Electronic Transactions on Numerical Analysis
http://etna.mcs.kent.edu/

An Optimization Based Multilevel Algorithm for Variational Image Segmentation Models*

Abdul K. Jumaat† and Ke Chen‡

Abstract

Variational active contour models have become very popular in recent years, especially global variational models which segment all objects in an image. Given a set of user-defined prior points, selective variational models aim to segment selectively one object only. We are concerned with fast solution of the latter models. Time marching methods with semi-implicit schemes (gradient descents) or additive operator splitting are used frequently to solve the resulting Euler Lagrange equations derived from these models. For images of moderate size, such methods are effective. However, to process images of large size, urgent need exists in developing fast iterative solvers. Unfortunately geometric multigrid methods do not converge satisfactorily. Here we propose an optimization based multilevel algorithm for efficiently solving a class of selective segmentation models. It also applies to solution of global segmentation models. In level set function formulation, our first variant of the proposed multilevel algorithm has the expected optimal $O(N \log N)$ efficiency for an image of size $n \times n$ with $N = n^2$. However modified localized models are proposed to exploit the local nature of segmentation contours and consequently our second variant after modification is up to practically super-optimal efficiency $O(\sqrt{N} \log N)$. Numerical results show that good segmentation quality is obtained and, as expected, excellent efficiency is observed in reducing computational time.

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

Key words: Active contours, image segmentation, level sets, multilevel, optimization methods, energy minimization

1 Introduction

Image segmentation can be defined as the process of separating objects from their surroundings. The principal goal of segmentation is to partition an image into homogenous regions, which connect spatially groups of pixels called classes, or subsets, with respect to one or more characteristics or features.

Different models and techniques have been developed so far, including histogram analysis and thresholding [26, 36], region growing [2], edge detection and active contours [3, 15]. Of all these techniques, variational techniques [15, 31] are proven to be very efficient for extracting homogeneous areas compared with other models such as statistical methods [17, 16, 18] or wavelet techniques [23, 27].

Segmentation models can be classified into two categories, namely, edge based and region based models; other models may mix these categories. Edge based models refer to the models that are able to drive the contours towards image edges by influence of an edge detector function used. The snake algorithm proposed by Kass et al [24] was the first edge based variational models for image segmentation. Further improvement on the algorithm with Geodesic Active

*Center for Mathematical Imaging Techniques and Department of Mathematical Sciences, University of Liverpool, United Kingdom. Web: http://www.liv.ac.uk/cmit

†Email: abdulkj@liverpool.ac.uk

‡Corresponding author's Web: http://www.liverpool.ac.uk/~cmchenke, email: k.chen@liverpool.ac.uk

Contours and the level-set formulation led to effective models [10, 37]. Region-based segmentation techniques try to separate all pixels of an object from its background pixels based on the intensity and hence find image edges between regions satisfying different homogeneity criteria. Examples of region-based techniques are Region growing [22, 7], Watershed algorithm [22, 8], Thresholding [22, 40], and Fuzzy clustering [38]. The most celebrated and efficient variational model for the images with and without noise is the Mumford-Shah [31] model, reconstructing the segmented image as a piecewise smooth intensity function. Since the model cannot be implemented directly and easily, it was often approximated. The Chan-Vese (CV) [15] model is simplified and reduced from [31], without approximation. The simplification is to replace the piecewise smooth function by a piecewise constant function and, in the case of two phases, the piecewise constant function divides an image into the foreground and the background.

Segmentation models described above are for global segmentation due to the fact that all features or objects in an image are to be segmented. In reality, not all objects can be identified in general because of non-convexity of such models. There exist many studies of these models. For the convex CV model [14], once discretised, the optimization problem can be solved by fast graph cut type method with $O(N \log N)$ efficiency (at the level of a multigrid method) for an image sized $n \times n$ with $N = n^2$ [41, 42].

This paper is concerned with another type of image segmentation models, namely selective segmentation. They are defined as the process of extracting one object of interest in an image based on some known geometric constraints [20, 35, 39]. Two effective models are Badshah-Chen [6] and Rada-Chen [35] which used a mixture of edge-based and region-based ideas in addition to imposing constraints. Recently, a convex selective variational image segmentation model called as Convex Distance Selective Segmentation was successfully proposed by Spencer and Chen [39]. The convex model allowing a global minimiser to be found independently of initialisation [39, 14]. The additive operator splitting (AOS) method with a balloon force term (suitable for images of moderate size, faster than gradient type methods) was proposed for such models. However, to process images of large size, urgent need exists in developing fast multilevel methods.

Both the multilevel and multigrid methods are developed using the idea of hierarchy of discretizations. However, a multilevel method is based on discretize-optimize scheme (algebraic) where minimization of a variational problem is solved directly without using a partial differential equation (PDE). In contrast, a multigrid method is based on optimize-discretize scheme (geometric) where it solves a PDE numerically. The two methods are inter-connected since both can have geometric interpretations and use similar inter-level information transfers.

The latter multigrid methods have been used to solve a few variational image segmentation models in the level set formulation. For geodesic active contours models, linear multigrid methods have been developed [25, 33, 34]. In 2008, Badshah and Chen [4] have successfully implemented a multigrid method to solve the Chan-Vese nonlinear elliptical partial differential equation. In 2009, Badshah and Chen [5] also have developed two multigrid algorithms for modelling variational multiphase image segmentation. While the practical performance of these methods is good, however, they are sensitive to parameters and hence not effective, mainly due to non-smooth coefficients which lead to smoothers not having an acceptable smoothing rate (which in turn are due to jumps or edges that separate segmented domains). Therefore the above multigrid methods behave like the cascadic multigrids [30] where only one multigrid cycle is needed.

Here we pursue the former type of optimization based multilevel methods, based on a discretize-optimize scheme where the minimization is solved directly (without using PDEs). The idea has been applied to other image problems in denoising and debluring [11, 12, 13], not yet to segmentation problems. However, the method is found to get stuck to local minima due to non-differentiability of the energy functional. To overcome that situation, Chan and Chen [11] have proposed the “patch detection" idea in the formulation of the multilevel method which

is efficient for image denoising problems. However, as image size increases, the method can be slow because of the patch detection idea searches the entire image for the possible patch size on the finest level after each multilevel cycle.

In this work, we will consider a differentiable form of variational image segmentation models and develop the multilevel algorithm for the resulting models without using a "patch detection" idea. We are not aware of any similar work on multilevel algorithms for segmentation models in the level set formulation. The key finding is that the resulting multilevel algorithm converges, while not very sensitive to parameter choices, unlike geometric multigrid methods [5] which are known to have problems in convergence.

The rest of the paper is organized in the following way. In Section 2, we briefly review the global segmentation model namely the Chan-Vese model [15] and two selective segmentation models which are Badshah-Chen model [6] and Rada-Chen model [35]. In Section 3, we present an optimization based multilevel algorithm for the selective segmentation models. In Section 4, we propose localized segmentation models and further present multilevel methods for solving them. In Section 5, we give some experimental results to test the presented algorithms. We compare the new methods to the previously fast methods from the literature namely the AOS method for Badshah-Chen [6] and Rada-Chen [35] models (since multigrid methods are not yet developed for these models). However, a multiscale AOS method (for Badshah-Chen [6] and Rada-Chen [35] models) based on the pyramid idea is implemented and included in the comparison. We conclude the paper in Section 6.

2 Review of three existing models

In this section we will first introduce the global segmentation model [15] because it provides the foundation for the selective segmentation models as well as a method for minimizing the associated functional. Next, we will discuss two selective segmentation models by Badshah-Chen [6] and Rada-Chen [35] before we address the fast solution issue for these models.

2.1 The Chan-Vese model

The Chan and Vese (CV) model [15] considers a special case of the piecewise constant Mumford-Shah functional [31] where it is restricted to only two phases (i.e. constants), representing the foreground and the background of the given image $z(x, y)$.

Assume that z is formed by two regions of approximately piecewise constant intensities of distinct (unknown) values c_1 and c_2 , separated by some (unknown) curve or contour Γ. Let the object to be detected be represented by the region Ω_1 with the value c_1 inside the curve Γ whereas outside Γ, in $\Omega_2 = \Omega \backslash \Omega_1$, the intensity of z is approximated with the value c_2. Then, with $\Omega = \Omega_1 \cup \Omega_2$, the Chan-Vese model minimizes the following functional

$$\min_{\Gamma, c_1, c_2} F_{CV}(\Gamma, c_1, c_2) = \mu \text{ length}(\Gamma) + \lambda_1 \int_{\Omega_1} (z - c_1)^2 dxdy + \lambda_2 \int_{\Omega_2} (z - c_2)^2 dxdy. \quad (1)$$

Here, the constants c_1 and c_2 are viewed as the average values of z inside and outside the variable contour Γ. The fixed parameters μ, λ_1, and λ_2 are non-negative but to be specified. In order to minimize equation (1), they applied the level set method [15] , where the unknown curve Γ is represented by the zero level set of the Lipschitz function such that

$$\begin{aligned} \Gamma &= \{(x, y) \in \Omega : \phi(x, y) = 0\}, \\ \Omega_1 &= \text{inside}(\Gamma) = \{(x, y) \in \Omega : \phi(x, y) > 0\}, \\ \Omega_2 &= \text{outside}(\Gamma) = \{(x, y) \in \Omega : \phi(x, y) < 0\}. \end{aligned}$$

To simplify the notation, denote the regularized versions of the Heaviside function and the

Dirac delta function, respectively, by

$$H(\phi(x,y)) = \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan\left(\frac{\phi}{\varepsilon}\right)\right) \qquad \text{and} \qquad \delta(\phi(x,y)) = \frac{\varepsilon}{\pi(\varepsilon^2+\phi^2)}.$$

Thus equation (1) becomes

$$\begin{aligned}\min_{\phi,c_1,c_2} F_{CV}(\phi,c_1,c_2) = \mu\int_\Omega |\nabla H(\phi)|\,dxdy + \lambda_1\int_\Omega (z-c_1)^2 H(\phi)\,dxdy \\ +\lambda_2\int_\Omega (z-c_2)^2(1-H(\phi))\,dxdy.\end{aligned} \tag{2}$$

Keeping the level set function ϕ fixed and minimizing (2) with respect to c_1 and c_2, we have

$$c_1(\phi) = \frac{\int_\Omega z(x,y)\,H(\phi)\,dxdy}{\int_\Omega H(\phi)\,dxdy}, \qquad c_2(\phi) = \frac{\int_\Omega z(x,y)\,(1-H(\phi))\,dxdy}{\int_\Omega (1-H(\phi))\,dxdy}. \tag{3}$$

