), now having only one local unknown $\phi_{i,j}$ (shown below in bold face for clarity), becomes the following linear equation:

$$\begin{aligned} & \left[D(\tilde{\phi})_{i,j} (\tilde{\phi}_{i+1,j} - \boldsymbol{\phi}_{i,j}) - D(\tilde{\phi})_{i-1,j} (\boldsymbol{\phi}_{i,j} - \tilde{\phi}_{i-1,j}) \right] \\ & + \lambda^2 \left[D(\tilde{\phi})_{i,j} (\tilde{\phi}_{i,j+1} - \boldsymbol{\phi}_{i,j}) - D(\tilde{\phi})_{i,j-1} (\boldsymbol{\phi}_{i,j} - \tilde{\phi}_{i,j-1}) \right] = f_{i,j} / \bar{\mu} \equiv \bar{f}_{i,j}. \end{aligned} \quad (3.8)$$

Our proposed algorithm solves this equation for $\boldsymbol{\phi}_{i,j}$ to update $\tilde{\phi}_{i,j}$ which leads to updating the coefficients and further iterations (before moving to the next pixel in a Gauss-Seidel fashion):

Algorithm 3.1: Algorithm for Smoother I. $\Phi^h \leftarrow \text{Smoother1}(\Phi^h, \bar{f}^h, \text{maxit}, \text{tol})$

```

for  $i=1:m_1$ 
  for  $j=1:m_2$ 
    for  $\text{iter}=1:\text{maxit}$ 
       $\tilde{\Phi}^h \leftarrow \Phi^h$ 

      
$$\Phi_{i,j} = \frac{\left[ \left\{ D(\tilde{\Phi}^h)_{i,j} \tilde{\Phi}_{i+1,j}^h + D(\tilde{\Phi}^h)_{i-1,j} \tilde{\Phi}_{i-1,j}^h + \lambda^2 D(\tilde{\Phi}^h)_{i,j} \tilde{\Phi}_{i,j+1}^h + \lambda^2 D(\tilde{\Phi}^h)_{i,j-1} \tilde{\Phi}_{i,j-1}^h \right\} - \bar{f}_{i,j} \right]}{D(\tilde{\Phi}^h)_{i,j} + D(\tilde{\Phi}^h)_{i-1,j} + \lambda^2 (D(\tilde{\Phi}^h)_{i,j} + D(\tilde{\Phi}^h)_{i,j-1})}$$


      if  $|\Phi_{i,j} - \tilde{\Phi}_{i,j}^h| < \text{tol}$  Stop for  $(i,j)$ 
      end
    end
  end
end

```

3.3 Smoother II

As our main equation has a TV operator, it is natural to consider the fixed-point method [33, 34]. We use the smoother proposed in [28] for the image TV denoising model, where the resulting system of linear equations is solved by the Gauss-Seidel relaxation method for a fixed (small) number of smoothing steps (instead of exact solve). Here the system of non-linear equations is linearized *globally* at each step by computing $D(\phi)$ at all points (i,j) , which differs from Smoother I. The algorithm proceeds as follows:

Algorithm 3.2: Algorithm for Smoother II. $\Phi^h \leftarrow \text{Smoother2}(\Phi^h, \bar{f}^h, \text{maxit}, \text{tol})$

```

for  $i=1:m_1$ 
  for  $j=1:m_2$ 
    
$$D(\Phi^h)_{i,j} = [(\Delta_+^x \Phi_{i,j})^2 + (\lambda \Delta_+^y \Phi_{i,j})^2 + \beta]^{-\frac{1}{2}}$$

  end
end
 $\varphi^h = \Phi^h$ 
for  $\text{iter}=1:\text{maxit}$ 
  for  $i=1:m_1$ 
    for  $j=1:m_2$ 
       $\tilde{\varphi}^h \leftarrow \varphi^h$ 

      
$$\varphi_{i,j} = \frac{\left[ \left\{ D(\Phi^h)_{i,j} \tilde{\varphi}_{i+1,j}^h + D(\Phi^h)_{i-1,j} \tilde{\varphi}_{i-1,j}^h + \lambda^2 D(\Phi^h)_{i,j} \tilde{\varphi}_{i,j+1}^h + \lambda^2 D(\Phi^h)_{i,j-1} \tilde{\varphi}_{i,j-1}^h \right\} - \bar{f}_{i,j} \right]}{D(\Phi^h)_{i,j} + D(\Phi^h)_{i-1,j} + \lambda^2 (D(\Phi^h)_{i,j} + D(\Phi^h)_{i,j-1})}$$

    end
  end
end
 $\Phi^h \leftarrow \varphi$ 

```

Here the coefficients are updated at the start of each smoothing step globally and are stored for relaxation use.

Remark 3.1. The above two smoothers are both fixed-point based. Then one may consider two related ideas: (i) Newton methods — which are found not to perform satisfactorily for this problem and also previously for the TV denoising problem [28]. (ii) line relaxation methods — which are found to work well but the improvements over Smoothers I and II are marginal (of course line relaxations are slightly more expensive to implement).

Yet there exist other smoothing ideas in the literature, e.g., the energy-minimizing smoothers of [35], the primal-dual smoother [8] and algebraic multigrid ideas [12, 14, 27] which remain to be tested for segmentation problems.

