© IEEE Transactions on Image Processing (2009), Vol. 18(5), pp.1097-1106.

On Two Multigrid Algorithms for Modelling Variational Multiphase Image Segmentation

*Noor Badshah and Ke Chen**

Abstract

In this paper we present two related multigrid algorithms for multiphase image segmentation. Algorithm I solves the model by Vese-Chan [23] (*Int. J. Computer Vision, 2002*). We first generalize our recently developed multigrid method to this multiphase segmentation model (MG1); we also give a local Fourier analysis for the local smoother which leads to a new and more effective smoother. Although MG1 is found many magnitudes faster than the fast method of additive operator splitting (AOS); both algorithms are not robust with regard to the initial guess. To overcome this dependence on the initial guess, we consider a hierarchical segmentation model [10] (*Pattern Recognition Letters, 2005*) which achieves multiphase segmentation by repeated use of the Chan-Vese two-phase model [6]; our Algorithm II solves this model by a multigrid algorithm (MG2). Numerical experiments show that both algorithms are efficient and in particular MG2 is more robust than MG1 with respect to initial guesses.

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

Keywords: Image segmentation, Level set formulation, Multigrids, AOS, Local Fourier analysis.

1 Introduction

Segmentation, referring to separating image features from backgrounds, is one of the most important tasks arising from computer vision (e.g. detecting objects) and many image processing fields (e.g. picking out special cells in cell imaging). Segmentation methods fall into several categories, including histogram analysis, region growing, edge detection and partial differential equations (PDE) based variational methods. Our main focus will be on nonlinear PDE based segmentation methods.

The PDE and variational based methods [15] are among the more recently developed tools for image segmentation. The snake model [12], the active contours [5], the gradient

*Correspondence author. Centre for Mathematical Imaging Techniques, Department of Mathematical Sciences, University of Liverpool, Peach Street, Liverpool L69 7ZL, UK. Email: k.chen@liv.ac.uk, Web: http://www.liv.ac.uk/∼cmchenke/cmit/

vector flow [25] and the curvature driven diffusion method [8] all belong to this class of PDE-based methods. In most situations, the level set method by [17] proves to be an indispensable tool for analysis and implementation. We remark that for geodesic active contours models, multigrid methods (linear) have been developed recently; see [13, 18, 19].

Our primary aim of this paper is an extension of the Badshah-Chen multigrid method [3] to the Vese-Chan multiphase image segmentation [23]. Two contributions are made in this work. Firstly direct application of [3] to [23] does not work. The observation can be explained by a local Fourier analysis (LFA) of the underlying smoother, suggesting that it is not effective. A close study shows that this ineffectiveness is due to a few image pixels only, where the linearized coefficients differ vastly. We then propose a different smoothing (under-relaxation) strategy at these 'odd' pixels. The LFA shows that the modified smoother is effective. Secondly we found that the Vese-Chan multiphase model [23] may not segment images (i.e. may not converge to the desired level set functions) if good initial guesses are not provided; this problem is inherent in the model rather than numerical solution methods. To overcome this latter problem we adopt the idea of [10] to decouple the multiphase model into repeated two phases in a hierarchical way. This prompts us to consider a multigrid method for each two phase problem. In addition to efficiency gain, as a by-product, this hierarchical approach is less dependent on initial guesses than when the method [10] alone is used. Moreover, even when good initial guesses are available, the hierarchical approach is also faster (in number of iterations and CPU) than a standard multiphase multigrid method; with our modified smoother, the two algorithms are comparable in speed.

The rest of paper is organized as follows. In Section 2 we give the brief detail of the local smoother used in a multigrid method [3] for the two-phase Chan-Vese model [6] and in Section 3 the generalized local smoother for the Vese-Chan multiphase segmentation model [23] and the corresponding nonlinear multigrid algorithm (MG1). Some LFA results are given to illustrate the shortcoming of the smoother, before we propose an improved smoother. In Section 4 the hierarchical image segmentation technique [10] is presented along with a multigrid solver (MG2). In Section 5 we present some numerical results of the two main algorithms, compared with the AOS solver, and we end the paper with some conclusions.

2 Solution of a two-phase image segmentation model

We first recall the active contour without edges method for two-phase segmentation by Chan-Vese [6], and then review the local smoother used in our recently used multigrid method [3].

Let $z = z(x, y)$ be the given image in a continuous domain $\Omega \in \mathbb{R}^2$ and in practice only a discrete image matrix $z \in \mathbb{R}^{m_1 \times m_2}$ will be given. We hope to segment z into separate and meaningful regions.

Following the Mumford-Shah [16] methodology, the Chan-Vese [6] idea is to locate two regions of approximatively piecewise constant intensities, of distinct values c_1 and c_2 ('c_1' for inside and 'c_2' for outside) by global optimization, where the object to be detected is represented by the region with intensity c_1. Let Γ denote the boundary that separates the two regions. Then the minimization problem is

$$\begin{aligned}\min_{\Gamma,c_1,c_2} F(\Gamma,c_1,c_2) &= \mu|\Gamma| + \lambda_1 \int_{inside(\Gamma)} |z(x,y)-c_1|^2 dxdy \\ &+ \lambda_2 \int_{outside(\Gamma)} |z(x,y)-c_2|^2 dxdy, \end{aligned} \quad (1)$$

where c_1 and c_2 are average values of image grey-scales inside and outside Γ respectively. To proceed with the minimization, using a level set formulation [17], let $\Gamma = \{(x,y) : \phi(x,y) = 0\}$, $inside(\Gamma) = \{(x,y) : \phi(x,y) > 0\}$, $outside(\Gamma) = \{(x,y) : \phi(x,y) < 0\}$. Then the minimization problem becomes

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

where $H(w)$ is one dimensional Heaviside function defined by 1 if $w \geq 0$ and 0 if $w < 0$. To allow conventional differentiation, introduce H_ϵ and δ_ϵ as a C^1 approximation and regularization of the Heaviside function H and Delta function δ e.g. $H_\epsilon(w) = \frac{1}{2}\Big(1 + \frac{2}{\pi}\arctan(\frac{w}{\epsilon})\Big)$ and $\delta_\epsilon(w) = H'_\epsilon(w)$. Further minimizing the energy functional, we obtain:

$$c_1(\phi) = \frac{\int_\Omega z(x,y) H_\epsilon(\phi) dxdy}{\int_\Omega H_\epsilon(\phi) dxdy}, \quad c_2(\phi) = \frac{\int_\Omega z(x,y)(1-H_\epsilon(\phi)) dxdy}{\int_\Omega (1-H_\epsilon(\phi)) dxdy},$$

$$\begin{aligned}\delta_\epsilon(\phi)\Big[\mu\ \mathrm{div}\Big(\frac{\nabla\phi}{|\nabla\phi|}\Big) - \lambda_1(z(x,y)-c_1)^2 + \lambda_2(z(x,y)-c_2)^2\Big] &= 0 \quad in \quad \Omega, \\ \frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial\vec{n}} &= 0 \quad \text{on } \partial\Omega \end{aligned} \quad (3)$$

Here once the desirable level set function $\phi = \phi(x,y)$ is found, the two-phase segmented image is given by

$$u(x,y) = c_1 H(\phi(x,y)) + c_2(1 - H(\phi(x,y)). \quad (4)$$

We give some detail of the local smoother used in the multigrid method [3] for the two-phase segmentation (3). Denote the approximation at (i,j) by $\phi_{i,j} = \phi(x_i, y_j)$. Using finite difference scheme to discretize the Euler-Lagrange's equation for ϕ, the equation at grid point (i,j) is given by

$$\begin{aligned}\delta_\epsilon(\phi_{i,j})\Bigg[\frac{\Delta^x_-}{h_1}\frac{\mu\Delta^x_+\phi_{i,j}/h_1}{\sqrt{(\Delta^x_+\phi_{i,j}/h_1)^2 + (\Delta^y_+\phi_{i,j}/h_2)^2 + \beta}} + \frac{\Delta^y_-}{h_2}\frac{\mu\Delta^y_+\phi_{i,j}/h_2}{\sqrt{(\Delta^x_+\phi_{i,j}/h_1)^2 + (\Delta^y_+\phi_{i,j}/h_2)^2 + \beta}} \\ -\lambda_1(z_{i,j}-c_1)^2 + \lambda_2(z_{i,j}-c_2)^2\Bigg] = 0,\end{aligned}$$

$$\implies \quad \Delta_-^x \frac{\bar{\mu}\Delta_+^x \phi_{i,j}}{\sqrt{(\Delta_+^x \phi_{i,j})^2 + (\lambda \Delta_+^y \phi_{i,j})^2 + \bar{\beta}}} \;+\; \Delta_-^y \frac{\bar{\mu}\lambda^2 \Delta_+^y \phi_{i,j}}{\sqrt{(\Delta_+^x \phi_{i,j})^2 + (\lambda \Delta_+^y \phi_{i,j})^2 + \bar{\beta}}}$$
$$= \lambda_1 (z_{i,j} - c_1)^2 - \lambda_2 (z_{i,j} - c_2)^2, \qquad (5)$$

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

$$\phi_{i,0} = \phi_{i,1}, \quad \phi_{i,m_2+1} = \phi_{i,m_2}, \quad \phi_{0,j} = \phi_{1,j}, \quad \phi_{m_1+1,j} = \phi_{m_1,j}, \qquad (6)$$

assuming $z \in \mathbb{R}^{m_1 \times m_2}$. The system of nonlinear equations is linearized locally, by computing $D(\phi)$ on each grid (i,j) locally using current iterations. Then we obtain a system of linear equations, which is solved by using a Gauss-Seidel method for a few steps to smooth the error. We are using a few steps of this smoother to smooth the error in a nonlinear multigrid context.