After that, by fixing constants c_1 and c_2 in $F_{CV}(\phi,c_1,c_2)$, first variation with respect to ϕ yields the following Euler-Lagrange equation:

$$\begin{cases} \mu\delta(\phi)\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \lambda_1\delta(\phi)(z-c_1)^2 + \lambda_2\delta(\phi)(z-c_2)^2 = 0, \quad in \quad \Omega \\ \frac{\delta(\phi)}{|\nabla\phi|}\,\frac{\partial u}{\partial\vec{n}} = 0, \quad on \quad \partial\Omega. \end{cases} \tag{4}$$

Notice that the nonlinear coefficient in equation (4) may have a zero denominator, so the equation is not defined in such cases. A commonly-adopted idea to deal with $|\nabla\phi| = 0$ was to introduce a small positive parameter β to (2) and (4), so the new Euler Lagrange equation becomes

$$\begin{cases} \mu\delta(\phi)\nabla\cdot\left(\frac{\nabla\phi}{\sqrt{|\nabla\phi|^2+\beta}}\right) - \lambda_1\delta(\phi)(z-c_1)^2 + \lambda_2\delta(\phi)(z-c_2)^2 = 0, \quad in \quad \Omega \\ \frac{\delta(\phi)}{|\nabla\phi|}\,\frac{\partial u}{\partial\vec{n}} = 0, \quad on \quad \partial\Omega. \end{cases}$$

where corresponds to minimizing the following differentiable energy function, instead of (2)

$$\begin{aligned}\min_{\phi,c_1,c_2} F_{CV}(\phi,c_1,c_2) = \mu\int_\Omega \sqrt{|\nabla H(\phi)|^2+\beta}\,dxdy + \lambda_1\int_\Omega (z-c_1)^2 H(\phi)\,dxdy \\ +\lambda_2\int_\Omega (z-c_2)^2(1-H(\phi))\,dxdy.\end{aligned} \tag{5}$$

2.2 The Badshah-Chen model

The selective segmentation model by Badshah-Chen (BC) [6] combines the edge based model of Gout et al [20, 21] with intensity fitting terms of Chan-Vese [15]. For image $z(x,y)$ with a marker set $\mathcal{A} = \{w_i = (x_i^*, y_i^*) \in \Omega, 1 \le i \le n_1\} \subset \Omega$ of n_1 geometrical points on or near the target object [35, 43], the selective segmentation idea tries to detect the boundary of a single object among all homogeneity intensity objects in Ω closest to $\mathcal{A}$; here $n_1 \ge 3$. The geometrical points in $\mathcal{A}$ define an initial polygonal contour and guide its evolution towards Γ [43].

The BC minimization equation [6] is given by

$$\begin{aligned}\min_{\Gamma,c_1,c_2} F_{BC}(\Gamma,c_1,c_2) = \mu\int_\Gamma d(x,y)\,g(|\nabla z(x,y)|)\,dxdy \;+ \\ \lambda_1\int_{\text{inside}(\Gamma)} (z-c_1)^2\,dxdy + \lambda_2\int_{\text{outside}(\Gamma)} (z-c_2)^2\,dxdy.\end{aligned} \tag{6}$$

In this model, the function $g(|\nabla z|) = \frac{1}{1+\eta|\nabla z(x,y)|^2}$ is an edge detector which helps to stop the evolving curve on the edge of the targeted object. The strength of detection is adjusted by a

parameter η. The function $g(|\nabla z|)$ is constructed to take small values near to 0 near object edges and large values near to 1 in flat regions. The $d(x,y)$ is a marker distance function which is close to 0 when approaching the points from marker set, given as:

$$d(x,y) = \text{distance}((x,y),\mathcal{A}) = \prod_{i=1}^{n_1}\left(1 - e^{-\frac{(x-x_i^*)^2}{2\kappa^2}} - e^{-\frac{(y-y_i^*)^2}{2\kappa^2}}\right), \quad \forall (x,y) \in \Omega$$

where κ is a positive constant. Alternative distance functions $d(x,y)$ are also possible [35, 43]. Using a level set formulation, the functional (6) becomes

$$\begin{aligned}\min_{\phi,c_1,c_2} F_{BC}(\phi,c_1,c_2) = \mu\int_\Omega d(x,y)g(|\nabla z(x,y)|)|\nabla H(\phi)|\,dxdy \\ +\lambda_1\int_\Omega (z-c_1)^2 H(\phi)\,dxdy + \lambda_2\int_\Omega (z-c_2)^2(1-H(\phi))\,dxdy.\end{aligned} \tag{7}$$

Keeping the level set function ϕ fixed and minimizing (6) with respect to c_1 and c_2, we have

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

Finally keeping constants c_1 and c_2 fixed in $F_{BC}(\phi,c_1,c_2)$, and the following Euler-Lagrange equation for ϕ is derived:

$$\begin{cases} \mu\delta(\phi)\nabla\cdot dg\left(\frac{\nabla\phi}{\sqrt{|\nabla\phi|^2+\beta}}\right) - \lambda_1\delta(\phi)(z-c_1)^2 + \lambda_2\delta(\phi)(z-c_2)^2 = 0, \quad in \quad \Omega \\ dg\frac{\delta(\phi)}{|\nabla\phi|}\ \frac{\partial u}{\partial\vec{n}} = 0, \quad on \quad \partial\Omega. \end{cases} \tag{8}$$

The small positive parameter β is introduced to avoid singularities in (8) which corresponds to minimizing the following differentiable form of the BC model in replace of (7)

$$\begin{aligned}\min_{\phi,c_1,c_2} F_{BC}(\phi,c_1,c_2) = \mu\int_\Omega G(x,y)\sqrt{|\nabla H(\phi)|^2+\beta}\,dxdy \\ +\lambda_1\int_\Omega (z-c_1)^2 H(\phi)\ dxdy + \lambda_2\int_\Omega (z-c_2)^2(1-H(\phi))\ dxdy\end{aligned} \tag{9}$$

where $G = d(x,y)g(x,y)$. To encourage faster convergence, a balloon force term $\alpha G|\nabla\phi|$ is added to the (8) as used by [6].

2.3 The Rada-Chen model

The Rada-Chen (RC) model [35] imposes a further constraint on Ω_1 to ensure that its area is closest to the internal area defined by the marker set. From the polygon formed by the geometrical points in the set $\mathcal{A}$, denote by A_1 and A_2 respectively the area inside and outside the polygon. They compute A_1 and A_2 to approximate the area of the object they try to capture. The RC model also incorporates the similar edge detection function as in the BC model into the regularization term. The energy minimization problem is given by

$$\begin{aligned}\min_{\Gamma,c_1,c_2} F_{RC}(\Gamma,c_1,c_2) = \mu\int_\Gamma g(|\nabla z(x,y)|)\ dxdy + \lambda_1\int_{\Omega_1}(z-c_1)^2 dxdy \\ +\lambda_2\int_{\Omega_2}(z-c_2)^2 dxdy + \nu\left(\int_{\Omega_1} dxdy - \mathrm{A}_1\right)^2 + \nu\left(\int_{\Omega_2} dxdy - \mathrm{A}_2\right)^2.\end{aligned} \tag{10}$$

Rewriting (10) in level-set formulation as in (3), we arrived at the following Euler-Lagrange equation for ϕ:

$$\begin{cases} \mu\delta(\phi)\nabla\cdot g\left(\frac{\nabla\phi}{\sqrt{|\nabla\phi|^2+\beta}}\right) - \lambda_1\delta(\phi)(z-c_1)^2 + \lambda_2\delta(\phi)(z-c_2)^2 \\ -\nu\delta(\phi)\left[\left(\int_\Omega H(\phi)\ dxdy - \mathrm{A}_1\right) - \left(\int_\Omega (1-H(\phi))\,dxdy - \mathrm{A}_2\right)\right] = 0, \quad in \quad \Omega \\ dg\frac{\delta(\phi)}{|\nabla\phi|}\ \frac{\partial u}{\partial\vec{n}} = 0, \qquad on \quad \partial\Omega. \end{cases} \tag{11}$$

As with the BC model, in the actual implementation of the RC model, the small positive parameter β is introduced to avoid singularities in (11) where corresponds to minimizing the following differentiable form of the RC model

$$\begin{aligned}\min_{\phi,c_1,c_2} F_{RC}(\phi,c_1,c_2) = \mu\int_\Omega g(|\nabla z(x,y)|)\sqrt{|\nabla H(\phi)|^2+\beta}\,dxdy + \\ \lambda_1\int_\Omega (z-c_1)^2 H(\phi)\,dxdy + \lambda_2\int_\Omega (z-c_2)^2(1-H(\phi))\,dxdy + \\ \nu\left(\int_\Omega H(\phi)\,dxdy - \mathrm{A}_1\right)^2 + \nu\left(\int_\Omega (1-H(\phi))\,dxdy - \mathrm{A}_2\right)^2.\end{aligned} \tag{12}$$

To encourage faster convergence, they also added the balloon force term which is defined as $\alpha g(x,y)|\nabla\phi(x,y)|$ to (11).

We will use the term **BC0** and **RCO** to refer the AOS algorithm previously used to solve BC model and RC model in [6] and [35] respectively

Of course, it is known that such AOS method is not designed for processing large image. To assist AOS, a pyramid method can be used. The basic idea in a pyramid method is, in the process of curve evolution, the pyramid scheme is used to decompose an image into different scale images and then coarse segmentation is performed on the coarse-scale image using the AOS method instead of directly using the original-size image. Then, the segmentation result is interpolated and adopted as an initial contour for the fine-scale image, thus gradually optimizing the contour and reaching the final segmentation result.

We refer the pyramid method for BC and RC models as **BCP** and **RCP** respectively.

The above forms of variational models, the BC model (9), the RC model (12) respectively, will be conveniently solved by our new proposed multilevel scheme shortly. The **BC0**, **RCO**, **BCP** and **RCP** will be used as comparison methods to our method in segmenting large images.

As remarked before, the reason for seeking alternative optimization based multilevel methods instead of applying a geometric multigrid method is that there are no effective smoothers for the latter case and consequently there exist no converging multigrid methods for the Euler-lagrange equations for our variational models.