3.4 The multi-grid algorithm

To solve Eq. (3.3), our FAS multilevel algorithm may be summarized as follows, see, e.g., [13, 28, 30] and [9]:

Algorithm 3.3: Multigrid Algorithm

We use these multigrid parameters:

ν_1 pre-smoothing steps on each level

ν_2 post-smoothing steps on each level

ℓ the number of multigrid cycles on each level ($\ell=1$ for V-cycling and $\ell=2$ for W-cycling). Here we take the V-cycle with $\ell=1$.

FAS Multigrid Cycle

$$\Phi^h \leftarrow \text{FASCYC}(\Phi^h, \bar{f}^h, \text{iter}, \nu_1, \nu_2, \ell)$$

1. If Ω^h is the coarsest grid, then solve Eq. (3.3) using time marching technique of [36] and then stop.

Else implement a smoother, i.e.,

$$\Phi^h \leftarrow \text{Smoother}^{\nu_1}(\Phi^h, \bar{f}^h, \nu_1). \quad (\text{Pre-Smoothing})$$

2. **Restriction:**

$$\begin{aligned} \Phi^{2h} &= I_h^{2h} \Phi^h, \quad \bar{\Phi}^{2h} = \Phi^{2h}, \\ \bar{f}^{2h} &= I_h^{2h} (\bar{f}^h - N^h \Phi^h) + N^{2h} \Phi^{2h}, \\ \Phi^{2h} &\leftarrow \text{FASCYC}_\ell^{2h}(\Phi^{2h}, \bar{f}^{2h}, \text{iter}, \nu_1, \nu_2, \ell). \end{aligned}$$

3. **Interpolation**

$$\Phi^h \leftarrow \Phi^h + I_{2h}^h (\Phi^{2h} - \bar{\Phi}^{2h})$$

- 4.

$$\Phi^h \leftarrow \text{Smoother}^{\nu_2}(\Phi^h, \bar{f}^h, \nu_2). \quad (\text{Post-Smoothing})$$

4 Local Fourier analysis of smoothers

The standard FAS multilevel algorithm (such as Algorithm 3.3) does not automatically converge for many problems, if simple smoothers are used (i.e. Gauss-Seidel for linear problems and Gauss-Seidel-Newton for nonlinear problems). The key for convergence lies in effective smoothers or reduction of residuals to a smoothed form (where high frequency components are small regardless of the overall error [13, 30]). Here we show some local Fourier analysis (LFA) results to suggest that our smoothers are effective.

It should be remarked that LFA is in general not applicable to nonlinear smoothers. Here for our linearized smoothers, the analysis can only be done for each individual smoothing iteration and the obtained smoothing rates change from iteration to iteration. However we look for general trends e.g. if the three consecutive smoothing rates are 0.59,0.61,0.44 (instead of a constant rate say 0.5), we say the underlying smoother is effective. Likewise consecutive rates such as 1.2,0.89,0.99 may indicate a poor smoother.

For simplicity, we consider the case of a square image $m = m_1 = m_2$ so $\lambda^2 = 1$. Denote $h = h_1 = h_2$. Following (3.8), a typical grid equation on Ω^h after linearization is

$$-(g_1 + 2g_2 + g_3)\phi_{i,j} + g_1\phi_{i-1,j} + g_3\phi_{i,j-1} + g_2(\phi_{i,j+1} + \phi_{i,j+1}) = \bar{f}_{i,j}, \tag{4.1}$$

where g_1, g_2, g_3 are based a previous iteration and to be considered as local constants. Although both smoothers I and II may be written as

$$-(g_1 + 2g_2 + g_3)\phi_{i,j}^{(k+1)} + g_1\phi_{i-1,j}^{(k+1)} + g_3\phi_{i,j-1}^{(k+1)} + g_2(\phi_{i,j+1}^{(k)} + \phi_{i+1,j}^{(k)}) = \bar{f}_{i,j}, \tag{4.2}$$

the choices of g_1, g_2, g_3 are quite different. For Smoother I, we take

$$g_1 = D(\tilde{\phi})_{i-1,j} = D(\phi^{(k)})_{i-1,j}, \quad g_2 = D(\tilde{\phi})_{i,j} = D(\phi^{(k)})_{i,j}, \quad g_3 = D(\tilde{\phi})_{i,j-1} = D(\phi^{(k)})_{i,j-1},$$

where $\phi^{(k)}$ is the previous iterate at step k (local fixed-point). For Smoother II, we take

$$g_1 = D(\tilde{\Phi})_{i-1,j}, \quad g_2 = D(\tilde{\Phi})_{i,j}, \quad g_3 = D(\tilde{\Phi})_{i,j-1},$$

where $\tilde{\Phi}$ is the iterate at the previous sweep (global fixed-point).

Recall that the local Fourier analysis (LFA) measures the largest amplification factor in a relaxation scheme [3, 13, 30]. Let the general Fourier component be

$$B_{\theta_1, \theta_2}(x_i, y_j) = \exp\left(\mathbf{i}\alpha_1 \frac{x_i}{h} + \mathbf{i}\alpha_2 \frac{y_j}{h}\right) = \exp\left(\frac{2\mathbf{i}\theta_1 i \pi}{m} + \frac{2\mathbf{i}\theta_2 j \pi}{m}\right)$$

and define the local error functions by

$$e_{i,j}^{(k+1)} = \phi_{i,j} - \phi_{i,j}^{(k+1)}, \quad e_{i,j}^{(k)} = \phi_{i,j} - \phi_{i,j}^{(k)}.$$