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

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

Equation (5) can be written as

$$\bar{\mu}\Big[D(\phi)_{i,j}\Delta_+^x \phi_{i,j} - D(\phi)_{i-1,j}\Delta_+^x \phi_{i-1,j}\Big] + \bar{\mu}\lambda^2\Big[D(\phi)_{i,j}\Delta_+^y \phi_{i,j} - D(\phi)_{i,j-1}\Delta_+^y \phi_{i,j-1}\Big]$$
$$= \lambda_1 (z_{i,j} - c_1)^2 - \lambda_2 (z_{i,j} - c_2)^2 \equiv f_{i,j}.$$

Let $\widetilde{\phi}$ be the current iterate of ϕ. The linear equation in $\phi_{i,j}$ is then

$$\bar{\mu}\Big[D(\widetilde{\phi})_{i,j}(\widetilde{\phi}_{i+1,j} - \phi_{i,j}) - D(\widetilde{\phi})_{i-1,j}(\phi_{i,j} - \widetilde{\phi}_{i-1,j})\Big] +$$
$$\bar{\mu}\lambda^2\Big[D(\widetilde{\phi})_{i,j}(\widetilde{\phi}_{i,j+1} - \phi_{i,j}) - D(\widetilde{\phi})_{i,j-1}(\phi_{i,j} - \widetilde{\phi}_{i,j-1})\Big] = f_{i,j}$$

i.e.

$$D(\widetilde{\phi})_{i,j}(\widetilde{\phi}_{i+1,j} - \phi_{i,j}) - D(\widetilde{\phi})_{i-1,j}(\phi_{i,j} - \widetilde{\phi}_{i-1,j}) +$$
$$\lambda^2\Big[D(\widetilde{\phi})_{i,j}(\widetilde{\phi}_{i,j+1} - \phi_{i,j}) - D(\widetilde{\phi})_{i,j-1}(\phi_{i,j} - \widetilde{\phi}_{i,j-1})\Big] = f_{i,j}/\bar{\mu} \equiv \bar{f}_{i,j},$$

or $N(\phi^h) = \bar{f}^h$ on level Ω^h.

Algorithm I (Local smoother): Let the smoothing step for $N(\phi^h) = \bar{f}^h$ be denoted by

$$\phi_\ell^h \longleftarrow Smoother(\phi^h, \bar{f}^h, maxit, tol)$$

where $h = (h_1, h_2)$ is the step size.
for $i = 1 : m_1$

for $j = 1 : m_2$

for iter=1:maxit

$\widetilde{\phi}^h \leftarrow \phi^h$, $a = D(\widetilde{\phi}^h)_{i,j}\widetilde{\phi}^h_{i+1,j} + D(\widetilde{\phi}^h)_{i-1,j}\widetilde{\phi}^h_{i-1,j}$,

$b = D(\widetilde{\phi}^h)_{i,j}\widetilde{\phi}^h_{i,j+1} + D(\widetilde{\phi}^h)_{i,j-1}\widetilde{\phi}^h_{i,j-1}$,

$$(\phi)^h_{i,j} = \frac{a + \lambda^2 b - \bar{f}_{i,j}}{D(\widetilde{\phi}^h)_{i,j} + D(\widetilde{\phi}^h)_{i-1,j} + \lambda^2(D(\widetilde{\phi}^h)_{i,j} + D(\widetilde{\phi}^h)_{i,j-1})}$$

if $\|\phi^h - \widetilde{\phi}^h\| < tol$ Stop

end

end

end

3 Multigrid algorithm I for multiphase segmentation (MG1)

The Vese-Chan multiphase segmentation model [23] is the extension of the 2-phase Chan-Vese segmentation model [6]. In [6], with one level set function, we can segment an image into two phases as one level set cannot represent more than two phases. In general, to divide an image into n phases, we need $\log_2 n$ level set functions. We remark that related work by Tai et al [14] and Ambrosio-Tortorelli [2] provides alternatives to these multiple level set functions.

Consider $p = \log_2 n$ level set functions $\phi_\ell : \Omega \to \mathbb{R}$ for $\ell = 1, 2, \ldots, p$. The union of the zero level sets of all ϕ_ℓ will represent the edges in the segmented image. For $1 \le s \le n = 2^p$, denote by $c_r = \text{mean}(z)$ the average value of image gray-scales in phase r and by χ_r the characteristic function for phase r. Then the proposed minimization energy for multiphase segmentation by Vese-Chan [23] is the following:

$$F_n(c, \Phi) = \sum_{1 \le r \le n} \int_\Omega (z(x,y) - c_r)^2 \chi_r dxdy \; + \; \mu \sum_{1 \le \ell \le p} \int_\Omega |\nabla H(\phi_\ell)| dxdy \tag{7}$$

where $c = (c_1, c_2, \ldots, c_n)$ and $\Phi = (\phi_1, \phi_2, \ldots, \phi_p)$; note $n = 2^p$. The main focus in this paper will be on the 4-phase segmentation i.e. $n = 4$ or $p = 2$, which we denote by SEG4. But all the ideas will carry over to more phases.

We shall consider the following minimization problem: $\min_{c,\Phi} F_4(c, \Phi)$ with

$$\begin{aligned} F_4(c, \Phi) \;\; = \;\; & \int_\Omega (z(x,y) - c_{11})^2 H(\phi_1) H(\phi_2) dxdy + \int_\Omega (z(x,y) - c_{10})^2 H(\phi_1)(1 - H(\phi_2)) dxdy \\ & + \int_\Omega (z(x,y) - c_{01})^2 (1 - H(\phi_1)) H(\phi_2) dxdy + \; \mu \int_\Omega |\nabla H(\phi_1)| dxdy \\ & + \int_\Omega (z(x,y) - c_{00})^2 (1 - H(\phi_1))(1 - H(\phi_2)) dxdy + \mu \int_\Omega |\nabla H(\phi_2)| dxdy \end{aligned} \tag{8}$$

where $c = (c_{11}, c_{10}, c_{01}, c_{00})$ and $\Phi = (\phi_1, \phi_2)$. Here the phase domains will be interlaced by the zero level sets of ϕ_1, ϕ_2 e.g. $\Omega_1 = \{(x,y) \; : \; \phi_1 > 0, \phi_2 > 0\}$, $\Omega_2 = \{(x,y) \; : \; \phi_1 >$

$0, \phi_2 < 0\}$ similar to the notation $inside(\Gamma)$ in (1). Once Φ is found, the segmented image u is

$$u = c_{11}H(\phi_1)H(\phi_2)+c_{10}H(\phi_1)(1-H(\phi_2))+c_{01}(1-H(\phi_1))H(\phi_2)+c_{00}(1-H(\phi_1))(1-H(\phi_2)).$$

Minimizing (8) with respect to c and Φ, we have:

$$c_{11}(\phi) = \frac{\int_\Omega zH(\phi_1)H(\phi_2)dxdy}{\int_\Omega H(\phi_1)H(\phi_2)dxdy}, \qquad c_{10}(\phi) = \frac{\int_\Omega zH(\phi_1)(1-H(\phi_2))dxdy}{\int_\Omega H(\phi_1)(1-H(\phi_2))dxdy},$$
$$c_{01}(\phi) = \frac{\int_\Omega z(1-H(\phi_1))H(\phi_2)dxdy}{\int_\Omega (1-H(\phi_1))H(\phi_2)dxdy}, \qquad c_{00}(\phi) = \frac{\int_\Omega z(1-H(\phi_1))(1-H(\phi_2))dxdy}{\int_\Omega (1-H(\phi_1))(1-H(\phi_2))dxdy}$$

and the following Euler-Lagrange's equations

$$\begin{cases} \delta_\epsilon(\phi_1)\Big[\mu\nabla\cdot\dfrac{\nabla\phi_1}{|\nabla\phi_1|} - [T_1H_\epsilon(\phi_2) + T_2(1-H_\epsilon(\phi_2))]\Big] = 0, \\ \delta_\epsilon(\phi_2)\Big[\mu\nabla\cdot\dfrac{\nabla\phi_2}{|\nabla\phi_2|} - [T_3H_\epsilon(\phi_1) + T_4(1-H_\epsilon(\phi_1))]\Big] = 0, \end{cases} \tag{9}$$

with Neumann's boundary conditions, where $T_1 = (z-c_{11})^2 - (z-c_{01})^2$,$T_2 = (z-c_{10})^2 - (z-c_{00})^2$, $T_3 = (z-c_{11})^2 - (z-c_{10})^2$ and $T_4 = (z-c_{01})^2 - (z-c_{00})^2$. We shall shortly discuss how to solve (9) efficiently.

An easy but less efficient alternative is to solve the following evolution problem

$$\begin{cases} \dfrac{\partial\phi_1}{\partial t} = \delta_\epsilon(\phi_1)\Big[\mu\nabla\cdot\dfrac{\nabla\phi_1}{|\nabla\phi_1|} - [T_1H_\epsilon(\phi_2) + T_2(1-H_\epsilon(\phi_2))]\Big], \\ \dfrac{\partial\phi_2}{\partial t} = \delta_\epsilon(\phi_2)\Big[\mu\nabla\cdot\dfrac{\nabla\phi_2}{|\nabla\phi_2|} - [T_3H_\epsilon(\phi_1) + T_4(1-H_\epsilon(\phi_1))]\Big], \end{cases} \tag{10}$$

with initial conditions $\phi_1(0,x,y) = \phi_{1,0}(x,y), \phi_2(0,x,y) = \phi_{2,0}(x,y)$. In [23] these parabolic equations were solved using the additive operator-splitting (AOS, semi-implicit) method which will be used later for comparison.