3 An $O(N\log N)$ optimization based multilevel algorithm

The main objective of this section is to present the first version of our multilevel formulation for two selective segmentation models: the BC model [6] and the RC model [35]. This section provides the foundation for the development of our main multilevel algorithm for the localized versions of these models. For simplicity for a given image of size $n\times n$, we shall assume $n=2^L$. The standard coarsening defines $L+1$ levels: $k=1$ (finest) $,2,...,L,\ L+1$ (coarsest) such that level k has $\tau_k\times\tau_k$ "superpixels" with each "superpixel" having pixels $b_k\times b_k$, where $\tau_k=n/2^{k-1}$ and $b_k=2^{k-1}$. Figure 2(a-e) show the case of $L=4,\ n=2^4$ for an 16×16 image with 5 levels: level 1 has each pixel of the default size of 1×1 while the coarsest level 5 has a single superpixel of size 16×16. If $n\neq 2^L$, the multilevel method can still be developed with some coarse level superpixels of square shapes and the rest of rectangular shapes.

3.1 Multilevel algorithm for the BC model

Our goal is to solve (9), i.e. the BC model [6] , using a multilevel method in a discretize-optimize scheme.

Before we proceed further, one may question how to discretize the total variation (TV) term in the form

$$TV(u)=\int_\Omega |\nabla u|\,dxdy$$

TV is most often discretized by

$$\begin{aligned} TV_d(u) &= \sum_{i=1}^{n-1}\sum_{j=1}^{n-1}\sqrt{\left(\nabla_x^+ u\right)_{i,j}^2 + \left(\nabla_y^+ u\right)_{i,j}^2} \\ &= \sum_{i=1}^{n-1}\sum_{j=1}^{n-1}\sqrt{\left(u_{i+1,j}-u_{i,j}\right)^2 + \left(u_{i,j+1}-u_{i,j}\right)^2} \end{aligned}$$

There are other ways to define discrete TV by finite difference, but the above form is the simplest one according to [29]. In addition, the reason why the form is considered as a discretization of TV relies in the notion of consistency, well known in numerical analysis where if we consider a regular function $U:\mathbb{R}^2\to\mathbb{R}$ and its discretization $(i,j)\mapsto U_h(i,j)=U(ih,jh)$ for $h>0$, we have

$$h^{-1}.\left|\nabla U_h\right|(ih,jh)\to\left|\nabla U(x,y)\right|$$

as $h\to 0,\quad ih\to x$, and $jh\to y$.

Furthermore, central differences are undesirable for TV discretization (in a discretize-optimize approach) because they miss thin structure [19] as the central differences at (i,j) does not depend on $u_{i,j}$:

$$\begin{aligned} TV_d(u) &= \sum_{i=1}^{n-1}\sum_{j=1}^{n-1}\sqrt{\left[\left(\nabla_x^+ u\right)/2+\left(\nabla_x^- u\right)/2\right]^2+\left[\left(\nabla_y^+ u\right)/2+\left(\nabla_y^- u\right)/2\right]^2} \\ &= \sum_{i=1}^{n-1}\sum_{j=1}^{n-1}\sqrt{\left[\left(u_{i+1,j}-u_{i,j}\right)/2+\left(u_{i,j}-u_{i-1,j}\right)/2\right]^2+\left[\left(u_{i,j+1}-u_{i,j}\right)/2+\left(u_{i,j}-u_{i,j-1}\right)/2\right]^2} \\ &= \sum_{i=1}^{n-1}\sum_{j=1}^{n-1}\sqrt{\left[\left(u_{i+1,j}-u_{i-1,j}\right)/2\right]^2+\left[\left(u_{i,j+1}-u_{i,j-1}\right)/2\right]^2}. \end{aligned}$$

To avoid the problem, one sided difference can be used. More discussion on discretizing TV can be found in [19] and [29] and the references therein.

Using the above information, the discretized version of (9) is given by:

$$\begin{aligned} \min_{\phi,c_1,c_2} F_{BC}(\phi,c_1,c_2) &\equiv \min_{\phi,c_1,c_2} F_{BC}^a(\phi_{1,1},\phi_{2,1},\ldots,\phi_{i-1,j},\phi_{i,j},\phi_{i+1,j},\ldots,\phi_{n,n},c_1,c_2) \\ &= \bar{\mu}\sum_{i=1}^{n-1}\sum_{j=1}^{n-1} G_{i,j}\sqrt{\left(H_{i,j}-H_{i,j+1}\right)^2+\left(H_{i,j}-H_{i+1,j}\right)^2+\beta} \\ &+ \quad \lambda_1\sum_{i=1}^{n}\sum_{j=1}^{n}\left(z_{i,j}-c_1\right)^2 H_{i,j}+\lambda_2\sum_{i=1}^{n}\sum_{j=1}^{n}\left(z_{i,j}-c_2\right)^2\left(1-H_{i,j}\right) \end{aligned} \tag{13}$$

where ϕ denotes a row vector, $\bar{\mu}=\dfrac{\mu}{h},\quad h=\dfrac{1}{n-1},\quad c_1=\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{n} z_{i,j}H_{i,j}\Big/\sum_{i=1}^{n}\sum_{j=1}^{n} H_{i,j},$

$$G_{i,j}=G(x_i,y_j),\qquad c_2=\sum_{i,j=1}^{n} z_{i,j}\left(1-H_{i,j}\right)\Big/\sum_{i,j=1}^{n}\left(1-H_{i,j}\right)\quad\text{and}\quad H_{i,j}=\frac{1}{2}+\frac{1}{\pi}\arctan\left(\frac{\phi_{i,j}}{\varepsilon}\right).$$

As a prelude to multilevel methods, consider the minimization of (13) by the coordinate descent method on the finest level 1:

Given $\phi^{(0)}=\left(\phi_{i,j}^{(0)}\right)$ *and set* $m=0$;

$$\textit{Solve } \phi_{i,j}^{(m+1)}=\arg\min_{\phi_{i,j}\in\mathbb{R}} F_{BC}^{loc}(\phi_{i,j},c_1,c_2) \textit{ for } i,j=1,2,...,n; \tag{14}$$

Repeat the above step with $m=m+1$ *until stopped.*

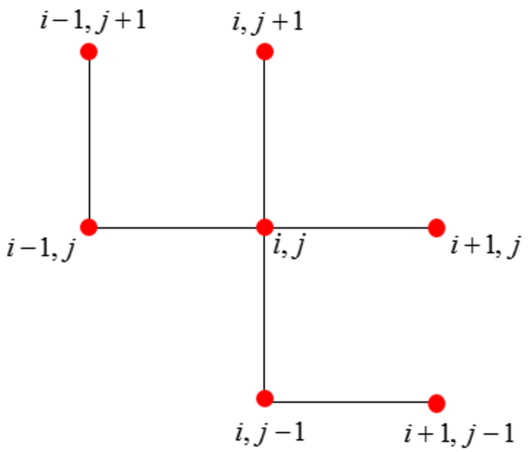


Figure 1: The interaction of $\phi_{i,j}$ at a central pixel (i, j) with neighboring pixels on the finest level 1. Clearly only 3 terms (pixels) are involved with $\phi_{i,j}$ (through regularisation).

Here equation (14) is obtained by expanding and simplifying the main model in (13) i.e.

$$\begin{aligned}
&F_{BC}^{loc}(\phi_{i,j}, c_1, c_2) \\
&\equiv F_{BC}^{a}\left(\phi_{1,1}^{(m-1)}, \phi_{2,1}^{(m-1)}, \ldots, \phi_{i-1,j}^{(m-1)}, \phi_{i,j}, \phi_{i+1,j}^{(m-1)}, \ldots, \phi_{n,n}^{(m-1)}, c_1, c_2\right) - F_{BC}^{(m-1)} \\
&= \bar{\mu}\Big[G_{i,j}\sqrt{\left(H_{i,j} - H_{i+1,j}^{(m)}\right)^2 + \left(H_{i,j} - H_{i,j+1}^{(m)}\right)^2 + \beta} \\
&\quad + G_{i-1,j}\sqrt{\left(H_{i,j} - H_{i-1,j}^{(m)}\right)^2 + \left(H_{i-1,j}^{(m)} - H_{i-1,j+1}^{(m)}\right)^2 + \beta} \\
&\quad + G_{i,j-1}\sqrt{\left(H_{i,j} - H_{i,j-1}^{(m)}\right)^2 + \left(H_{i,j-1}^{(m)} - H_{i+1,j-1}^{(m)}\right)^2 + \beta}\Big] \\
&\quad + \lambda_1(z_{i,j} - c_1)^2 H_{i,j} \; + \lambda_2(z_{i,j} - c_2)^2\,(1 - H_{i,j})
\end{aligned}$$

with Neumann's boundary condition, where $F_{BC}^{(m-1)}$ denotes the sum of all terms in F_{BC}^{a} that do not involve $\phi_{i,j}$. Minimization of c_1, c_2 follows as before. Clearly one seems that this is a coordinate descent method. As such the method will exhibit a functional decay property $F_{BC}^{a}(\phi^{(m)}) \leq F_{BC}^{a}(\phi^{(m-1)})$ from one substep to the next. It should be remarked that the formulation in (14) is based on the work in [9, 11].

Using (14), we illustrate the interaction of $\phi_{i,j}$ with its neighboring pixel on the finest level 1 in Figure 1. We will use this basic structure to develop a multilevel method.

The one-dimensional problem from (14) may be solved by any suitable optimization method – here from $\phi^{(m-1)} \to \phi \to \phi^{(m)}$, we solve its first order condition

$$\begin{aligned}
&\bar{\mu}G_{i,j}\frac{2H_{i,j} - H_{i+1,j}^{(m)} - H_{i,j+1}^{(m)}}{\sqrt{\left(H_{i,j} - H_{i+1,j}^{(m)}\right)^2 + \left(H_{i,j} - H_{i,j+1}^{(m)}\right)^2 + \beta}} + \bar{\mu}G_{i-1,j}\frac{H_{i,j} - H_{i-1,j}^{(m)}}{\sqrt{\left(H_{i,j} - H_{i-1,j}^{(m)}\right)^2 + \left(H_{i-1,j}^{(m)} - H_{i-1,j+1}^{(m)}\right)^2 + \beta}} \\
&+\bar{\mu}G_{i,j-1}\frac{H_{i,j} - H_{i,j-1}^{(m)}}{\sqrt{\left(H_{i,j} - H_{i,j-1}^{(m)}\right)^2 + \left(H_{i,j-1}^{(m)} - H_{i+1,j-1}^{(m)}\right)^2 + \beta}} + \lambda_1(z_{i,j} - c_1)^2 - \lambda_1(z_{i,j} - c_2)^2 = 0.
\end{aligned}$$