Here $\alpha_1 = 2\theta_1\pi/m, \alpha_2 = 2\theta_2\pi/m \in [-\pi, \pi]$. Then LFA involves expanding

$$e^{(k+1)} = \sum_{\theta_1, \theta_2 = -m/2}^{m/2} \psi_{\theta_1, \theta_2}^{(k+1)} B_{\theta_1, \theta_2}(x_i, y_j), \quad e^{(k)} = \sum_{\theta_1, \theta_2 = -m/2}^{m/2} \psi_{\theta_1, \theta_2}^{(k)} B_{\theta_1, \theta_2}(x_i, y_j) \quad (4.3)$$

in Fourier components. We shall estimate the maximum ratio

$$\bar{\mu} = \max_{\theta_1, \theta_2} \mu(\theta_1, \theta_2) = |\psi_{\theta_1, \theta_2}^{(k+1)} / \psi_{\theta_1, \theta_2}^{(k)}|$$

in the high frequency range $(\alpha_1, \alpha_2) \in [-\pi, \pi] \setminus [-\pi/2, \pi/2]$ which defines the smoothing rate [30].

Smoothers I and II. From (4.1) and (4.2), we obtain

$$-(g_1 + 2g_2 + g_3)e_{i,j}^{(k+1)} + g_1e_{i-1,j}^{(k+1)} + g_3e_{i,j-1}^{(k+1)} + g_2(e_{i,j+1}^{(k)} + e_{i,j+1}^{(k)}) = 0,$$

which defines the ratio (note $\alpha_j = 2\theta_j\pi/m$)

$$\mu(\theta_1, \theta_2) = \frac{|g_2(e^{i\alpha_1} + e^{i\alpha_2})|}{|g_1 + 2g_2 + g_3 - g_1e^{-i\alpha_1} - g_3e^{-i\alpha_2}|}. \quad (4.4)$$

To proceed with an analysis, we have to compute g_1, g_2 and g_3 or function

$$D(\phi) = \sqrt{(\Delta_+^x \phi)^2 + (\Delta_+^y \phi)^2 + \bar{\beta}}$$

numerically and work out the smoothing factor $\bar{\mu}$ for each set of coefficients g_1, g_2 and g_3 within a smoother. In our earlier work, we select a special set of such coefficients. Below we display the maximum of such factors

$$\hat{\mu} = \max_{g_1, g_2, g_3} \bar{\mu} = \max_{g_1, g_2, g_3} \max_{\theta_1, \theta_2} \mu(\theta_1, \theta_2). \quad (4.5)$$

As such a linear analysis is based on freezing the nonlinear coefficients, our results should be viewed only as a guide to smoothers' effectiveness and a way to distinguish smoothers.

Taking the test example of Problem 1 from Fig. 1 with $m = 32$, we can display $\hat{\mu}$ in the first 4 cycles of our MG algorithm as in Table 1 where Pre-1 refers to the case of "pre-smoothing" and Post-1 to "post-smoothing" etc. If we instead consider the average rate from all pixels, the averages are respectively 0.49 and 0.71 for Smoothers I and II. Clearly in this example Smoother I appears to be more effective than Smoother II in terms of rates. Such a claim will be tested in the next section.

Table 1: $\hat{\mu}$ in the first 4 cycles of our MG algorithm.

MG cycle	Smoothing Steps	Rate I: $\hat{\mu}^I$	Rate II: $\hat{\mu}^{II}$
1	Pre-1	0.4942	0.6776
	Pre-2	0.4941	0.9317
	Post-1	0.4942	0.9135
	Post-2	0.4942	0.9427
2	Pre-1	0.6003	0.9561
	Pre-2	0.6003	0.9174
	Post-1	0.6003	0.9581
	Post-2	0.6003	0.9577
3	Pre-1	0.7760	0.9533
	Pre-2	0.7760	0.9193
	Post-1	0.7757	0.9092
	Post-2	0.7749	0.9040
4	Pre-1	0.6025	0.9594
	Pre-2	0.6026	0.9456
	Post-1	0.6026	0.9286
	Post-2	0.6026	0.9678

5 Numerical results

In this section, we shall discuss several aspects of our MG (Algorithm 3.3). Firstly, we hope to demonstrate that our MG (with smoother I) can scale well with increasing sizes of an image and also that our MG is not sensitive to the choice of the initial guess for ϕ . The latter claim is further supported by incorporating the full multigrid idea (FMG) [30] and then our MG with FMG does not need an initial guess for ϕ . Secondly, we compare the performance of Algorithm 3.3 with smoother I and II as well as alternative smoothers. Thirdly, we demonstrate that an improved solution of global minimizer can be obtained using the multigrid method over previous methods. This by-product of MG is slightly surprising as it is not expected. Finally we will compare our MG with AOS and SI methods to illustrate the expected CPU saving.

We shall test on 4 images (mainly with $m = 256$ unless stated otherwise) as shown in Fig. 1 with Problem 3 used in [4] for discussing the global minimization issue. We shall take the following parameters

$$\mu = 0.01 \times (256)^2, \quad \lambda_1 = \lambda_2 = 1, \quad \beta = 10^{-6}, \quad \epsilon = 10^{-10}$$