3.1 Multigrid algorithm I

Instead of (10), we now consider solving (9) using a multigrid method. The general idea of multigrid method for an operator equation $N(\phi) = f$ is the following. Suppose that a smooth approximation $\widetilde{\phi}^h$ of ϕ has been obtained on some fine grid Ω^h; by 'smooth' we mean that both the solution $\widetilde{\phi}^h$ and the residual $r^h = f^h - N^h(\widetilde{\phi})$ are smooth (i.e. they contain mainly the low frequency components when projected into the Fourier space). If this assumption is true, then an improvement over $\widetilde{\phi}^h$ can be made by restricting the operator equation (residual equation) to a coarse grid (with much less unknowns) Ω^H and solving the (cheaper) coarse grid equation

$$N^H(w^H) = N^H(\phi^H) + Rr^h,$$

and once w^H is obtained, the correction to $\widetilde{\phi}^h$ can be obtained on Ω^h which leads to a better approximation of ϕ^h as follows

$$\bar{\phi}^h = \widetilde{\phi}^h + P(w^H - \phi^H)$$

where R is a restriction operator transferring information from Ω^h to Ω^H, P is an interpolation operator transferring information from Ω^H to Ω^h and $\phi^H = R\widetilde{\phi}^h$ is the initial solution on Ω^H. Refer to [4, 22, 9, 7, 3, 13, 18, 19] and the references therein. Crucially among the three key stages of a multigrid method, the step of getting a smooth solution $\widetilde{\phi}^h$ is called smoothing and the iterative method used for this purpose is named as a smoother whose effectiveness determines the success or the failure of a multigrid method.

Let $(\phi_\ell)_{i,j} = \phi_\ell(x_i, y_j)$, for $\ell = 1, 2$. Using finite difference schemes to discretize (9) for ϕ_ℓ, the equations at a pixel point (i, j) are given by

$$\begin{cases} \delta(\phi_1)_{i,j}\Big\{ \dfrac{\triangle_-^x}{h_1} \dfrac{\mu\triangle_+^x(\phi_1)_{i,j}/h_1}{\sqrt{(\triangle_+^x(\phi_1)_{i,j}/h_1)^2 + (\triangle_+^y(\phi_1)_{i,j}/h_2)^2 + \beta}} - (T_1)_{i,j}H_\epsilon(\phi_2)_{i,j} + \\ \quad \dfrac{\triangle_-^y}{h_2} \dfrac{\mu\triangle_+^y(\phi_1)_{i,j}/h_2}{\sqrt{(\triangle_+^x(\phi_1)_{i,j}/h_1)^2 + (\triangle_+^y(\phi_1)_{i,j}/h_2)^2 + \beta}} - (T_2)_{i,j}(1 - H_\epsilon(\phi_2)_{i,j})\Big\} = 0, \\ \delta(\phi_2)_{i,j}\Big\{ \dfrac{\triangle_-^x}{h_1} \dfrac{\mu\triangle_+^x(\phi_2)_{i,j}/h_1}{\sqrt{(\triangle_+^x(\phi_2)_{i,j}/h_1)^2 + (\triangle_+^y(\phi_2)_{i,j}/h_2)^2 + \beta}} - (T_3)_{i,j}H_\epsilon(\phi_1)_{i,j} + \\ \quad \dfrac{\triangle_-^y}{h_2} \dfrac{\mu\triangle_+^y(\phi_2)_{i,j}/h_2}{\sqrt{(\triangle_+^x(\phi_2)_{i,j}/h_1)^2 + (\triangle_+^y(\phi_2)_{i,j}/h_2)^2 + \beta}} - (T_4)_{i,j}(1 - H_\epsilon(\phi_1)_{i,j})\Big\} = 0, \end{cases} \tag{11}$$

where $(T_1)_{i,j} = (z_{i,j} - c_{11})^2 - (z_{i,j} - c_{01})^2$, $(T_2)_{i,j} = (z_{i,j} - c_{10})^2 - (z_{i,j} - c_{00})^2$, $(T_3)_{i,j} = (z_{i,j} - c_{11})^2 - (z_{i,j} - c_{10})^2$ and $(T_4)_{i,j} = (z_{i,j} - c_{11})^2 - (z_{i,j} - c_{00})^2$.

Let $\bar{\mu} = \mu/h_1$, $\bar{\beta} = h_1^2\beta$ and $\lambda = h_1/h_2$. Also denote $(f_1)_{i,j} = (T_1)_{i,j}H_\epsilon(\phi_2)_{i,j} + (T_2)_{i,j}(1 - H_\epsilon(\phi_2)_{i,j})$ and $(f_2)_{i,j} = (T_3)_{i,j}H_\epsilon(\phi_1)_{i,j} + (T_4)_{i,j}(1 - H_\epsilon(\phi_1)_{i,j})$. Further for $\ell = 1, 2$, denote the coefficients (to be freezed) by

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

We can simplify equation (11) to

$$\begin{aligned} &\Big[D_\ell(\phi_\ell)_{i,j}((\phi_\ell)_{i+1,j} - (\phi_\ell)_{i,j}) - D_\ell(\phi_\ell)_{i-1,j}((\phi_\ell)_{i,j} - (\phi_\ell)_{i-1,j})\Big] \\ &+\lambda^2\Big[D_\ell(\phi_\ell)_{i,j}((\phi_\ell)_{i,j+1} - (\phi_\ell)_{i,j}) - D_\ell(\phi_\ell)_{i,j-1}((\phi_\ell)_{i,j} - (\phi_\ell)_{i,j-1})\Big] = (\bar{f}_\ell)_{i,j}, \end{aligned} \tag{12}$$

where $\bar{f}_\ell = f_\ell/\bar{\mu}$, with the boundary conditions

$$(\phi_\ell)_{i,0} = (\phi_\ell)_{i,1}, \quad (\phi_\ell)_{i,m_2+1} = (\phi_\ell)_{i,m_2}, \ (\phi_\ell)_{0,j} = (\phi_\ell)_{1,j}, \quad (\phi_\ell)_{m_1+1,j} = (\phi_\ell)_{m_1,j}. \tag{13}$$

Let $\widetilde{\phi}_\ell$ be the approximation to ϕ_ℓ at the current iteration. Then from equation (12), pursuing only local unknowns ϕ_ℓ at (i,j), we have the following linear equations

$$\Big[D_\ell(\widetilde{\phi}_\ell)_{i,j}((\widetilde{\phi}_\ell)_{i+1,j} - (\phi_\ell)_{i,j}) - D_\ell(\widetilde{\phi}_\ell)_{i-1,j}((\phi_\ell)_{i,j} - (\widetilde{\phi}_\ell)_{i-1,j})\Big]$$
$$+\lambda^2\Big[D_\ell(\widetilde{\phi}_\ell)_{i,j}((\widetilde{\phi}_\ell)_{i,j+1} - (\phi_\ell)_{i,j}) - D_\ell(\widetilde{\phi}_\ell)_{i,j-1}((\phi_\ell)_{i,j} - (\widetilde{\phi}_\ell)_{i,j-1})\Big] = (\bar{f}_\ell)_{i,j}. \tag{14}$$

Our proposed algorithm solves these equations for $(\phi_\ell)_{i,j}$ to update $(\widetilde{\phi}_\ell)_{i,j}$ which leads to updating the coefficients locally and further iterations (before moving to the next pixel in a Gauss-Seidel fashion). Denote the system of non-linear equations from (12) by

$$\begin{cases} N_1^h(\phi_1^h) = \bar{f}_1^h, \\ N_2^h(\phi_2^h) = \bar{f}_2^h, \end{cases} \tag{15}$$

where ϕ_ℓ^h and f_ℓ^h are grid functions on a $m_1 \times m_2$ cell centered rectangular grid Ω^h with spacing $h = (h_1, h_2)$ and $l = 1, 2$.

We shall first summarize this local smoother and then present our algorithm I.

Algorithm II (Smoother for multiphase model): Let a smoothing step for (12) via (14) be

$$\phi_\ell^h \longleftarrow Smoother(\phi_\ell^h, \bar{f_\ell}^h, maxit, tol)$$

where $\ell = 1, 2$ and h is the step size on level Ω^h.

for $i = 1 : m_1$

for $j = 1 : m_2$

for iter=1:maxit

$\widetilde{\phi_\ell}^h \leftarrow \phi_\ell^h, \quad A_\ell = D_\ell(\widetilde{\phi}_\ell)_{i,j}^h(\widetilde{\phi}_\ell)_{i+1,j}^h + D_\ell(\widetilde{\phi}_\ell)_{i-1,j}^h(\widetilde{\phi}_\ell)_{i-1,j}^h,$
$B_\ell = D_\ell(\widetilde{\phi}_\ell)_{i,j}^h(\widetilde{\phi}_\ell)_{i,j+1}^h + D_\ell(\widetilde{\phi}_\ell)_{i,j-1}^h(\widetilde{\phi}_\ell)_{i,j-1}^h,$

$$(\phi_\ell)_{i,j}^h = \frac{A_\ell + \lambda^2 B_\ell - \bar{f}_{\ell i,j}}{D_\ell(\widetilde{\phi}_\ell)_{i,j}^h + D_\ell(\widetilde{\phi}_\ell)_{i-1,j}^h + \lambda^2(D_\ell(\widetilde{\phi}_\ell)_{i,j}^h + D_\ell(\widetilde{\phi}_\ell)_{i,j-1}^h)}$$

if $\|\phi_\ell^h - (\widetilde{\phi_\ell}^h\| < tol$ Stop

end

end

end

Multigrid Algorithm: Equation (15) will be solved by the following multigrid algorithm [3, 4, 9, 22, 7] – the full approximation scheme of Brandt.