As an example, if using the Newton iterations, one gets the form

$$\phi_{i,j}^{new} = \phi_{i,j}^{old} - T^{old}/B^{old} \tag{15}$$

where

$$T^{old} = \bar{\mu}G_{i,j}\frac{2H_{i,j}^{old}-H_{i+1,j}^{(m)}-H_{i,j+1}^{(m)}}{\sqrt{\left(H_{i,j}^{old}-H_{i+1,j}^{(m)}\right)^2+\left(H_{i,j}^{old}-H_{i,j+1}^{(m)}\right)^2+\beta}} + \bar{\mu}G_{i-1,j}\frac{H_{i,j}^{old}-H_{i-1,j}^{(m)}}{\sqrt{\left(H_{i,j}^{old}-H_{i-1,j}^{(m)}\right)^2+\left(H_{i-1,j}^{(m)}-H_{i-1,j+1}^{(m)}\right)^2+\beta}}$$
$$+\bar{\mu}G_{i,j-1}\frac{H_{i,j}^{old}-H_{i,j-1}^{(m)}}{\sqrt{\left(H_{i,j}^{old}-H_{i,j-1}^{(m)}\right)^2+\left(H_{i,j-1}^{(m)}-H_{i+1,j-1}^{(m)}\right)^2+\beta}} + \lambda_1(z_{i,j}-c_1)^2 - \lambda_1(z_{i,j}-c_2)^2,$$

$$B^{old} = \frac{2\bar{\mu}G_{i,j}}{\sqrt{\left(H_{i,j}^{old}-H_{i+1,j}^{(m)}\right)^2+\left(H_{i,j}^{old}-H_{i,j+1}^{(m)}\right)^2+\beta}} - \frac{\bar{\mu}G_{i,j}\left(2H_{i,j}^{old}-H_{i+1,j}^{(m)}-H_{i,j+1}^{(m)}\right)^2}{\sqrt{\left(\left(H_{i,j}^{old}-H_{i+1,j}^{(m)}\right)^2+\left(H_{i,j}^{old}-H_{i,j+1}^{(m)}\right)^2+\beta\right)^{3/2}}}$$
$$+\frac{\bar{\mu}G_{i-1,j}}{\sqrt{\left(H_{i,j}^{old}-H_{i-1,j}^{(m)}\right)^2+\left(H_{i-1,j}^{(m)}-H_{i-1,j+1}^{(m)}\right)^2+\beta}} - \frac{\bar{\mu}G_{i-1,j}\left(H_{i,j}^{old}-H_{i-1,j}^{(m)}\right)^2}{\sqrt{\left(\left(H_{i,j}^{old}-H_{i-1,j}^{(m)}\right)^2+\left(H_{i-1,j}^{(m)}-H_{i-1,j+1}^{(m)}\right)^2+\beta\right)^{3/2}}}$$
$$+\frac{\bar{\mu}G_{i,j-1}}{\sqrt{\left(H_{i,j}^{old}-H_{i,j-1}^{(m)}\right)^2+\left(H_{i,j-1}^{(m)}-H_{i+1,j-1}^{(m)}\right)^2+\beta}} - \frac{\bar{\mu}G_{i,j-1}\left(H_{i,j}^{old}-H_{i,j-1}^{(m)}\right)^2}{\sqrt{\left(\left(H_{i,j}^{old}-H_{i,j-1}^{(m)}\right)^2+\left(H_{i,j-1}^{(m)}-H_{i+1,j-1}^{(m)}\right)^2+\beta\right)^{3/2}}}.$$

To develop a multilevel method of this coordinate descent method, we may interpret solving (14) for a new iterate $\phi_{i,j}^{(m)}$ as looking for the best update (on an old iterate $\phi_{i,j}^{(m-1)}$; here a scalar constant) that minimizes the local merit functional $F_{BC}^{loc}(\phi_{i,j}, c_1, c_2)$. On level 1 the local minimization for c takes the form

$$F_{BC}^{loc}(\phi_{i,j}, c_1, c_2) = F_{BC}^{loc}\left(\phi_{i,j}^{(m)} + c, c_1, c_2\right).$$

Hence, we may rewrite (14) in an equivalent form:

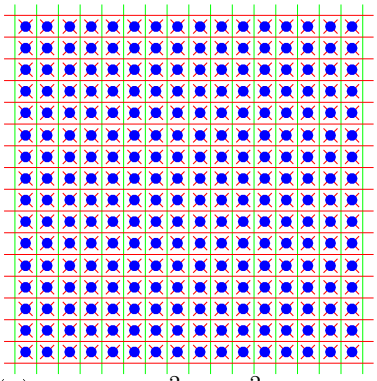
$$\begin{aligned} &\textit{Given } \left(\phi_{i,j}^{(m)}\right) \textit{ with } m = 0, \\ &\textit{Solve } \hat{c} = \underset{c\in\mathbb{R}}{\arg\min}\, F_{BC}^{loc}\left(\phi_{i,j}^{(m)} + c, c_1, c_2\right),\ \phi_{i,j}^{(m+1)} = \phi_{i,j}^{(m)} + \hat{c} \textit{ for } i,j = 1,2,...,n; \\ &\textit{Repeat the above step with } m = m+1 \textit{ until stopped.} \end{aligned} \tag{16}$$

Now consider how the update is done on a general level $k = 2, \ldots, L+1$. Similarly to $k = 1$, we derive the simplified formulation for each of the $\tau_k \times \tau_k$ subproblems, in a block of pixels $b_k \times b_k$ e.g. the multilevel method for k=2 is to look for the best correction constant to update this block so that the underlying merit functional, relating to all four pixels (see Fig.2(b)), achieves a local minimum.

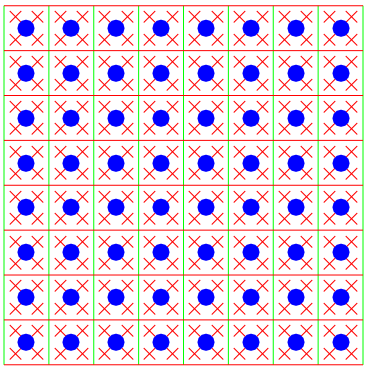
For levels $k = 1, ..., 5$, Figure 2 illustrates the multilevel partition of an image of size 16×16 pixels from (a) the finest level (level 1) until (e) the coarsest level (level 5).

Observe that $b_k\tau_k = n$ on level k, where τ_k is the number of boxes and b_k is the block size. So from Figure 1(a), $b_1 = 1$ and $\tau_1 = n = 16$. On other levels $k = 2, 3, 4$ and 5, we see that block size $b_k = 2^{k-1}$ and $\tau_k = 2^{L+1-k}$ since $n = 2^L$. Based on Figure 1, we illustrate a box $\odot$ interacting with neighboring pixels $\bullet$ in level 3. In addition, Figure 2 (f) illustrates that fact that variation by $c_{i,j}$ inside an active block only involves its boundary of precisely $4b_k - 4$ pixels, not all b_k^2 pixels, in that box, denoted by symbols $\lhd$, $\rhd$, Δ, ∇. This is important in efficient implementation.

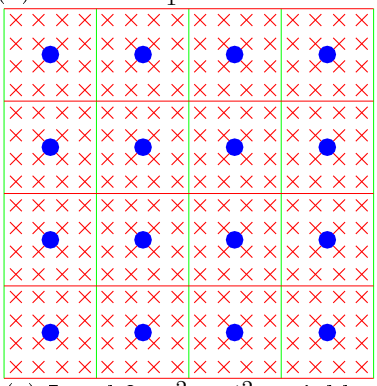
With the above information, we are now ready to formulate the multilevel approach for general level k. Let's set the following: $b = 2^{k-1}$, $k_1 = (i-1)\,b + 1$, $k_2 = ib$, $\ell_1 = (j-1)\,b + 1$, $\ell_2 = jb$, and $c = (c_{i,j})$. Then, the computational stencil involving c on level k can be shown as



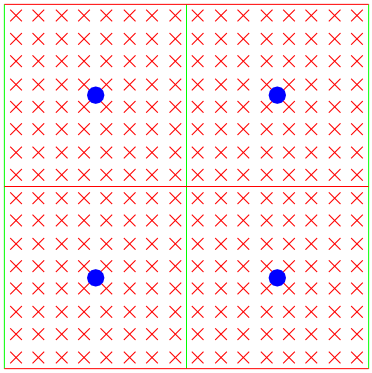
(a) Level 1: $\tau_1^2 = 16^2$ variables



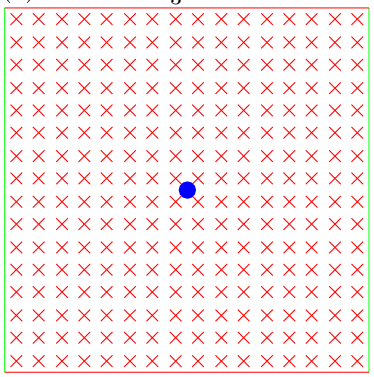
(b) Level 2: $\tau_2^2 = 8^2$ variables



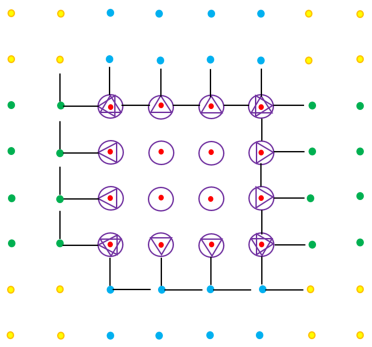
(c) Level 3: $\tau_3^2 = 4^2$ variables



(d) Level 4: $\tau_4^2 = 2^2$ variables



(e) Level 5: $\tau_5^2 = 1$ variable



(f) Level 3 block with $b_3^2 = 16$ pixels but only 12 effective terms in local minimization F_{BC}^{loc}

Figure 2: Illustration of multilevel coarsening. Partitions (a)-(e): the red "×" shows image pixels, while blue • illustrates the variable c. (f) shows on coarse level 3 the difference of inner and boundary pixels interacting with neighboring pixels •. The middle boxes ⊙ indicate the inner pixels which do not involve c, others boundary pixels denoted by symbols $\lhd$, $\rhd$, Δ, ∇ involve c as in (16) via F_{BC}^{loc}.