and the initial guess is

$$\phi = -\sqrt{(x-150)^2 + (y-140)^2} + 100 \quad \text{for } m = 256.$$

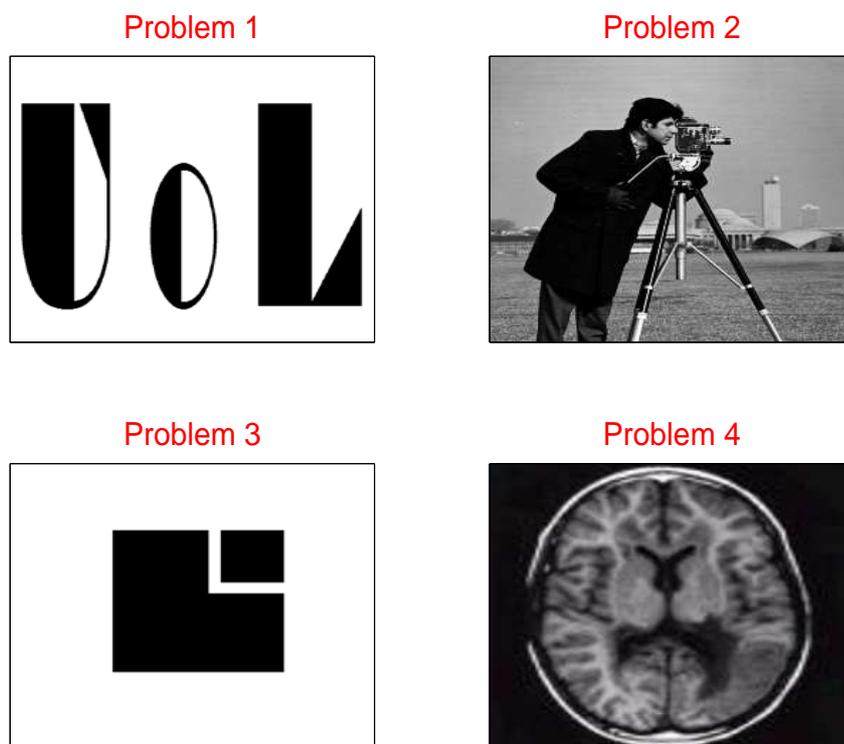


Figure 1: Segmentation Test Images.

5.1 Convergence tests and full multigrid grid

Our preliminary results suggest that our MG (with Smoother I) can converge for varying choices of the initial ϕ e.g. specifying ϕ having a small circle at a fixed position. Fig. 2 (for Problem 4) shows how the segmented image is obtained from the initial guess for ϕ (first plot) and then the results from cycles 1,2,3 (second to the last plot). Here only 3 MG cycles are required to drive the relative residual (in vector 2-norm) to below 10^{-6} .

However to eliminate the need of an initial guess, we consider the use of a full multigrid idea [30] which starts the solution of (3.1) on the coarsest grid. Then each solution is interpolated onto the next fine grid to give an initial guess until we reach the finest grid where we start the MG algorithm. This is shown in Fig. 3, where the first plot shown the position of the starting ϕ on the coarsest 4×4 level, the second plot shows the initial guess derived by FMG on the finest level and only 2 full MG cycles are needed to reach the segmented result in the bottom plot.

To test on the scalability of MG, we display in Table 2 the number of steps needed to reach a desirable accuracy for Problems 2–3 (with μ fixed as before). The CPU times are obtained from running Matlab 7 on a Pentium PC for illustration purpose. Here FMG means that the initial guess of ϕ is obtained from a FMG method following by the