Algorithm III (Multigrid Algorithm): Assume we have set up these multigrid parameters:

ν_1 pre-smoothing steps on each level

ν_2 post-smoothing steps on each level

γ the number of multigrid cycles on each level ($\gamma = 1$ for V-cycling and $\gamma = 2$ for W-cycling). Set $tol = 0.1$. Here we present one step V-cycle of nonlinear multigrid method for SEG4. A multigrid cycle refers to one call to the procedure

$$\phi_\ell^h \longleftarrow FASCYC^h(\phi_\ell^h, \bar{f_\ell}^h, \nu_1, \nu_2)$$

1. If Ω^h is the coarsest grid, then solve equation (15) using a time marching technique (the AOS method) and then stop.
   Else do the Pre-Smoothing step:

$$\phi_\ell^h \longleftarrow Smoother(\phi_\ell^h, \bar{f}_\ell^{\,h}, \nu_1, tol)$$

2. **Restriction:**

$$\phi_\ell^{2h} = I_h^{2h}\phi_\ell^h, \quad \bar{\phi}_\ell^{\,2h} = \phi_\ell^{2h},$$
$$\bar{f}_\ell^{\,2h} = I_h^{2h}(\bar{f}_\ell^{\,h} - N_\ell^h\phi_\ell^h) + N_\ell^{2h}\phi_\ell^{2h},$$
$$\phi_\ell^{2h} \longleftarrow FASCYC^{2h}(\phi_\ell^{2h}, \bar{f}_\ell^{\,2h}, \nu_1, \nu_2)$$

3. **Interpolation**

$$\phi_\ell^h \longleftarrow \phi_\ell^h + I_{2h}^h(\phi_\ell^{2h} - \bar{\phi}_\ell^{\,2h})$$

4. Implement the Post-Smoothing step:

$$\phi_\ell^h \longleftarrow Smoother(\phi_\ell^h, \bar{f}_\ell^{\,h}, \nu_2, tol).$$

Here the restriction operator I_h^{2h} is by full weighting and the interpolation I_{2h}^h by the bilinear operator [22].

As we know, an effective multigrid algorithm relies on two necessary ingredients: the smooth residual errors on fine levels and effective error corrections from coarse levels. In practice, effectiveness of a smoother to smooth errors is the key to the success. In the next section we give the local Fourier analysis of our smoother in Algorithm II.

3.2 Local Fourier analysis and a modified smoother

The local Fourier analysis (LFA) is a suitable tool to analyze the convergence rate of any iterative method for linear equations. However our underlying equations are nonlinear so a LFA will consider a linearized equation, and as linearization occurs locally at each pixel, we shall look for the maximum rate from all pixel locations.

Consider Algorithm II in the case of a square image with $m = m_1 = m_2$ and also $h_1 = h_2 = h$ for simplicity; then $\lambda = 1$. Given the previous iterate at step k, $\widetilde{\phi}_\ell = \phi_\ell^{(k)}$, denote $a_1 = D_1(\widetilde{\phi}_1)_{i-1,j}$, $a_2 = D_1(\widetilde{\phi}_1)_{i,j}$, $a_3 = D_1(\widetilde{\phi}_1)_{i,j-1}$, $b_1 = D_2(\widetilde{\phi}_2)_{i-1,j}$, $b_2 = D_2(\widetilde{\phi}_2)_{i,j}$, $b_3 = D_2(\widetilde{\phi}_2)_{i,j-1}$ which are to be considered as local constants. From (12), the grid equation at (i,j) is the following

$$\begin{cases} -(a_1+2a_2+a_3)(\phi_1)_{i,j} + a_1(\phi_1)_{i-1,j} + a_3(\phi_1)_{i,j-1} + a_2[(\phi_1)_{i+1,j} + (\phi_1)_{i,j+1}] \\ \qquad\qquad = (\bar{f}_1)_{i,j}, \\ -(b_1+2b_2+b_3)(\phi_2)_{i,j} + b_1(\phi_2)_{i-1,j} + b_3(\phi_2)_{i,j-1} + b_2[(\phi_2)_{i+1,j} + (\phi_2)_{i,j+1}] \\ \qquad\qquad = (\bar{f}_2)_{i,j}. \end{cases} \tag{16}$$

And our local smoother can be written as

$$\begin{cases} -(a_1+2a_2+a_3)(\phi_1)_{i,j}^{(k+1)} + a_1(\phi_1)_{i-1,j}^{(k+1)} + a_3(\phi_1)_{i,j-1}^{(k+1)} + a_2[(\phi_1)_{i+1,j}^{(k)} + (\phi_1)_{i,j+1}^{(k)}] \\ \qquad\qquad = (\bar{f}_1)_{i,j}, \\ -(b_1+2b_2+b_3)(\phi_2)_{i,j}^{(k+1)} + b_1(\phi_2)_{i-1,j}^{(k+1)} + b_3(\phi_2)_{i,j-1}^{(k+1)} + b_2[(\phi_2)_{i+1,j}^{(k)} + (\phi_2)_{i,j+1}^{(k)}] \\ \qquad\qquad = (\bar{f}_2)_{i,j}. \end{cases} \tag{17}$$

Define the error functions by $e_1^{(k)} = \phi_1 - \phi_1^{(k)}$ and $e_2^{(k)} = \phi_2 - \phi_2^{(k)}$. Then using (16) and (17), the error equations are

$$\begin{cases} a_1(e_1)_{i-1,j}^{(k+1)} + a_3(e_1)_{i,j-1}^{(k+1)} + a_2[(e_1)_{i+1,j}^{(k)} + (e_1)_{i,j+1}^{(k)}] - (a_1+2a_2+a_3)(e_1)_{i,j}^{(k+1)} = 0 \\ b_1(e_2)_{i-1,j}^{(k+1)} + b_3(e_2)_{i,j-1}^{(k+1)} + b_2[(e_2)_{i+1,j}^{(k)} + (e_2)_{i,j+1}^{(k)}] - (b_1+2b_2+b_3)(e_2)_{i,j}^{(k+1)} = 0. \end{cases} \tag{18}$$

Recall that the LFA measures the largest amplification factor in a relaxation scheme [4, 9, 22]. With $\mathbf{i} = \sqrt{(-1)}$, let a general Fourier component be

$$\Theta_{\alpha,\beta}(x_i, y_j) = \exp\left(\mathbf{i}\theta_\alpha \frac{x_i}{h} + \mathbf{i}\theta_\beta \frac{y_j}{h}\right) = \exp\left(\frac{2\mathbf{i}\alpha i\pi}{m} + \frac{2\mathbf{i}\beta j\pi}{m}\right).$$

Note that $\theta_\alpha, \theta_\beta \in [-\pi, \pi]$. The LFA expands

$$e_1^{(k)} = \sum_{\alpha,\beta=-m/2}^{m/2} (\psi_1^{(k)})_{\alpha,\beta}\Theta_{\alpha,\beta}(x_i, y_j), \quad e_2^{(k)} = \sum_{\alpha,\beta=-m/2}^{m/2} (\psi_2^{(k)})_{\alpha,\beta}\Theta_{\alpha,\beta}(x_i, y_j)$$

in Fourier components. We look for the largest spectral radius (maximum eigenvalue) of the amplification matrix $\mathcal{A}_{\alpha,\beta}$ [7, 22]:

$$\begin{bmatrix} (\psi_1^{(k+1)})_{\alpha,\beta} \\ (\psi_2^{(k+1)})_{\alpha,\beta} \end{bmatrix} = \mathcal{A}_{\alpha,\beta} \begin{bmatrix} (\psi_1^{(k)})_{\alpha,\beta} \\ (\psi_2^{(k)})_{\alpha,\beta} \end{bmatrix}.$$

After plugging in these components into (18) for $e_1^{(k+1)}, e_1^{(k)}$ and $e_2^{(k+1)}, e_2^{(k)}$, we have:

$$\mathcal{A}_{\alpha,\beta} = \begin{bmatrix} \dfrac{a_2\left(e^{\frac{2\alpha\pi}{m}} + e^{\frac{2\mathbf{i}\beta\pi}{m}}\right)}{\left(a_1+2a_2+a_3-a_1e^{\frac{-2\mathbf{i}\alpha\pi}{m}} - a_3e^{\frac{-2\mathbf{i}\beta\pi}{m}}\right)} & 0 \\ 0 & \dfrac{b_2\left(e^{\frac{2\mathbf{i}\alpha\pi}{m}} + e^{\frac{2\mathbf{i}\beta\pi}{m}}\right)}{\left(b_1+2b_2+b_3-b_1e^{\frac{-2\mathbf{i}\alpha\pi}{m}} - b_3e^{\frac{-2\mathbf{i}\beta\pi}{m}}\right)} \end{bmatrix}.$$

At the kth iteration, each rate $\bar{\mu}^{(k)}(i,j) = \max_{\alpha,\beta} \rho(\mathcal{A}_{\alpha,\beta})$ in the high frequency range $(\theta_\alpha, \theta_\beta) \in [-\pi, \pi] \setminus [-\frac{\pi}{2}, \frac{\pi}{2}]$, measuring the effectiveness of a smoother [4], is dependent on $a_\ell, b_\ell, \ell = 1, 2, 3$, which in turn depend on the pixel location (i, j). Therefore we should look for the largest smoothing rate for all i, j (i.e. among all such pixels): $\hat{\mu} = \max_{a_1,a_2,a_3,b_1,b_2,b_3} \bar{\mu}^{(k)}(i,j)$.