follows

$$\begin{array}{c|c|c|c}
\vdots & \vdots & \cdots & \vdots \\
\tilde{\phi}_{k_1-1,\ell_2+1}+c_{i-1,j+1} & \tilde{\phi}_{k_1,\ell_2+1}+c_{i,j+1} & \cdots & \tilde{\phi}_{k_2,\ell_2+1}+c_{i,j+1} \quad \tilde{\phi}_{k_2+1,\ell_2+1}+c_{i+1,j+1} \\
\hline
\tilde{\phi}_{k_1-1,\ell_2}+c_{i-1,j} & \tilde{\phi}_{k_1,\ell_2}+c_{i,j} & \cdots & \tilde{\phi}_{k_2,\ell_2}+c_{i,j} \quad \tilde{\phi}_{k_2+1,\ell_2}+c_{i+1,j} \\
\cdots & \vdots & \cdots & \vdots \quad \cdots \\
\tilde{\phi}_{k_1-1,\ell_1}+c_{i-1,j} & \tilde{\phi}_{k_1,\ell_1}+c_{i,j} & \cdots & \tilde{\phi}_{k_2,\ell_1}+c_{i,j} \quad \tilde{\phi}_{k_2+1,\ell_1}+c_{i+1,j} \\
\hline
\tilde{\phi}_{k_1-1,\ell_1-1}+c_{i-1,j-1} & \tilde{\phi}_{k_1,\ell_1-1}+c_{i,j-1} & \cdots & \tilde{\phi}_{k_2,\ell_1-1}+c_{i,j-1} \quad \tilde{\phi}_{k_2+1,\ell_1-1}+c_{i+1,j-1} \\
\vdots & \vdots & \cdots & \vdots \quad \vdots
\end{array} \tag{17}$$

The illustration shown above is consistent with Figure 2 (f) and the key point is that interior pixels (non-boundary pixels) do not involve $c_{i,j}$ in the formulation's first nonlinear term. This is because the finite differences are not changed at interior pixels by the same update as in

$$\begin{aligned}
&\sqrt{\left(\tilde{\phi}_{k,l}+c_{i,j}-\tilde{\phi}_{k+1,l}-c_{i,j}\right)^2+\left(\tilde{\phi}_{k,l}+c_{i,j}-\tilde{\phi}_{k,l+1}-c_{i,j}\right)^2+\beta} \\
&\qquad =\sqrt{\left(\tilde{\phi}_{k,l}-\tilde{\phi}_{k+1,l}\right)^2+\left(\tilde{\phi}_{k,l}-\tilde{\phi}_{k,l+1}\right)^2+\beta}.
\end{aligned}$$

Then, as a local minimization for c, the problem (16) is equivalent to minimizing the following

$$\begin{aligned}
F_{BC1}\left(c_{i,j}\right) =&\bar{\mu}\sum_{\ell=\ell_1}^{\ell_2} G_{k_1-1,\ell}\sqrt{\left[c_{i,j}-\left(\tilde{\phi}_{k_1-1,\ell}-\tilde{\phi}_{k_1,\ell}\right)\right]^2+\left(\tilde{\phi}_{k_1-1,\ell}-\tilde{\phi}_{k_1-1,\ell+1}\right)^2+\beta} \\
&+\bar{\mu}\sum_{k=k_1}^{k_2-1} G_{k,\ell_2}\sqrt{\left[c_{i,j}-\left(\tilde{\phi}_{k,\ell_2+1}-\tilde{\phi}_{k,\ell_2}\right)\right]^2+\left(\tilde{\phi}_{k,\ell_2}-\tilde{\phi}_{k+1,\ell_2}\right)^2+\beta} \\
&+\bar{\mu}G_{k_2,\ell_2}\sqrt{\left[c_{i,j}-\left(\tilde{\phi}_{k_2,\ell_2+1}-\tilde{\phi}_{k_2,\ell_2}\right)\right]^2+\left[c_{i,j}-\left(\tilde{\phi}_{k_2+1,\ell_2}-\tilde{\phi}_{k_2,\ell_2}\right)\right]^2+\beta} \\
&+\bar{\mu}\sum_{\ell=\ell_1}^{\ell_2-1} G_{k_2,\ell}\sqrt{\left[c_{i,j}-\left(\tilde{\phi}_{k_2+1,\ell}-\tilde{\phi}_{k_2,\ell}\right)\right]^2+\left(\tilde{\phi}_{k_2,\ell}-\tilde{\phi}_{k_2,\ell+1}\right)^2+\beta} \\
&+\bar{\mu}\sum_{k=k_1}^{k_2} G_{k,\ell_1-1}\sqrt{\left[c_{i,j}-\left(\tilde{\phi}_{k,\ell_1-1}-\tilde{\phi}_{k,\ell_1}\right)\right]^2+\left(\tilde{\phi}_{k,\ell_1-1}-\tilde{\phi}_{k+1,\ell_1-1}\right)^2+\beta} \\
&+\lambda_2\sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2}\left(1-H\left(\tilde{\phi}_{k,\ell}+c_{i,j}\right)\right)\left(z_{k,\ell}-c_2\right)^2 \\
&+\lambda_1\sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2}\left(H\left(\tilde{\phi}_{k,\ell}+c_{i,j}\right)\right)\left(z_{k,\ell}-c_1\right)^2.
\end{aligned} \tag{18}$$

For the third term, we may note $\sqrt{(c-a)^2+(c-b)^2+\beta}=\sqrt{2\left(c-\frac{a+b}{2}\right)^2+2\left(\frac{a-b}{2}\right)^2+\beta}$. Further we conclude that the local minimization problem for block (i,j) on level k with

respect to $c_{i,j}$ amounts to minimising the following equivalent functional

$$
\begin{aligned}
F_{BC1}\left(c_{i,j}\right) = \quad & \bar{\mu}\sum_{\ell=\ell_1}^{\ell_2} G_{k_1-1,\ell}\sqrt{(c_{i,j}-h_{k_1-1,\ell})^2+\upsilon_{k_1-1,\ell}^2+\beta} \\
& + \bar{\mu}\sum_{k=k_1}^{k_2-1} G_{k,\ell_2}\sqrt{\left(c_{i,j}-\upsilon_{k,\ell_2}\right)^2+h_{k,\ell_2}^2+\beta} \\
& + \bar{\mu}\sum_{\ell=\ell_1}^{\ell_2-1} G_{k_2,\ell}\sqrt{(c_{i,j}-h_{k_2,\ell})^2+\upsilon_{k_2,\ell}^2+\beta} + \bar{\mu}\sum_{k=k_1}^{k_2} G_{k,\ell_1-1}\sqrt{(c_{i,j}-\upsilon_{k,\ell_1-1})^2+h_{k,\ell_1-1}^2+\beta} \\
& + \bar{\mu}\sqrt{2}G_{k_2,\ell_2}\sqrt{\left(c_{i,j}-\bar{\upsilon}_{k_2,\ell_2}\right)^2+\bar{h}_{k_2,\ell_2}^2+\frac{\beta}{2}} + \lambda_1\sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2} H(\tilde{\phi}_{k,\ell}+c_{i,j})(z_{k,\ell}-c_1)^2 \\
& + \lambda_2\sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2}\left(1-H(\tilde{\phi}_{k,\ell}+c_{i,j})\right)(z_{k,\ell}-c_2)^2
\end{aligned}
\tag{19}
$$

where we have used the following notation (which will be used later also):

$$
\begin{array}{lll}
h_{k,\ell}=\tilde{\phi}_{k+1,\ell}-\tilde{\phi}_{k,\ell}, & \upsilon_{k,\ell}=\tilde{\phi}_{k,\ell+1}-\tilde{\phi}_{k,\ell}, & \upsilon_{k_2,\ell_2}=\tilde{\phi}_{k_2,\ell_2+1}-\tilde{\phi}_{k_2,\ell_2}, \\
h_{k_2,\ell_2}=\tilde{\phi}_{k_2+1,\ell_2}-\tilde{\phi}_{k_2,\ell_2}, & \bar{\upsilon}_{k_2,\ell_2}=\frac{\upsilon_{k_2,\ell_2}+h_{k_2,\ell_2}}{2}, & \bar{h}_{k_2,\ell_2}=\frac{\upsilon_{k_2,\ell_2}-h_{k_2,\ell_2}}{2}, \\
h_{k_1-1,\ell}=\tilde{\phi}_{k_1,\ell}-\tilde{\phi}_{k_1-1,\ell}, & \upsilon_{k_1-1,\ell}=\tilde{\phi}_{k_1-1,\ell+1}-\tilde{\phi}_{k_1-1,\ell}, & \upsilon_{k,\ell_2}=\tilde{\phi}_{k,\ell_2+1}-\tilde{\phi}_{k,\ell_2}, \\
h_{k,\ell_2}=\tilde{\phi}_{k+1,\ell_2}-\tilde{\phi}_{k,\ell_2}, & h_{k_2,\ell}=\tilde{\phi}_{k_2+1,\ell}-\tilde{\phi}_{k_2,\ell}, & \upsilon_{k_2,\ell}=\tilde{\phi}_{k_2,\ell+1}-\tilde{\phi}_{k_2,\ell}, \\
\upsilon_{k,\ell_1-1}=\tilde{\phi}_{k,\ell_1}-\tilde{\phi}_{k,\ell_1-1}, & h_{k,\ell_1-1}=\tilde{\phi}_{k+1,\ell_1-1}-\tilde{\phi}_{k,\ell_1-1}. &
\end{array}
$$

On the coarsest level, we look for a **single** constant update for the current approximation $\tilde{\phi}$ that is $F_{BC1}\left(\tilde{\phi}+c\right)$,

$$
\begin{aligned}
\min_c F_{BC1}(\tilde{\phi}+c) = \bar{\mu}\sum_{i=1}^{n-1}\sum_{j=1}^{n-1} G_{i,j}\sqrt{\left(\tilde{\phi}_{i,j}+c-\tilde{\phi}_{i,j+1}-c\right)^2+\left(\tilde{\phi}_{i,j}+c-\tilde{\phi}_{i+1,j}-c\right)^2+\beta} \\
+\lambda_1\sum_{i=1}^{n}\sum_{j=1}^{n} H(\tilde{\phi}_{i,j}+c)(z_{i,j}-c_1)^2+\lambda_2\sum_{i=1}^{n}\sum_{j=1}^{n}\left(1-H(\tilde{\phi}_{k,\ell}+c_{i,j})\right)(z_{i,j}-c_1)^2
\end{aligned}
$$

which is equivalent to

$$
\begin{aligned}
\min_c \hat{F}_{BC1}(\tilde{\phi}+c) = \lambda_1\sum_{i=1}^{n}\sum_{j=1}^{n} H(\tilde{\phi}_{i,j}+c)(z_{i,j}-c_1)^2 \\
+\lambda_2\sum_{i=1}^{n}\sum_{j=1}^{n}\left(1-H(\tilde{\phi}_{k,\ell}+c_{i,j})\right)(z_{i,j}-c_1)^2.
\end{aligned}
\tag{20}
$$

In general, (19) can be written as $\min_{c_{i,j}\in\mathbb{R}} F_{BC1}\left(\tilde{\phi}+\mathbb{P}_c\right)$ where $\mathbb{P}_c = c\vec{d}$ and $\tilde{\phi}, \vec{d} \in \mathbb{R}^{n^2}$. To interpret our method as a hierarchical gradient descent method, we may view a general update as choosing the best c to solve $\min_c F^a_{BC}(\tilde{\phi}+\mathbb{P}_c)$ where e.g.

level 1, at pixel $(1,1): \vec{d} = (\boxed{1},\ 0,\ \cdots,\ 0;\ 0,\ 0,\cdots,\ 0;\ \cdots;\ \ 0,\ 0,\cdots,\ 0),$

$(2,1): \vec{d} = (0,\ \boxed{1},\ \cdots,\ 0;\ 0,\ 0,\cdots,\ 0;\ \cdots;\ \ 0,\ 0,\cdots,\ 0),$

level 2, superpixel $(1,1): \vec{d} = (\boxed{1,\ 1},\ 0,\ \cdots,0;\ \boxed{1,\ 1},\ 0,\ \cdots,0;\cdots;\ 0,0,0,\cdots,\ 0).$

The solutions of the above local minimization problems , solved using a Newton method as in (15) or a fixed point method for t iterations (inner iteration), defines the updated solution

$\tilde{\phi} = \tilde{\phi} + Q_k c$. Here Q_k is the interpolation operator distributing $c_{i,j}$ to the corresponding $b_k \times b_k$ block on level k as illustrated in (17). Then we obtain a multilevel method if we cycle through all levels and all blocks on each level until the solution converges to the prescribe tolerance, *tol* or reach the prescribe maximum cycle (outer iteration).