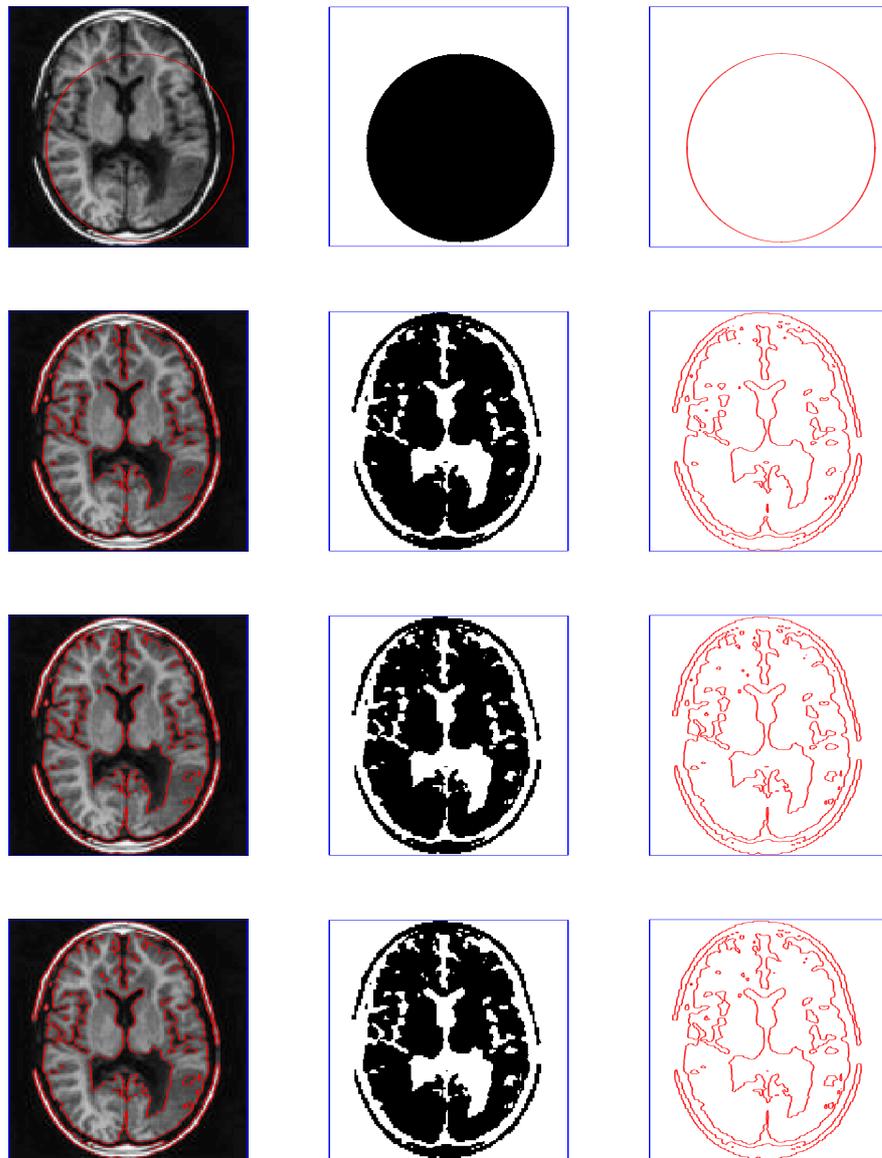


Figure 2: Solution of Problem 4 by MG. Left: z with $\phi=0$, Middle: $\phi < 0$ and Right: Contour plot of $\phi=0$. The top to bottom plots are from MG cycles 0,1,2,3.

normal MG cycles. Clearly MG (with smoother I) is efficient. Our MG works the best with $\nu_1 = \nu_2 = 2$, but it will not work well if much less smoothing steps are used e.g. $\nu_1 = 1, \nu_2 = 0$.

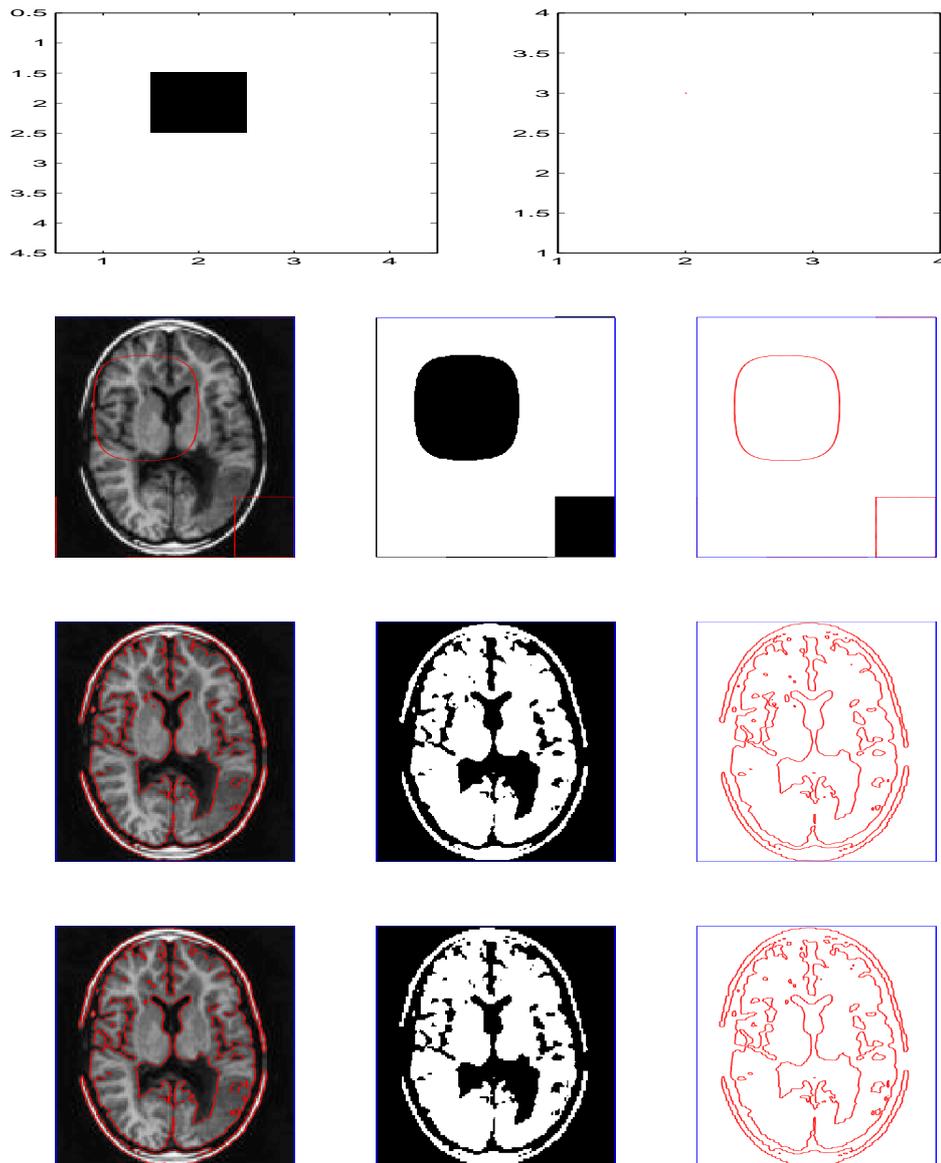


Figure 3: Solution of Problem 4 by FMG and MG. Left: z with $\phi=0$, Middle: $\phi < 0$ and Right: Contour plot of $\phi=0$. The top to bottom plots are from the FMG initial guess on the coarsest 4×4 level (top left: $\phi < 0$ and top right: contour plot of $\phi=0$), the FMG result and MG cycles 1, 2.

5.2 Algorithm 3.3 with different smoothers

We have conducted several tests of our MG with different smoothers:

- (i) smoother I (our preferred choice);

Table 2: Test of scalability of MG and FMG.