However, due to the high nonlinearity, we found it useful to define the smoothing rate as the maximum of the above accumulated rates out of all s relaxation steps by

$$\hat{\mu}_s = \max_{i,j} \bar{\mu}^{(1)}(i,j)\bar{\mu}^{(2)}(i,j)\cdots\bar{\mu}^{(s)}(i,j).$$

Clearly for linear equations where a_ℓ, b_ℓ are constants, $\bar{\mu} = \bar{\mu}^{(k)}$ is a constant so $\hat{\mu}_s = \bar{\mu}^s$. Here as a_ℓ, b_ℓ are not constants, with this particular definition, we would allow the possibility of $\bar{\mu}^{(k)}(i,j) \approx 1$ for some i, j, k; as long as $\hat{\mu}_s \ll 1$, we would say a smoother is effective.

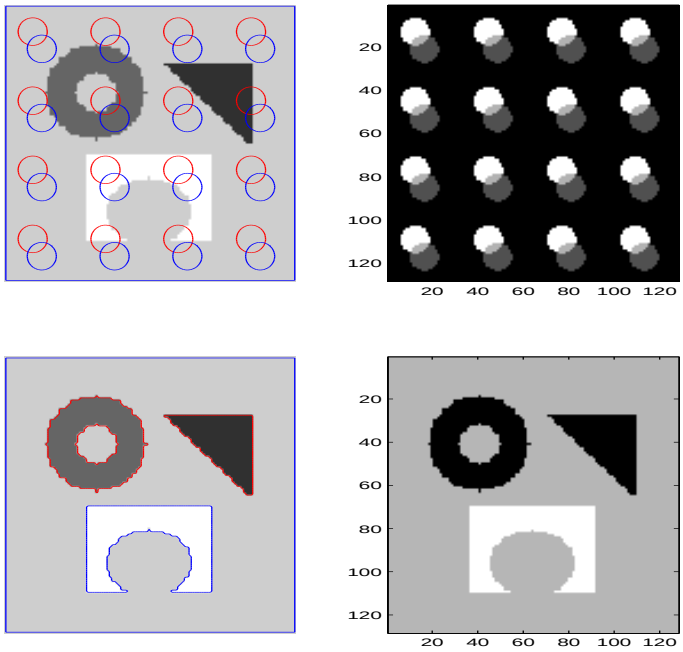

Fig. 1: Segmentation of the top left image into the bottom right image.

In Table 1, we take the particular example of Figure 1 of segmenting an artificial image. We use the image size $m = 32$; note the similar results are obtained with taking $m = 64$. Here in Table 1, the "odd pixels" refer to positions where the relative ratios

Tab. 1: The smoothing rate for a local smoother with 3 inner iterations

Outer iterations s	The smoothing rate $\hat{\mu}_s$	The smoothing rate taking out "odd pixels" $\hat{\mu}_s^*$
1	0.6862	0.5720
2	0.6861	0.3170
3	0.6861	0.2747

between a_2 and $\max(a_1, a_3)$ or the ratios between b_2 and $\max(b_1, b_3)$ are quite large; clearly our smoother is ineffective overall due to these odd pixels. This prompted us to consider how to improve the overall smoothing rate (Column 2 in Table 1).

A modified smoother. To motivate the idea, consider the particular case of an odd pixel assigned with

$$a_1 = 0.3536, a_2 = 10000, a_3 = 0.3536, b_1 = 0.3536, b_2 = 10000, b_3 = 0.3536 \tag{19}$$

and the above LFA gives a local (large) rate of $\mu = 0.99996$. Now we propose, as an alternative to (17), the following under-relaxation smoothing scheme at these odd pixels

$$\begin{cases} a_1(\phi_1)_{i-1,j}^{(k+1)} + a_3(\phi_1)_{i,j-1}^{(k+1)} + a_2[(\phi_1)_{i+1,j}^{(k)} + (\phi_1)_{i,j+1}^{(k)}] \\ \qquad - (a_1+2a_2+a_3)(1+\omega)(\phi_1)_{i,j}^{(k+1)} + \omega(a_1+2a_2+a_3)(\phi_1)_{i,j}^{(k)} = (\bar{f}_1)_{i,j}, \\ b_1(\phi_2)_{i-1,j}^{(k+1)} + b_3(\phi_2)_{i,j-1}^{(k+1)} + b_2[(\phi_2)_{i+1,j}^{(k)} + (\phi_2)_{i,j+1}^{(k)}] \\ \qquad - (b_1+2b_2+b_3)(1+\omega)(\phi_2)_{i,j}^{(k+1)} + \omega(b_1+2b_2+b_3)(\phi_2)_{i,j}^{(k)} = (\bar{f}_2)_{i,j}, \end{cases} \tag{20}$$

for some $0 \le \omega \le 1$ (note $\omega = 0$ reduces to the previous local smoother). The new error equation is

$$\begin{cases} a_1(e_1)_{i-1,j}^{(k+1)} + a_3(e_1)_{i,j-1}^{(k+1)} + a_2[(e_1)_{i+1,j}^{(k)} + (e_1)_{i,j+1}^{(k)}] \\ \qquad - (a_1+2a_2+a_3)(1+\omega)(e_1)_{i,j}^{(k+1)} + \omega(a_1+2a_2+a_3)(e_1)_{i,j}^{(k)} = 0, \\ b_1(e_2)_{i-1,j}^{(k+1)} + b_3(e_2)_{i,j-1}^{(k+1)} + b_2[(e_2)_{i+1,j}^{(k)} + (e_2)_{i,j+1}^{(k)}] \\ \qquad - (1+\omega)(b_1+2b_2+b_3)(e_2)_{i,j}^{(k+1)} + \omega(b_1+2b_2+b_3)(e_2)_{i,j}^{(k)} = 0. \end{cases} \tag{21}$$

Then the corresponding new Fourier amplification matrix is

$$\mathcal{A}_{\alpha,\beta} = \begin{bmatrix} \dfrac{a_2\left(e^{\frac{2\alpha\pi}{m}} + e^{\frac{2\mathbf{i}\beta\pi}{m}}\right) + \omega(a_1+2a_2+a_3)}{\left((1+\omega)(a_1+2a_2+a_3) - a_1 e^{\frac{-2\mathbf{i}\alpha\pi}{m}} - a_3 e^{\frac{-2\mathbf{i}\beta\pi}{m}}\right)} & 0 \\ 0 & \dfrac{b_2\left(e^{\frac{2\mathbf{i}\alpha\pi}{m}} + e^{\frac{2\mathbf{i}\beta\pi}{m}}\right) + \omega(b_1+2b_2+b_3)}{\left((1+\omega)(b_1+2b_2+b_3) - b_1 e^{\frac{-2\mathbf{i}\alpha\pi}{m}} - b_3 e^{\frac{-2\mathbf{i}\beta\pi}{m}}\right)} \end{bmatrix}.$$

Now for (19) with $\omega = 0.7$, the above new scheme yields a much pleasing rate of $\mu = 0.75026$.

Therefore, our new smoother will be (20) using a variable ω written in a form similar to (14) as

$$\begin{aligned} & D_\ell(\widetilde{\phi}_\ell)_{i,j}\Big[(\widetilde{\phi}_\ell)_{i+1,j} - (1+\omega)(\phi_\ell)_{i,j} + \omega(\widetilde{\phi}_\ell)_{i,j}\Big] \\ -\ & D_\ell(\widetilde{\phi}_\ell)_{i-1,j}\Big[(1+\omega)(\phi_\ell)_{i,j} - \omega(\widetilde{\phi}_\ell)_{i,j} - (\widetilde{\phi}_\ell)_{i-1,j}\Big] \\ +\ & \lambda^2 D_\ell(\widetilde{\phi}_\ell)_{i,j}\Big[(\widetilde{\phi}_\ell)_{i,j+1} - (1+\omega)(\phi_\ell)_{i,j} + \omega(\widetilde{\phi}_\ell)_{i,j}\Big] \\ & - \lambda^2 D_\ell(\widetilde{\phi}_\ell)_{i,j-1}\Big[(1+\omega)(\phi_\ell)_{i,j} - \omega(\widetilde{\phi}_\ell)_{i,j} - (\widetilde{\phi}_\ell)_{i,j-1})\Big] = (\bar{f}_\ell)_{i,j}. \end{aligned} \tag{22}$$

It may be stated as follows.