So finally our implementation of the proposed multilevel method is then summarized in Algorithm 1. Here Steps $2-3$ simply update $\tilde{\phi}$ from the finest to the coarsest level $k =$

Algorithm 1 BC1 – Multilevel algorithm for the BC model

Given z, an initial guess $\tilde{\phi}$ and the stop tolerance tol with $L+1$ levels,

1) *Iteration starts with $\phi_{old} = \tilde{\phi}$ ($\tilde{\phi}$ contains the initial guess before the first iteration and the updated solution at all later iterations)*

2) *Smooth for t iterations the approximation on the finest level $k = 1$ that is solve* $\min_{\phi_{i,j}} F_{BC}^{loc}(\phi_{i,j}, c_1, c_2)$ or (16) *for $i, j = 1, 2, ...n$*

3) *Iterate for t times on each coarse level $k = 2, 3, ...L, L+1$:*
   - ► *If $k \leq L$, compute the minimizer c of* (19) or solve $\min_{c_{i,j}} F_{BC1}(c_{i,j})$;
   - ► *If $k = L+1$, solve* (20) or $\min_c \hat{F}_{BC1}(\tilde{\phi} + c)$ *on the coarsest level.*

   Add the correction $\tilde{\phi} = \tilde{\phi} + Q_k c$ where Q_k is the interpolation operator distributing $c_{i,j}$ to the corresponding $b_k \times b_k$ block on level k as illustrated in (17).

4) *Return to Step 1 unless $\frac{\|\tilde{\phi} - \phi_{old}\|_2}{\|\tilde{\phi}\|_2} < tol$ or until the prescribed maximum of cycles is reached. Otherwise exit with $\phi = \tilde{\phi}$.*

$1, 2, \ldots, L, L+1$ so they might be viewed as a single step. We will use the term **BC1** to refer the multilevel Algorithm 1.

In this algorithm, we recommend that we start updating our multilevel algorithm in a fast manner is to adjust the fine structure before the coarse level. We found in a separate experiment that if we adjust the coarse structure before the fine level, the convergence is slower.

3.2 Multilevel algorithm for the RC model

Generalization of the above algorithm to other models is much similar. For the RC model, the discretized version of in (12) takes the following form

$$\begin{aligned}\min_{\phi, c_1, c_2} F_{RC}(\phi, c_1, c_2) = \bar{\mu} \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} g_{i,j} \sqrt{(H_{i,j} - H_{i,j+1})^2 + (H_{i,j} - H_{i+1,j})^2 + \beta} \\ + \lambda_1 \sum_{i=1}^{n} \sum_{j=1}^{n} (z_{i,j} - c_1)^2 H_{i,j} + \lambda_2 \sum_{i=1}^{n} \sum_{j=1}^{n} (z_{i,j} - c_2)^2 (1 - H_{i,j}) \\ + \nu \Big(-\mathrm{A}_1 + \sum_{i=1}^{n} \sum_{j=1}^{n} H_{i,j} \Big)^2 + \nu \Big(-\mathrm{A}_2 + \sum_{i=1}^{n} \sum_{j=1}^{n} \Big(1 - H_{i,j}\Big) \Big)^2. \end{aligned} \tag{21}$$

Consider the minimization of (21) by the coordinate descent method on the finest level 1:

Given $\phi^{(m)} = \left(\phi_{i,j}^{(0)}\right)$ *with* $m = 0$,

$$\textit{Solve } \phi_{i,j}^{(m)} = \underset{\phi_{i,j}, c_1, c_2 \in \mathbb{R}}{\arg\min} F_{RC}^{loc}(\phi_{i,j}, c_1, c_2) \textit{ for } i, j = 1, 2, ..., n, \tag{22}$$

Set $\phi_{i,j}^{(m+1)} = \left(\phi_{i,j}^{(m)}\right)$ *and repeat the above steps with* $m = m + 1$ *until stopped. Here*

$$
\begin{aligned}
F_{RC}^{loc}\left(\phi_{i,j},c_1,c_2\right) = \quad F_{RC} - F_0 \;=\; & \bar{\mu}\Big[g_{i,j}\sqrt{\left(H_{i,j}-H_{i+1,j}^{(m)}\right)^2+\left(H_{i,j}-H_{i,j+1}^{(m)}\right)^2+\beta} \\
& + g_{i-1,j}\sqrt{\left(H_{i,j}-H_{i-1,j}^{(m)}\right)^2+\left(H_{i-1,j}^{(m)}-H_{i-1,j+1}^{(m)}\right)^2+\beta} \\
& + g_{i,j-1}\sqrt{\left(H_{i,j}-H_{i,j-1}^{(m)}\right)^2+\left(H_{i,j-1}^{(m)}-H_{i+1,j-1}^{(m)}\right)^2+\beta}\Big] \\
& + \lambda_1(z_{i,j}-c_1)^2H_{i,j} + \lambda_2(z_{i,j}-c_2)^2(1-H_{i,j}) \\
& + \nu(H_{i,j}-\mathrm{A}_1)^2 + \nu((1-H_{i,j})-\mathrm{A}_2)^2.
\end{aligned}
$$

The term F_0 refers to a collection of all terms that are not dependent on $\phi_{i,j}$. For $\phi_{i,j}$ at the boundary, Neumann's condition is used. In order to introduce the multilevel algorithm we first rewrite (22) in an equivalent form:

$$
\hat{c} = \arg\min_{c\in\mathbb{R}} F_{RC}^{loc}\left(\phi_{i,j}^{(m)}+c,c_1,c_2\right),\ \phi_{i,j}^{(m+1)} = \phi_{i,j}^{(m)} + \hat{c} \text{ for } i,j = 1,2,...,n. \tag{23}
$$

Similar to BC1, we arrive at the following local functional for $\hat{c}$ on a general level:

$$
\begin{aligned}
F_{RC1}\left(c_{i,j}\right) =& \bar{\mu}\sum_{\ell=\ell_1}^{\ell_2} g_{k_1-1,\ell}\sqrt{\left(c_{i,j}-h_{k_1-1,\ell}\right)^2+\upsilon_{k_1-1,\ell}^2+\beta} + \nu\sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2}\left(-\mathrm{A}_1+H(\tilde{\phi}_{k,\ell}+c_{i,j})\right)^2 \\
&+\bar{\mu}\sum_{\ell=\ell_1}^{\ell_2-1} g_{k_2,\ell}\sqrt{\left(c_{i,j}-h_{k_2,\ell}\right)^2+\upsilon_{k_2,\ell}^2+\beta} + \bar{\mu}\sum_{k=k_1}^{k_2} g_{k,\ell_1-1}\sqrt{\left(c_{i,j}-\upsilon_{k,\ell_1-1}\right)^2+h_{k,\ell_1-1}^2+\beta} \\
&+\bar{\mu}\sqrt{2}g_{k_2,\ell_2}\sqrt{\left(c_{i,j}-\bar{\upsilon}_{k_2,\ell_2}\right)^2+\bar{h}_{k_2,\ell_2}^2+\frac{\beta}{2}} + \lambda_1\sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2} H(\tilde{\phi}_{k,\ell}+c_{i,j})\left(z_{k,\ell}-c_1\right)^2 \\
&+\lambda_2\sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2}\left(1-H(\tilde{\phi}_{k,\ell}+c_{i,j})\right)\left(z_{k,\ell}-c_2\right)^2 + \bar{\mu}\sum_{k=k_1}^{k_2-1} g_{k,\ell_2}\sqrt{\left(c_{i,j}-\upsilon_{k,\ell_2}\right)^2+h_{k,\ell_2}^2+\beta} \\
&+\nu\sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2}\left(-\mathrm{A}_2+\left(1-H(\tilde{\phi}_{k,\ell}+c_{i,j})\right)\right)^2.
\end{aligned} \tag{24}
$$

A single constant update on the current $\tilde{\phi}$ on the coarsest level is given by solving

$$
\begin{aligned}
\min_c \hat{F}_{RC1}(\tilde{\phi}+c) = & \lambda_1\sum_{i=1}^{n}\sum_{j=1}^{n} H(\tilde{\phi}_{i,j}+c)\left(z_{i,j}-c_1\right)^2 + \nu\sum_{i=1}^{n}\sum_{j=1}^{n}\Big(-\mathrm{A}_1+H(\tilde{\phi}_{i,j}+c)\Big)^2 \\
& + \lambda_2\sum_{i=1}^{n}\sum_{j=1}^{n}\Big(1-H(\tilde{\phi}_{i,j}+c)\Big)\left(z_{i,j}-c_2\right)^2 \\
& + \nu\sum_{i=1}^{n}\sum_{j=1}^{n}\Big(-\mathrm{A}_2+\Big(1-H(\tilde{\phi}_{i,j}+c)\Big)\Big)^2.
\end{aligned} \tag{25}
$$

Our implementation of the proposed multilevel method is then summarized in Algorithm 2 which will be referred to as **RC1**.