Problem	Method	Image Size m (for $m \times m$)	MG cycles	CPU
2	MG	128	4	9
		256	5	15
		512	5	27
		1024	5	90
	FMG	128	3	15
		256	3	17
		512	3	27
		1024	3	83
3	MG	128	2	5
		256	2	6
		512	2	12
		1024	2	50
	FMG	128	2	11
		256	2	13
		512	2	21
		1024	2	61

- (ii) smoother II (previously found suitable for a different problem);
- (iii) line relaxation smoothers [30];
- (iv) a Gauss-Seidel-Newton smoother.

Omitting the computational results, we remark that these observations can be made:

- a) Our preferred smoother I is (as predicted by the analysis in the previous section) is up to twice as fast as Smoother II as I requires less MG cycles and less CPU time.
- b) Line smoothers lead to less MG cycles but more CPU time than I.
- c) The Gauss-Seidel-Newton smoother does not lead to convergence of the MG (as also known for a different problem [28]).

As mentioned in Remark 3.1, there are other potentially useful smoothers that might be considered for the segmentation model which can be pursued in the near future.

5.3 Improved solution of global minimizers

Here we will give some evidence of obtaining improved solution of global minimizer. X. Bresson et al. [4] discussed the drawback of variational segmentation model, the main one being the existence of local minima in the energy. In non-convex optimization, the local minima often lead to unsatisfactory results. The solution of Problem 3 (see Fig.1 as in [4]) is displayed in Fig. 4, where both the SI and the AOS method get stuck in a

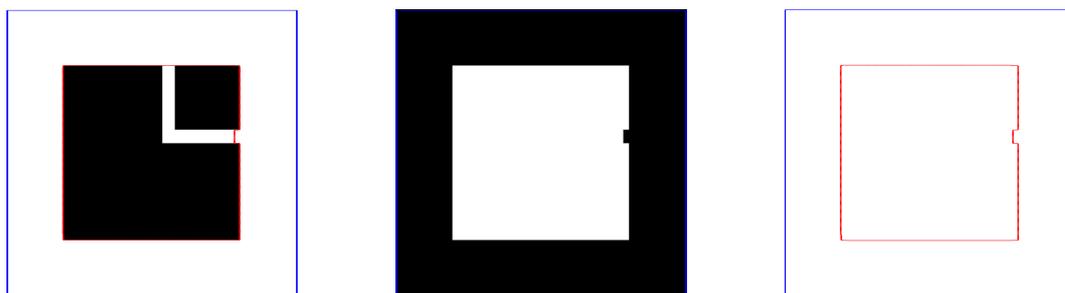


Figure 4: The incorrect solution of Problem 3 by AOS (or SI) method. Left: z with $\phi=0$, Middle: $\phi < 0$ and Right: Contour plot of $\phi=0$.

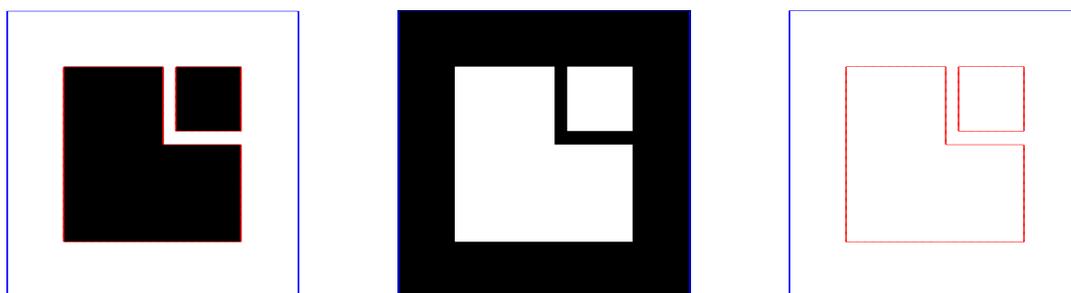


Figure 5: The correct solution of Problem 3 by our MG method. Left: z with $\phi=0$, Middle: $\phi < 0$ and Right: Contour plot of $\phi=0$.

local minimum with $F(\phi, c_1, c_2) = 1.44 \times 10^8$. We then use our multigrid method and the image has been segmented correctly as shown in Fig. 5. Clearly our multigrid method manages to get closer to the global minimizer with $F(\phi, c_1, c_2) = 3.94 \times 10^4$. The same comparison is made on Problem 2, where the result from the SI and the AOS method is shown in Fig. 6 with $F(\phi, c_1, c_2) = 4.65 \times 10^7$ and the result from our MG shown in Fig. 7 with a smaller $F(\phi, c_1, c_2) = 4.05 \times 10^7$. This is not coincidental as we have also tested many other examples and observed the same results. We have not yet found a reason for the unexpected success regarding global minimizer but we believe that this must be due to our use of coarse levels on which fine level local minimizers are not represented.

5.4 Comparison of complexity and CPU

Finally we compare the speed of MG (with smoother I) with SI and AOS methods.

Before numerical results, it is of interest to estimate the computational complexity of the algorithms involved. To be concrete, we assume that the inner solver of a SI method is by a conjugate gradient method for 25 steps, and in the MG method $\nu_1=2$ and $\nu_2=1$. Then consider segmenting some image of size $m \times m$. The setup cost for the 4 main coefficients is about $4 \times 8m^2 = 32m^2$ operations for all methods per step. The cost of each step of SI method is thus $W_1 = 25 \times 5m^2 + 32m^2 = 157m^2$. For the AOS, each trisolve costs $4m^2$



Figure 6: Solution of Problem 2 by AOS (or SI) method. Left: z with $\phi=0$, Middle: $\phi<0$ and Right: Contour plot of $\phi=0$.