Algorithm IV (Modified smoother for multiphase model): Denote a smoothing step for (15), using (22), by

$$\phi_\ell^h \longleftarrow Smoother(\phi_\ell^h, \bar{f}_\ell{}^h, \text{maxit}, \omega, K, tol)$$

where $\ell = 1, 2$ and h is the step size on level Ω^h. Set $K = 100$.
for $i = 1 : m_1$
    for $j = 1 : m_2$

for $iter = 1 : \text{maxit}$

if $|D_\ell(\widetilde{\phi}_\ell)^h_{i,j}| \geq K \max(|D_\ell(\widetilde{\phi}_\ell)^h_{i-1,j}|, |D_\ell(\widetilde{\phi}_\ell)^h_{i,j-1}|)$ for any ℓ, set $\omega = 0.7$; otherwise set $\omega = 0$.

$$\widetilde{\phi}_\ell^{\,h} \leftarrow \phi^h_\ell,$$

$$A_\ell = D_\ell(\widetilde{\phi}_\ell)^h_{i,j}((\widetilde{\phi}_\ell)^h_{i+1,j}+\omega(\widetilde{\phi}_\ell)^h_{i,j})+D_\ell(\widetilde{\phi}_\ell)^h_{i-1,j}((\widetilde{\phi}_\ell)^h_{i-1,j}+\omega(\widetilde{\phi}_\ell)^h_{i,j}),$$

$$B_\ell = D_\ell(\widetilde{\phi}_\ell)^h_{i,j}((\widetilde{\phi}_\ell)^h_{i,j+1}+\omega(\widetilde{\phi}_\ell)^h_{i,j})+D_\ell(\widetilde{\phi}_\ell)^h_{i,j-1}((\widetilde{\phi}_\ell)^h_{i,j-1}+\omega(\widetilde{\phi}_\ell)^h_{i,j}),$$

$$(\phi_\ell)^h_{i,j} = \frac{A_\ell + \lambda^2 B_\ell - \bar{f}_{\ell i,j}}{(1+\omega)(D_\ell(\widetilde{\phi}_\ell)^h_{i,j} + D_\ell(\widetilde{\phi}_\ell)^h_{i-1,j} + \lambda^2(D_\ell(\widetilde{\phi}_\ell)^h_{i,j} + D_\ell(\widetilde{\phi}_\ell)^h_{i,j-1}))}$$

if $\|\phi^h_\ell - \widetilde{\phi}_\ell^{\,h}\| < tol$ Stop

end

end

end

We now repeat the smoothing analysis as was done in Table 1 and show the new results in Table 2. Clearly the new rates are much more acceptable (note the accumulated number of smoothing steps is $3s$ since we use 3 inner iterations for each outer step). In Section 5,

Tab. 2: The smoothing rate for a modified local smoother

Outer iterations s	The smoothing rate $\hat{\mu}_s$
1	0.5720
2	0.3170
3	0.2747

we shall compare the performance of the two smoothers in MG1.

4 Multigrid algorithm II for multiphase segmentation (MG2)

As previously remarked, a time-marching solution scheme is employed in the original work of Vese-Chan [23]. Realizing that this scheme is extremely slow to converge, Jeon et al [10] proposed a hierarchical image segmentation method which essentially abandoned this multiphase model in favour of the earlier Chan-Vese [6] model. The idea of Jeon et al [10] is the following. We first use the two-phase model [6] to segment the given image z into two phases (a domain and its complement) using a single level set function ϕ. We then segment one of the phases using the two-phase model [6] again and this process is repeated until the desirable number of phases is achieved. Here there are two key decisions made: (i) The domain having the larger intensity variation will be the next segmentation target. (ii) The domain having the smaller intensity variation will be replaced by a uniform intensity equal to the average intensity of the larger intensity domain. This gives rise to a new image z_{new} – a modified image of z – the new image z_{new} will be segmented. Here the purpose of (ii) is to avoid re-segmenting the domain with the smaller intensity variation.

The aim of this section is to combine the multigrid algorithm [3] for two-phase segmentation with this unsupervised hierarchical image segmentation algorithm [10] and then to assess if any advantage can be gained over Algorithm III — our motivation stems from an observation of [3] that the multigrid algorithm can help reach a global minimizer of a two-phase model (i.e. less dependent on initial guesses) while it is not true with the multiphase model (from using Algorithm III).

We first review the important definition used in steps (i-ii) above and then present the combined hierarchical multigrid algorithm. We shall denote by

$$[\phi, c_1, c_2] \leftarrow MGM(\phi, z)$$

the process of utilizing the multigrid algorithm [3] to segment a given image z by working out the desirable level set function ϕ, the two associated constants c_1, c_2; refer to (4).

Definition 4.1 (Intensity Variation [10]): Let z be the given image and S_1, S_2 denote a partition of z, obtained by segmentation using one level set function. Then the intensity variation across S_i is given by

$$\mathrm{Var}(S_i) = \frac{1}{M_i} \sum_{\ell=1}^{M_i} (z(x_\ell, y_\ell) - C_\ell)^2,$$

where C_ℓ represents the average intensity of S_i and M_i is the number of pixels in S_i.

Algorithm V (Hierarchical segmentation by multigrid method):
Let n be the required number of segmentation phases and z the given image.
Assume ϕ_0 is an initial contour (which can be a simple pattern or may be worked out by a full multigrid idea as in [3]).
for $i = 1, \ldots, n-1$
$\phi_i \leftarrow \phi_0$
$[\phi_i, c_{i1}, c_{i2}] \leftarrow MGM(\phi_i, z)$, using the 2-phase multigrid method.
Define $S_1 = \{(k,\ell) \mid (\phi_i)_{k,\ell} < 0\}$ and $S_2 = \{(k,\ell) \mid (\phi_i)_{k,\ell} \geq 0\}$.
Compute $\mathrm{Var}(S_1), \mathrm{Var}(S_2)$.
Find $j = \mathrm{argmin}_\ell \mathrm{Var}(S_\ell)$ and denote $q = \{1,\ 2\} \backslash \{j\}$.
Save the index set $W_i = S_j$. (Note $j, q = 1$ or 2.)
If $i > 1$, find the true index set by modifying $W_i = (W_z \backslash W_{i-1}) \cup W_i$ else continue.
Set $z(S_j) = c_{iq}$ since S_q is the domain with the larger variation.
end
Set the final (phase) index set $W_n = (W_z \backslash W_{n-2}) \cup S_q$.

Here assume that W_z denotes the index set of all pixels so the quantity $(W_z \backslash W_{i-1})$ singles out the index set being segmented. Also we can only use the sets S_1, S_2 to identify the domain with the larger variation (to proceed) but its complement (phase i) must be found through $(W_z \backslash W_{i-1})$.

We remark that the main algorithm presented in [10, p.1465] has a major typo, where 'smallest' should mean 'largest', and also the loop should end at $n-1$ rather than n.

Finally once the algorithm is completed, the segmented image will be separated by the index sets $W_1, W_2, \ldots, W_n$ from which we compute the mean gray values C_j's, as in (3), by

$$C_j = \sum_{(i,k)\in W_j} z_{i,k}/M_j$$

where M_j is the cardinality of W_j. Further the (piecewise) segmented image can be written as $u = (u_{i,k})$ with

$$u_{i,k} = C_j \quad \text{if} \quad (i,k) \in W_j \quad \text{for all} \quad (i,k) \in W_z = W_1 \cup \cdots \cup W_n, \tag{23}$$

which is similar to the two-phase case (4) with $n = 2$. Clearly for n phases, we now require altogether $n-1$ level set functions while the previous multiphase method [23] only requires $\log_2 n$ level set functions. For small n, the difference is small; however for large n MG2 will have to store much more level set functions (matrices) than MG1.

5 Numerical Results

In this section we present experimental results to illustrate the two multigrid algorithms (MG1 and MG2) versus the time marching method i.e.

MG1	–	Algorithm III with the local smoother (Algorithm II);
MG1m	–	Algorithm III with the modified local smoother (Algorithm IV);
MG2	–	Algorithm V with the hierarchical segmentation
AOS	–	The additive operator splitting method (time-marching [10])
SA	–	The smoother alone, Fixed point Gauss Seidel iterations.

We shall first compare the qualitative results of segmentation and then compare these solvers in speed of segmentation (iteration steps and CPU time). Although we have done many test examples, we show one artificial image and a real life image here, as shown in Figure 2.

Segmentation results. The main parameter μ in the segmentation model balances the regularization term and the fitting term (fitting a phase domain with its average gray-scale levels). Here for Problems 1 and 2 we take $\mu = m_1 m_2/12$ and $\mu = m_1 m_2/500$ respectively.

In Figure 3, Problem 1 (in image size of 128×128) is solved with MG1, MG1m, MG2 and AOS. Even for this small image, we give these computational details to get an impression of these methods: MG1 takes 15 iterations (MG cycles) with CPU time of 12 seconds. MG1m takes 10 iterations with CPU time of 9 seconds. MG2 takes 10 iterations with CPU time of 10 seconds. In the last row final results with AOS are obtained in 230 iterations with CPU time of 64 seconds. Clearly all segmented images are similar to each other while all MG algorithm performances are similar to each other and are faster