Before we conclude this section, we give a brief convergence analysis of **BC1** and **RC1**. Let $N = n^2$ be the total number of pixels (unknowns). First, we compute the number of floating point operations (flops) for **BC1** for level k as follows:

Algorithm 2 RC1 – Multilevel algorithm for the RC model

Given z, an initial guess $\tilde{\phi}$ and the stop tolerance tol with $L+1$ levels,

1) *Iteration starts with $\phi_{old} = \tilde{\phi}$ ($\tilde{\phi}$ contains the initial guess before the first iteration and the updated solution at all later iterations)*

2) *Smooth for t iterations the approximation on the finest level 1 i.e. solve* $\min_{\phi_{i,j}} F^{loc}_{RC}(\phi_{i,j}, c_1, c_2)$ or (23) *for $i, j = 1, 2, ...n$*

3) *Iterate for t times on each coarse level $k = 2, 3, ...L, L+1$:*
   - ► *If $k \leq L$, compute the minimizer c of* (24) or solve $\min_{c_{i,j}} F_{RC1}(c_{i,j})$;
   - ► *If $k = L+1$, solve* (25) or $\min_c \hat{F}_{RC1}(\tilde{\phi} + c)$ *on the coarsest level .*

   Add the correction $\tilde{\phi} = \tilde{\phi} + Q_k c$ where Q_k is the interpolation operator distributing $c_{i,j}$ to the corresponding $b_k \times b_k$ block on level k as illustrated in (17).

4) *Return to Step 1 unless $\frac{\|\tilde{\phi} - \phi_{old}\|_2}{\|\tilde{\phi}\|_2} < tol$ or until the prescribed maximum of cycles is reached. Otherwise exit with $\phi = \tilde{\phi}$.*

Quantities	Flop counts for **BC1**
$h,\ v$	$4b_k\tau_k$
$\lambda_1\, term$	$2N$
$\lambda_2\, term$	$2N$
s smoothing steps	$38b_k\tau_k s$

Then, the flop counts for all level is $\xi_{BC1} = \sum_{k=1}^{L+1} (4N + 4b_k\tau_k + 38b_k\tau_k s)$ where $k = 1$ (finest) and $k = L+1$ (coarsest). Next, we compute the upper bound for **BC1** as follows:

$$\xi_{BC1} = 4(L+1)N + \sum_{k=1}^{L+1}\left(\frac{4N}{b_k} + \frac{38Ns}{b_k}\right) = 4(L+1)N + (4+38s)N\sum_{k=0}^{L}\left(\frac{1}{2^k}\right)$$
$$< 4N\log n + 12N + 76Ns \approx O\left(N\log N\right)$$

Similarly, the flops for **RC1** is given as

Quantities	Flop counts for **RC1**
$h,\ v$	$4b_k\tau_k$
λ_1 term	$2N$
λ_2 term	$2N$
ν term	$4N$
s smoothing steps	$31b_k\tau_k s$

Hence, the total flop counts for **RC1** is $\xi_{RC1} = \sum_{k=1}^{L+1} (8N + 4b_k\tau_k + 31b_k\tau_k s)$. This gives the upper bound for **RC1** as

$$\xi_{RC1} = 8(L+1)N + \sum_{k=1}^{L+1}\left(\frac{4N}{b_k} + \frac{31Ns}{b_k}\right) = 8(L+1)N + (4+31s)N\sum_{k=0}^{L}\left(\frac{1}{2^k}\right)$$
$$< 8N\log n + 16N + 62Ns \approx O\left(N\log N\right)$$

One can observe that both **BC1** and **RC1** are of the optimal complexity $O(N \log N)$ expected of a multilevel method and $\xi_{RC1} > \xi_{BC1}$.

It may be remarked that both algorithms **BC1** and **RC1** are easily paralellizable and hence there is much potential to explore parallel efficiency. However below we consider how to improve the sequence efficiency in a simple and yet effective manner.

4 The new localized models

The complexity of the above presented algorithms is $O(N \log N)$ per cycle through all levels, for an image sized $n \times n$ with $N = n^2$. As this is optimal, for most problems, there is no need to consider further reduction for man problems e.g. for image denoising. However, segmentation is a special problem because evolution of level set functions ϕ is always local in selective segmentation. Below we turn this locality into reformulations and explore further reduction of the $O(N \log N)$ complexity, consequently achieving the super-optimal efficiency.

Motivated by developing faster solution algorithms than Algorithms 1-2 and by methods using narrow band region-based active contours, localized models amenable to fast solution are proposed in this section respectively for the BC model [6] and the RC model [35] . It is followed by presenting the corresponding multilevel algorithms to solve them. As expected, the complexity of the new models will be directly linked to the length of segmented objects at each iteration; at the discrete level, this length is usually $O(\sqrt{N})$. Our use of narrow band regions is fundamentally different from active contours in that we apply the idea to a model, not just to a numerical procedure.

The key notation used below is the following, as shown in Fig.3. Given a level set function

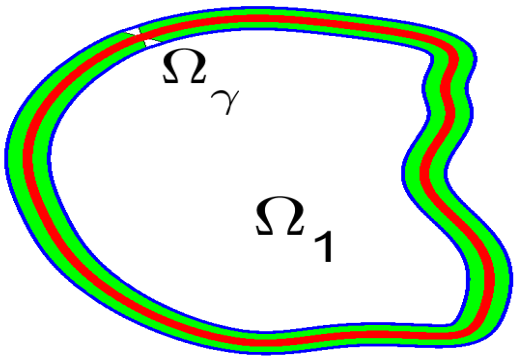


Figure 3: New modelling setup: replacement of domain Ω_1 by a smaller domain Ω_γ.

ϕ (intended to represent Ω_1), a local function b defined by

$$b\left(\phi\left(x,y\right),\gamma\right) = H\left(\phi\left(x,y\right)+\gamma\right)\left(1-H(\phi(x,y)-\gamma)\right) \tag{26}$$

characterizes the narrow band region domain $\Omega_\gamma = \Omega_1\left(\gamma\right) \cup \Gamma \cup \Omega_2\left(\gamma\right)$ surrounding the boundary Γ, with $\Omega_1\left(\gamma\right)$ and $\Omega_2\left(\gamma\right)$ denoting the γ band inside and outside region from Γ respectively. Similar notation is also used by [28, 43]. Note $b = 1$ inside Ω_γ and 0 outside, and similarly $b\left(\phi\left(x,y\right),\gamma\right) H\left(\phi\right) = 1$ inside $\Omega_1\left(\gamma\right)$ and 0 outside i.e. $b\left(\phi\left(x,y\right),\gamma\right)\left(1-H\left(\phi\right)\right) = 1$ inside $\Omega_2\left(\gamma\right)$ and 0 outside. Further after discretization, we define the notation for the set falling into the γ-band where $b = 1$:

$$B(\phi) = \left\{(i,j) \;\middle|\; -\gamma \le \phi_{i,j} \le \gamma \text{ i.e. } \phi\left(x,y\right)+\gamma > 0 \text{ and } \phi(x,y)-\gamma < 0\right\}. \tag{27}$$

We propose a localized version of the BC model [6] by the following

$$\min_{\Gamma,c_1,c_2}\left\{F_{BL}\left(\Gamma,c_1,c_2\right)=\mu\int_{\Gamma}dgds+F_{BL}^{\gamma}\left(\Gamma,c_1,c_2\right)\right\} \tag{28}$$

where the refinement is seen in

$$F_{BL}^{\gamma}\left(\Gamma,c_1,c_2\right)=\lambda_1\int_{\Omega_1(\gamma)}\left(z-c_1\right)^2dxdy+\lambda_2\int_{\Omega_2(\gamma)}\left(z-c_2\right)^2dxdy.$$

In level set formation,

$$\begin{aligned}\min_{\phi,c_1,c_2}F_{BL}\left(\phi,c_1,c_2\right)=\mu\int_{\Omega}dg\sqrt{\left|\nabla H\left(\phi\right)\right|^2+\beta}\,dxdy+\lambda_1\int_{\Omega}\left(z-c_1\right)^2b\left(\phi,\gamma\right)H\left(\phi\right)\,dxdy\\ +\lambda_2\int_{\Omega}\left(z-c_2\right)^2b\left(\phi,\gamma\right)\left(1-H\left(\phi\right)\right)\,dxdy.\end{aligned} \tag{29}$$

Next, we propose a localized RC model of the form

$$\begin{aligned}\min_{\phi,c_1,c_2}F_{RL}\left(\phi,c_1,c_2\right)=\mu\int_{\Omega}g\left(\left|\nabla z\left(x,y\right)\right|\right)\sqrt{\left|\nabla H\left(\phi\right)\right|^2+\beta}\,dxdy+\\ \lambda_1\int_{\Omega}\left(z-c_1\right)^2b\left(\phi,\gamma\right)H\left(\phi\right)\,dxdy+\lambda_2\int_{\Omega}\left(z-c_2\right)^2b\left(\phi,\gamma\right)\left(1-H\left(\phi\right)\right)dxdy\\ +\nu\Big(\int_{\Omega}b\left(\phi,\gamma\right)H\left(\phi\right)\,dxdy-\mathrm{A}_1\Big)^2+\nu\Big(\int_{\Omega}b\left(\phi,\gamma\right)\left(1-H\left(\phi\right)\right)\,dxdy-\mathrm{A}_2\Big)^2.\end{aligned} \tag{30}$$

5 Multilevel algorithms for localized segmentation models

We now show how to adapt the above Algorithms 1-2 to the new formulations (29) and (30).

Multilevel algorithm for the localized BC model. Discretize functional (29) as

$$\begin{aligned}F_{BL}\left(\phi,c_1,c_2\right)=\ &\bar{\mu}\sum_{i=1}^{n-1}\sum_{j=1}^{n-1}G_{i,j}\sqrt{\left(H_{i,j}-H_{i,j+1}\right)^2+\left(H_{i,j}-H_{i+1,j}\right)^2+\beta}\\ &+\lambda_1\sum_{i=1}^{n}\sum_{j=1}^{n}\left(z_{i,j}-c_1\right)^2H_{i,j}b_{i,j}\\ &+\lambda_2\sum_{i=1}^{n}\sum_{j=1}^{n}\left(z_{i,j}-c_2\right)^2\left(1-H_{i,j}\right)b_{i,j}\end{aligned} \tag{31}$$

where $G=dg,\ G_{i,j}=G(x_i,y_j),\ (i,j)\in B(\phi)$. Minimization of (31) by the coordinate descent method on the finest level 1 leads to the following local minimization for only $(i,j)\in B(\phi^{(m)})$:

$$\begin{aligned}F_{BL}^{loc}(\phi_{i,j},c_1,c_2)=&\bar{\mu}\Big[G_{i,j}\sqrt{\left(H_{i,j}-H_{i+1,j}^{(m)}\right)^2+\left(H_{i,j}-H_{i,j+1}^{(m)}\right)^2+\beta}\\ &+G_{i-1,j}\sqrt{\left(H_{i,j}-H_{i-1,j}^{(m)}\right)^2+\left(H_{i-1,j}^{(m)}-H_{i-1,j+1}^{(m)}\right)^2+\beta}\\ &+G_{i,j-1}\sqrt{\left(H_{i,j}-H_{i,j-1}^{(m)}\right)^2+\left(H_{i,j-1}^{(m)}-H_{i+1,j-1}^{(m)}\right)^2+\beta}\Big]\\ &+\lambda_1(z_{i,j}-c_1)^2H_{i,j}b_{i,j}+\lambda_2(z_{i,j}-c_2)^2\left(1-H_{i,j}\right)b_{i,j}\end{aligned} \tag{32}$$

where $b_{i,j}=1$ if $(i,j)\in B(\phi^{(m)})$ else $b_{i,j}=0$.