Figure 7: Solution of Problem 2 by our MG method. Left: z with $\phi=0$, Middle: $\phi<0$ and Right: Contour plot of $\phi=0$.

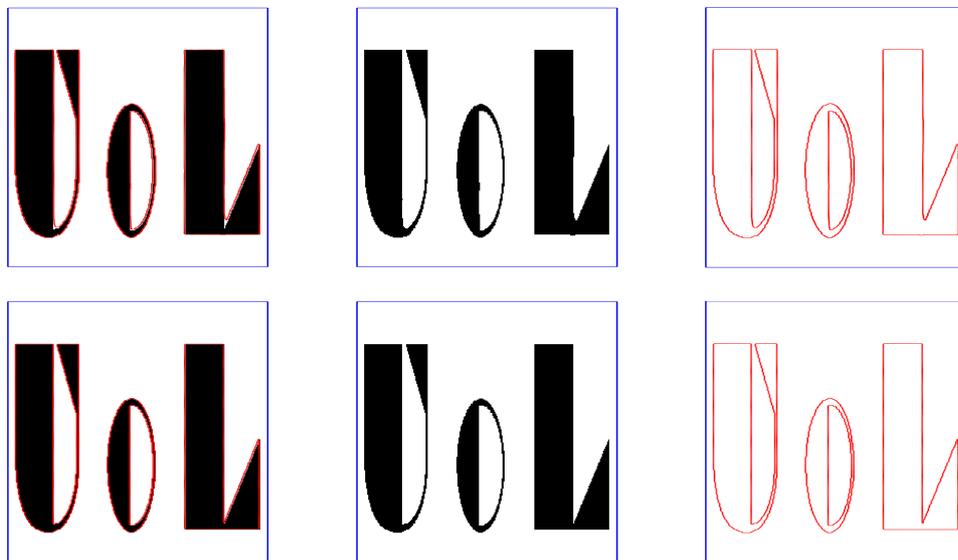


Figure 8: Solution of Problem 1 by our MG method. Left: z with $\phi=0$, Middle: $\phi<0$ and Right: Contour plot of $\phi=0$. Top: MG cycle 1 and bottom: MG cycle 3.

Table 3: Comparison of MG with SI and AOS methods.

Image size	SI method		AOS Method		MG	
	Itr	CPU	Itr	CPU	Itr	CPU
128×128	30	29	35	14	2	9
256×256	85	511	80	138	3	20
512×512	–	–	5000	3.6×10^4	3	41
1024×1024	–	–	–	–	3	165

operations so the cost of each step is $W_2 = 2 \times 4m^2 + 32m^2 = 40m^2$. Finally the finest level smoothing cost for MG is $11m^2 \times (v_1 + v_2) = 33m^2$ so the finest level cost is $33m^2 + 32m^2 = 65m^2$. The cost per MG step over all levels is $W_3 = 4/3 \times 65m^2 \approx 87m^2$. Therefore the practical efficiency of these methods, although all of $\mathcal{O}(N) = \mathcal{O}(m^2)$ complexity per step, will depend on the number of actual iteration steps used for achieving the same accuracy.

We take Problem 1 as an example to illustrate the results, where the segmented results for $m = 256$ from MG cycles 1,3 are shown in Fig. 8. In Table 3 we compare the number of iteration (Itr) and the CPU time for various m . From Table 3 we see that the SI method and the AOS method are adequate for images of moderate size, but our MG method is much more efficient for large images, where SI and AOS are either too slow or fail to deliver a result within 24 hours (marked as entry '–').

6 Conclusions

We have proposed an effective smoother for a nonlinear multigrid method to solve the Chan-Vese active contour without edges model. A linear Fourier analysis shows that our local smoother I is better than the global smoother II. For large images, as expected, our MG leads to much faster solutions than unilevel methods of SI and AOS. A somewhat surprising observation is that our MG can reach closer to the global minimizer than SI and AOS methods in all of our test cases. Future work will address multigrid methods for other variational formulations [10] and alternative multilevel methods [7].

Acknowledgments

NB thanks UET, Peshawar, Pakistan for funding a scholarship to support this work.

References

- [1] L. Ambrosio and V. M. Tortorelli, Approximation of functionals depending on jumps by elliptic functionals via Γ -convergence, *Comm. Pure Appl. Math.*, 43 (1990), 999-1036.