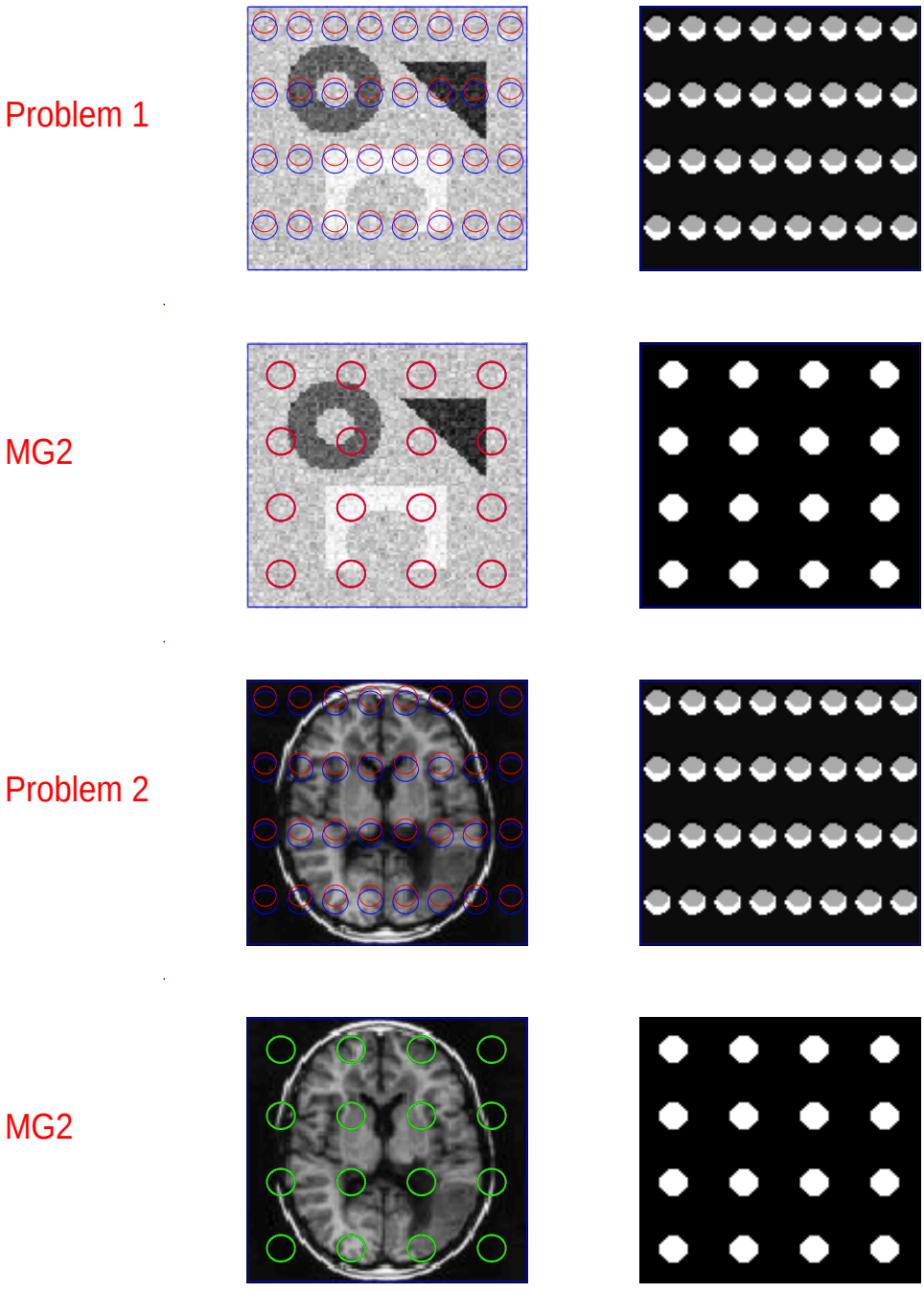

Fig. 2: Test Problems 1 and 2 with the initial guess contours for MG1, MG1m, AOS methods. For MG2, the initial guesses are for a two-phase model.

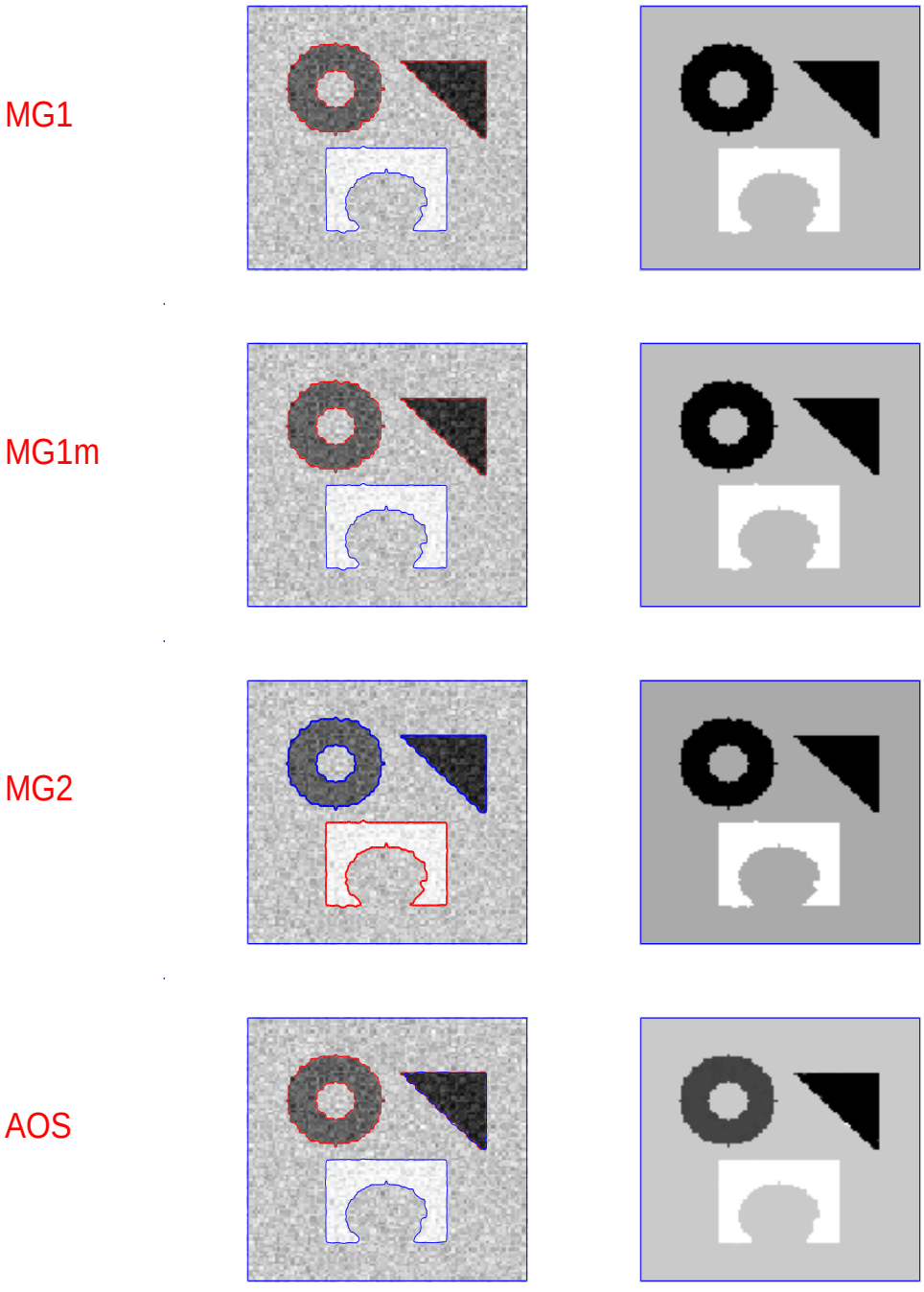

Fig. 3: Problem 1 solved by MG1, MG1m, MG2, AOS methods. Top row: results with MG1, row 2: results with MG1m, row 3: the results with MG2 and row 4: results with AOS. In row 3 the left image is the three phase segmentation and the right image is the final segmented image.

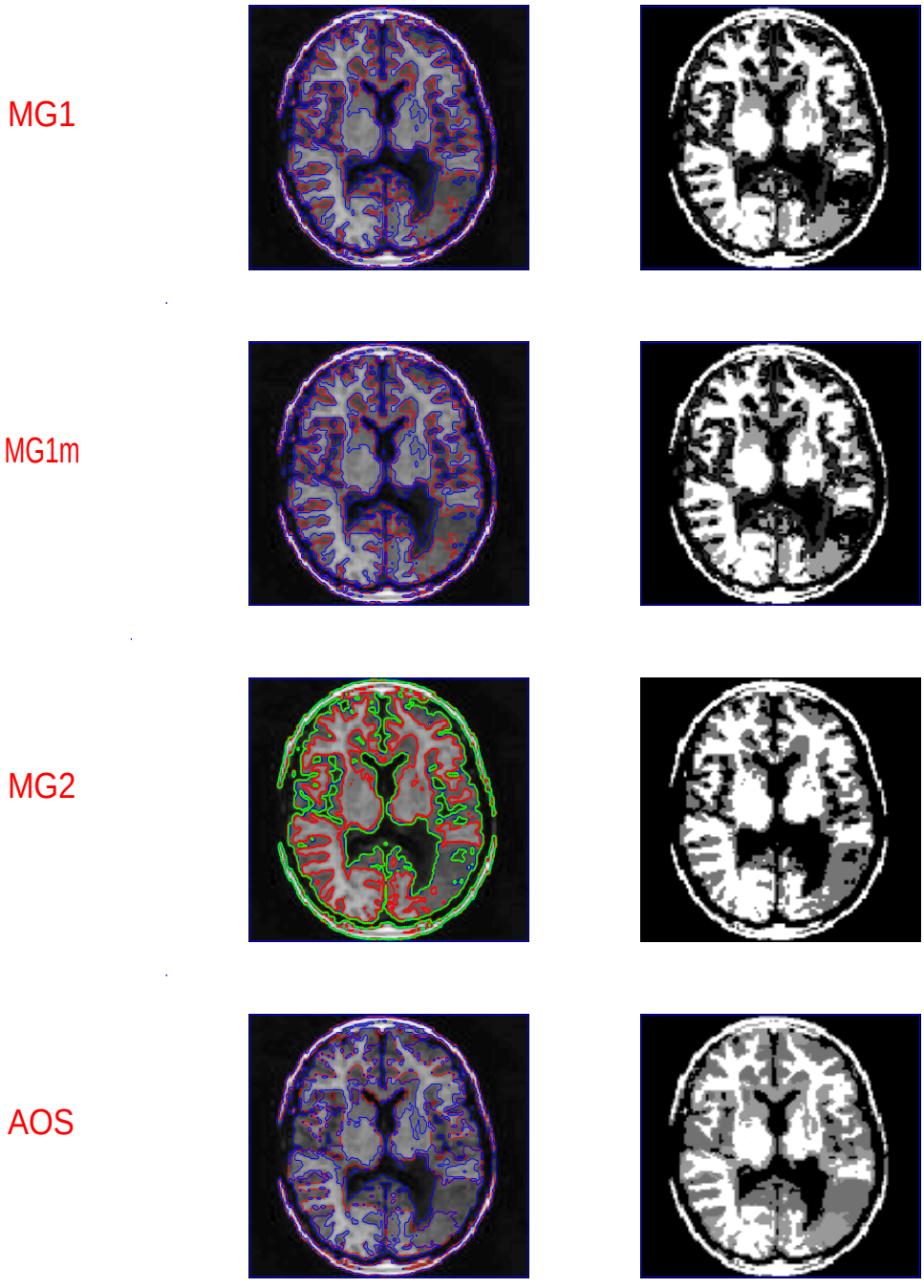

Fig. 4: Problem 2 solved by MG1, MG1m, MG2, AOS methods. As with Figure 3, Top row: results with MG1, row 2: results with MG1m, row 3: the results with MG2 and row 4: results with AOS. In row 3 the left image is the three phase segmentation and the right image is the final segmented image.