- [2] T. Asano, D. Z. Chen, N. Katoh and T. Tokuyama, Polynomial-time solutions to image segmentation, Proc. of the 7th Ann. SIAM-ACM Conference on Discrete Algorithms, 1996, 104-113.
- [3] A. Brandt, Multi-level adaptive solutions to BVPs, Math. Comp., 31 (1977) 333-390.
- [4] X. Bresson, S. Esedoglu, P. Vanderghenst, J.P. Thiran and S. Osher, Global minimizers of the active contour/snake model, UCLA CAM Report 05-04, 2005.
- [5] A. Bruhn, J. Weickert, T. Kohlberger and C. Schnörr, A multigrid platform for real-time motion computation with discontinuity-preserving variational methods, Technical Report No. 136, Department of Mathematics, Saarland University, Saarbrücken, Germany, May 2005.
- [6] V. Caselles, R. Kimmel and G. Sapiro, Geodesic active contours, Int. J. Computer Vision, 22 (1997), 61-79.
- [7] T. F. Chan and K. Chen, An optimization-based multilevel algorithm for total variation image denoising, SIAM J. Multiscale Mod. Simu., 5 (2006), 615-645.
- [8] T. F. Chan, K. Chen and J. L. Carter, Iterative methods for solving the dual formulation arising from image restoration, Elec. Trans. Numer. Anal., 26 (2007), 299-311.
- [9] T. F. Chan, K. Chen and X. C. Tai, Nonlinear multilevel schemes for solving the total variation image minimization problem, in Image Processing Based on PDEs, eds. X.-C. Tai, K.-A. Lie, T.F. Chan and S. Osher, Springer-Verlag (2006), 69-94.
- [10] T. F. Chan and J. H. Shen, Image Processing and Analysis - Variational, PDE, Wavelet, and Stochastic Methods, SIAM Publications, Philadelphia, USA, 2005.
- [11] T. F. Chan and L. A. Vese, Active contours without edges, IEEE Trans. Image Proc., 10 (2001), 266-277.
- [12] Q. S. Chang and I. Chern, Acceleration methods for total variation based image denoising, SIAM J. Sci. Comput., 25 (2003), 982-994.
- [13] K. Chen, Matrix Preconditioning Techniques and Applications, Cambridge University Press, Cambridge UK, 2005.
- [14] K. Chen and J. Savage, An accelerated algebraic multigrid algorithm for total-variation denoising, BIT, 47 (2007), 277-296.
- [15] C. Frohn-Schauf, S. Henn and K. Witsch, Nonlinear multigrid methods for total variation image denoising, Comput Visual Sci., 7 (2004), 199-206.
- [16] V. E. Henson, Multigrid methods nonlinear problems: an overview, in Computational Imaging – Proceedings of the SPIE, eds. C. A. Bouman and R. L. Stevenson, 5016 (2003), 36-48.
- [17] M. Jeon, M. Alexander, W. Pedrycz and N. Pizzi, Unsupervised hierarchical image segmentation with level set and additive operator splitting, Phys. Rev. Lett., 26 (2005), 1461-1469.
- [18] M. Kass, A. Witkin and D. Terzopoulos, Active contours models, Int. J. Computer Vision, 1 (1988), 321-331.
- [19] T. Lu, P. Neittaanmäki and X. C. Tai, A parallel splitting up method and its application to Navier-Stokes equations, Appl. Math. Lett., 4 (1991), 25-29.
- [20] L. Modica, The gradient theory of phase transitions and the minimal interface criterion, Arch. Rational Mech. Anal., 98 (1987), 123-142.
- [21] L. Modica and S. Mortola, Un esempio di Un Γ -convergenza, Boll. Un. Mat. Ital., B(5), 14 (1977), 285-299.
- [22] J.-M. Morel and S. Solimini, Variational Methods in Image Segmentation, Vol.14, Progress in Nonlinear Diff. Eqns. and Their Applics., Birkhäuser, Boston, 1995.
- [23] D. Mumford and J. Shah, Optimal approximation by piecewise smooth functions and associated variational problems, Comm. Pure Appl. Math., 42 (1989), 577-685.
- [24] M. V. Oehsen, Multiscale Methods for Variational Image Denoising, Logos Verlag, Berlin,

- 2002.
- [25] S. Osher and J. A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, *J. Comput. Phys.*, 79 (1988), 12-49.
 - [26] L. I. Rudin, S. Osher and E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D*, 60 (1992), 259-268.
 - [27] J. W. Ruge and K. Stuben, Algebraic multigrid, in *Multigrid Methods, Frontiers in Applied Mathematics (3)*, ed. S. F. McCormick, SIAM, Philadelphia (1987), 73-130.
 - [28] J. Savage and K. Chen, An improved and accelerated non-linear multigrid method for total-variation denoising, *Int. J. Computer Math.*, 82 (2005), 1001-1015.
 - [29] J. H. Shen, Γ -convergence approximation to piecewise constant Mumford-Shah segmentation, *ACIVS 2005 (Int'l Conf. Advanced Concepts Intell. Vision Systems)*, eds. J. Blanc-Talon et al, *Lec. Notes Comp. Sci.*, 3708 (2005), 499-506.
 - [30] U. Trottenberg, C. W. Oosterlee and A. Schuller, *Multigrid*, Academic Press, London, UK, 2000.
 - [31] P. S. Vassilevski and J. G. Wade, A comparison of multilevel methods for total variation regularization, *Elec. Trans. Numer. Anal.*, 6 (1997), 255-270.
 - [32] L. A. Vese and T. F. Chan, A multiphase level set framework for image segmentation using the Mumford and Shah model, *Int. J. Computer Vision*, 50 (2002), 271-293.
 - [33] C. R. Vogel, *Computational Methods for Inverse Problems*, SIAM publications, USA, 2002.
 - [34] C. R. Vogel and M. E. Oman, Iterative methods for total variation denoising, *SIAM J. Sci. Statist. Comput.*, 17 (1996), 227-238.
 - [35] W. L. Wan, T. F. Chan and B. Smith, An energy-minimizing interpolation for robust multigrid methods, *SIAM J. Sci. Comput.*, 21 (2000), 1632-1649.
 - [36] J. Weickert, B. M. ter Haar Romeny and M. A. Viergever, Efficient and reliable schemes for nonlinear diffusion filtering, *IEEE Trans. Image Proc.*, 7 (1998), 398-410.