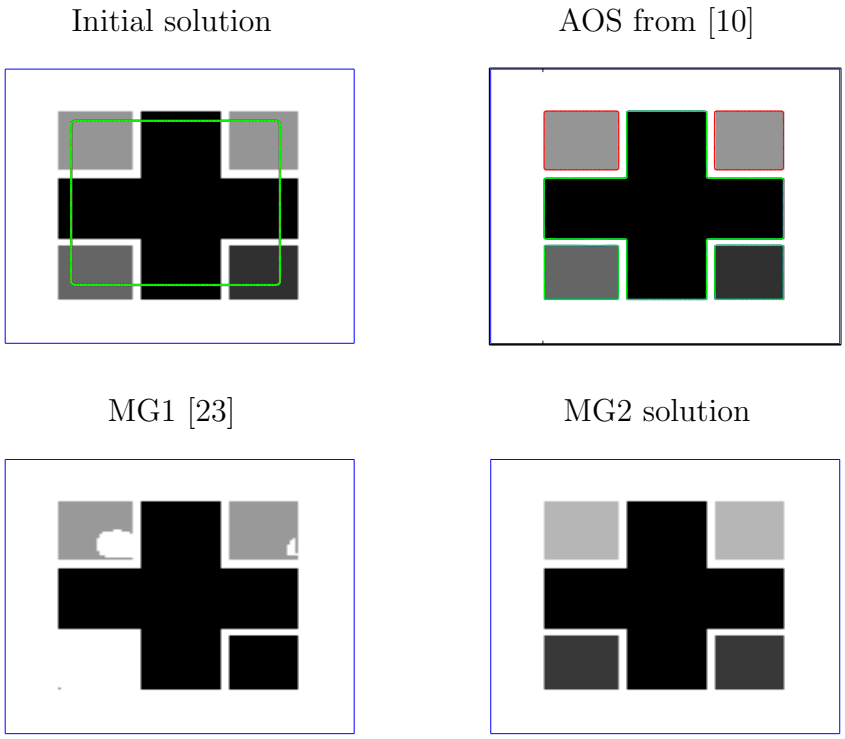

Fig. 5: Problem 3 to illustrate reliance on the initial solutions: clearly AOS (initialized as in Fig.2, only giving 3 phases) and the MG1 (initialized as the top left, without convergence) failed while MG2 (bottom right) segments 4 phases correctly.

than AOS. In Figure 4, Problem 2 is solved with MG1, MG1m, MG2 and AOS. A similar summary can be made.

Performance comparison. In Table 3 we compare the methods discussed in this paper, by CPU times tested on the image in Fig 1 in different (larger) sizes. With MG2, in the 8th column of the table we use the notation $p(q)$ implying p iterations used for the first segmentation and q iterations used for the second segmentation. One observes that, even for images of small sizes, there is some benefit in using MG algorithms. However for large sizes, one could see a huge difference in CPU times, with MG algorithms outperforming the AOS by many magnitudes.

Finally we show one example of testing initial solutions in Fig.5 where we can see that our MG2 converges correctly for the same initial solutions with which both the AOS (i.e. algorithm [10] alone) and the MG1 failed to converge to the correct result.

6 Conclusions

In this paper we have introduced two multigrid algorithms for multiphase variational image segmentation. As expected of a multigrid method, both algorithms are much faster than the Additive Operator Splitting (AOS) method. To deliver an acceptable segmentation, MG1 requiring less level set functions can be dependent of initial guesses

Tab. 3: Comparison of MG1, MG1m and MG2 with AOS and SA (Smoother alone) methods in number of iterations ('Itr') and CPU time ('CPU'). Here '—' implies no convergence (to the tolerance) is yet achieved after 24 hours.

Image Size	AOS		SA		MG1		MG1m		MG2	
	Itr	CPU	Itr	CPU	Itr	CPU	Itr	CPU	Itr	CPU
128×128	80	22	50	3	3	5	2	2	2(2)	11
256×256	150	193	300	54	4	13	2	7	2(2)	15
512×512	1500	42600	1350	975	4	74	2	33	3(3)	43
1024×1024	—	—	1450	3854	4	525	2	148	3(3)	154

while MG2 requiring more level set functions is practically not dependent of initial guesses. Future work will consider models that require only one level set function [2, 14] as well as other models [5, 8].

References

[1] C. Alvino, G. Unal, G. Slabaugh, B. Peny and T. Fang, Efficient segmentation based on Eikonal and diffusion equations, *Inter. J. Computer Math.*, 84(9):1309-1324, 2007.

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

[3] N. Badshah and Ke Chen, On a multilevel technique for image segmention by the Chan-Vese active coutours method, *Comm. Comput. Phys.*, 4(2):294-316, 2008.

[4] A. Brandt, Multi-Level Adaptive Solutions to BVPs, *Math. Comp.*, 31:333-390, 1977.

[5] V. Caselles, R. Kimmel and G. Sapiro. Geodesic active contours. *Int. J. Computer Vision*, 22(1):61-79, 1997.

[6] T. F. Chan and L. A. Vese, Active contour without edges. *IEEE Trans. Image Proc.*, 10(2):266-277, 2001.

[7] T. F. Chan, Ke Chen and J. L. Carter, Iterative methods for solving the dual formulation arising from image restoration. *Electronic Transactions on Numerical Analysis*, 26:299-311, 2007.

[8] T. F. Chan and J. H. Shen, *Image Processing and Analysis—Variational, PDE, wavelet, and stochastic methods*, SIAM Publications, Philadelphia, USA, 2005.

[9] Ke Chen, Matrix Preconditioning Techniques and Applications. *Cambridge University Press, UK*, 2005.

[10] M. Jeon, M. Alexander, W. Pedrycz and N. Pizzi, Unsupervised hierarchical image segmentation with level set and additive operator splitting. *Pattern Recognition Letters*, 26:1461–1469, 2005.

[11] Y. M. Jung, S. H. Kang and J. H. Shen, Multiphase image segmentation via Modica–Mortola phase transition, *SIAM J. Appl. Math.*, 67(5):1213-1232, 2007.

[12] M. Kass, A. Witkin and D. Terzopoulos, Snakes: active contour models. *Int. J. Computer Vision*, 1(4):321-331, 1978.

[13] A. Kenigsberg, R. Kimmel and I. Yavneh, A multigrid approach for fast geodesic active contours. Technion - Computer Science Department - Technical Report CIS-2004-06, Israel, 2004.

[14] H. W. Li and Xue-Cheng Tai, Piecewise constant level set methods for multiphase motion, *Int. J. Numer. Anal. Modelling*, 4(2):291-305, 2007.

[15] J. M. Moral and S. Solimini, Variational methods in image segmentation. In: *Progress in Nonlinear Differential Equations and Their Applications*, Birkhauser, Basel, 1995.

[16] D. Mumford and J. Shah, Optimal approximation by piecewise smooth functions and associated variational problem. *Comm. Pure Appl. Math.*, 42:577-685, 1989.

[17] S. Osher and J. A. Sethian, Fonts propagating with curvature dependent speed: Algorithm based on Hamilton-Jacobi formultions. *Journal of Computational Physics*, 79:12-49, 1998.

[18] G. Papandreou and P. Maragos, A fast multigrid implicit algorithm for the evolution of geodesic active contours. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'04)*, Washington DC, 2:689-694, 2004.

[19] G. Papandreou and P. Maragos, Multigrid geometric active contour models. *IEEE Trans. Image Proc.*,16 (1):229-240, 2007.

[20] L. I. Rudin, S. Osher and E. Fatemi, Nonlinear total variation based noise removal algorithm, *Physica D*, 60:259-268, 1992.

[21] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, UK, 2nd Edition, 1999.

[22] U. Trottenberg, C. W. Oosterlee and A. Schuller. *Multigrid*, Academic Press, London, UK, 2001.

[23] 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(3):271-293, 2002.

[24] J. Weickert, B. M. ter Haar Romeny and M. A. Viergever, Efficient and reliable schemes for nonlinear diffusion filtering, *IEEE Trans. Image Proc.*, 7(3):398-412,1998.

[25] C. Y. Xu and J. L. Prince, Snakes, shapes and gradient vector flow, *IEEE Trans. Image Proc.*, 7(3):359-369, 1998.