Algorithm 3 BC2 – Multilevel algorithm for the new local BC model

- Input γ and the other quantities as in Algorithm 1
- Apply Algorithm 1 to new functionals from replacing

$$\min_{\phi_{i,j}} F^{loc}_{BC}(\phi_{i,j}, c_1, c_2) \quad \text{on the finest level by} \quad \min_{\phi_{i,j}} F^{loc}_{BL}(\phi_{i,j}, c_1, c_2)$$

$$\min_{c_{i,j}} F_{BC1}(c_{i,j}) \quad \text{on coarse levels by} \quad \min_{c_{i,j}} F_{BC2}(c_{i,j})$$

All other steps are identical.

Further, the multilevel method for the localized BC model (29) at a general level for updating block $[k_1, k_2] \times [\ell_1, \ell_2]$ amounts to minimizing the following local functional

$$\begin{aligned}
F_{BC2}\,(c_{i,j}) =& \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2} G_{k_1-1,\ell} \sqrt{(c_{i,j} - h_{k_1-1,\ell})^2 + \upsilon^2_{k_1-1,\ell} + \beta} + \bar{\mu} \sum_{k=k_1}^{k_2-1} G_{k,\ell_2} \sqrt{(c_{i,j} - \upsilon_{k,\ell_2})^2 + h^2_{k,\ell_2} + \beta} \\
&+ \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2-1} G_{k_2,\ell} \sqrt{(c_{i,j} - h_{k_2,\ell})^2 + \upsilon^2_{k_2,\ell} + \beta} + \bar{\mu} \sum_{k=k_1}^{k_2} G_{k,\ell_1-1} \sqrt{(c_{i,j} - \upsilon_{k,\ell_1-1})^2 + h^2_{k,\ell_1-1} + \beta} \\
&+ \bar{\mu}\sqrt{2} G_{k_2,\ell_2} \sqrt{(c_{i,j} - \bar{\upsilon}_{k_2,\ell_2})^2 + \bar{h}^2_{k_2,\ell_2} + \beta/2} \\
&+ \lambda_1 \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \Big|_{(k,\ell)\in B(\tilde{\phi})} H(\tilde{\phi}_{k,\ell} + c_{i,j})(z_{k,\ell} - c_1)^2 b(\tilde{\phi}_{k,\ell} + c_{i,j}, \gamma) \\
&+ \lambda_2 \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \Big|_{(k,\ell)\in B(\tilde{\phi})} \big(1 - H(\tilde{\phi}_{k,\ell} + c_{i,j})\big)(z_{k,\ell} - c_2)^2 b(\tilde{\phi}_{k,\ell} + c_{i,j}, \gamma)
\end{aligned} \tag{33}$$

similar to Algorithm 1, where $(i,j) \in B(\tilde{\phi})$. The difference is that $\tilde{\phi} := \tilde{\phi} + c_{i,j}$ only needs updating if the set $[k_1, k_2] \times [\ell_1, \ell_2] \cap B(\tilde{\phi})$ is non-empty. We will use the term **BC2** to refer the multilevel Algorithm 3.

Multilevel algorithm for the localized RC model. Functional (30) is discretized as

$$\begin{aligned}
F_{RL}\,(\phi, c_1, c_2) = \;& \bar{\mu} \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} g_{i,j} \sqrt{(H_{i,j} - H_{i,j+1})^2 + (H_{i,j} - H_{i+1,j})^2 + \beta} \\
&+ \lambda_1 \sum_{i=1}^{n} \sum_{j=1}^{n} (z_{i,j} - c_1)^2 H_{i,j} b_{i,j} + \lambda_2 \sum_{i=1}^{n} \sum_{j=1}^{n} (z_{i,j} - c_2)^2 (1 - H_{i,j})\, b_{i,j} \\
&+ \nu\Big(-\mathrm{A}_1 + \sum_{i=1}^{n} \sum_{j=1}^{n} H_{i,j} b_{i,j}\Big)^2 + \nu\Big(-\mathrm{A}_2 + \sum_{i=1}^{n} \sum_{j=1}^{n} (1 - H_{i,j})\, b_{i,j}\Big)^2.
\end{aligned} \tag{34}$$

Further at a general level, whenever a block $[k_1, k_2] \times [\ell_1, \ell_2]$ overlaps with $B(\tilde{\phi})$ (i.e. the set

Algorithm 4 RC2 – Multilevel algorithm for the new and local RC model

- Input γ and the other quantities as in Algorithm 2
- Apply Algorithm 2 to new functionals from replacing

$$\min_{\phi_{i,j}} F_{RC}^{loc}(\phi_{i,j}, c_1, c_2) \quad \text{on the finest level by} \quad \min_{\phi_{i,j}} F_{RL}^{loc}(\phi_{i,j}, c_1, c_2)$$

$$\min_{c_{i,j}} F_{RC1}(c_{i,j}) \quad \text{on coarse levels by} \quad \min_{c_{i,j}} F_{RC2}(c_{i,j})$$

All other steps are identical.

$[k_1, k_2] \times [\ell_1, \ell_2] \cap B(\tilde{\phi})$ is non-empty), the multilevel method minimizes

$$\begin{aligned}
F_{RC2}(c_{i,j}) =& \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2} g_{k_1-1,\ell} \sqrt{(c_{i,j} - h_{k_1-1,\ell})^2 + \upsilon_{k_1-1,\ell}^2 + \beta} + \bar{\mu} \sum_{k=k_1}^{k_2-1} g_{k,\ell_2} \sqrt{(c_{i,j} - \upsilon_{k,\ell_2})^2 + h_{k,\ell_2}^2 + \beta} \\
&+ \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2-1} g_{k_2,\ell} \sqrt{(c_{i,j} - h_{k_2,\ell})^2 + \upsilon_{k_2,\ell}^2 + \beta} + \bar{\mu}\sqrt{2} g_{k_2,\ell_2} \sqrt{(c_{i,j} - \bar{\upsilon}_{k_2,\ell_2})^2 + \bar{h}_{k_2,\ell_2}^2 + \frac{\beta}{2}} \\
&+ \bar{\mu} \sum_{k=k_1}^{k_2} g_{k,\ell_1-1} \sqrt{(c_{i,j} - \upsilon_{k,\ell_1-1})^2 + h_{k,\ell_1-1}^2 + \beta} \\
&+ \lambda_1 \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \Big|_{(k,\ell)\in B(\tilde{\phi})} b\left(\tilde{\phi}_{k,\ell} + c_{i,j}, \gamma\right) H(\tilde{\phi}_{k,\ell} + c_{i,j}) \, (z_{k,\ell} - c_1)^2 \\
&+ \lambda_2 \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \Big|_{(k,\ell)\in B(\tilde{\phi})} \left(1 - H(\tilde{\phi}_{k,\ell} + c_{i,j})\right) b\left(\tilde{\phi}_{k,\ell} + c_{i,j}, \gamma\right) (z_{k,\ell} - c_2)^2 \\
&+ \nu \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \Big|_{(k,\ell)\in B(\tilde{\phi})} \left(-\mathrm{A}_1 + b\left(\tilde{\phi}_{k,\ell} + c_{i,j}, \gamma\right) H\left(\tilde{\phi}_{k,\ell} + c_{i,j}\right)\right)^2 \\
&+ \nu \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \Big|_{(k,\ell)\in B(\tilde{\phi})} \left(-\mathrm{A}_2 + \left(1 - H\left(\tilde{\phi}_{k,\ell} + c_{i,j}\right)\right) b\left(\tilde{\phi}_{k,\ell} + c_{i,j}, \gamma\right)\right)^2 \qquad (35)
\end{aligned}$$

and then updates $\tilde{\phi}$ by $\tilde{\phi} + c_{i,j}$. We will refer this Algorithm 4 as **RC2**.

For a single object extraction, Algorithms 3 - 4 have a complexity of $O(\gamma n \log N) = O(\sqrt{N} \log N)$ where $\log N$ refers to the number of levels. However they are only applicable to our selective models; for global models such as the CV model, the band idea promotes local minimizers and is hence not useful.

6 Numerical experiments

In order to demonstrate the strengths and limitations of the proposed multilevel method for both the original and the localized segmentation models, we performed several experiments. The main algorithms to be compared are

Name	Algorithm		Description
BC0	Old	:	The AOS algorithm [6] for the original BC model [6];
BCP	Old	:	The Pyramid scheme for BC0;
BC1	New	:	The multilevel Algorithm 1 for the BC model;
BC2	New	:	The multilevel Algorithm 3 for the localized BC model;
RC0	Old	:	The AOS algorithm [35] for the original RC model [35];
RCP	Old	:	The Pyramid scheme for RC0;
RC1	New	:	The multilevel Algorithm 2 for the RC model;
RC2	New	:	The multilevel Algorithm 4 for the localized RC model.

Our aims of the tests are
i) to verify numerically the efficiency as n increases – is an algorithm faster or slower than or of the same magnitude as $O(N \log N)$ where $N = n^2$;
ii) to compare the quality, we use the so-called the Jaccard similarity coefficient (JSC) and Dice similarity coefficient (DSC):

$$JSC = \frac{|S_n \cap S_*|}{|S_n \cup S_*|}, \qquad DSC = \frac{2\,|S_n \cap S_*|}{|S_n| + |S_*|}$$

where S_n is the set of the segmented domain Ω_1 and S_* is the true set of Ω_1. The similarity functions return values in the range $[0, 1]$. The value 1 indicates perfect segmentation quality while the value 0 indicates poor quality.

The test images used in this paper are listed in Figure 4. They are four images used which include 3 real medical images and 1 synthetic image (Problems 1 which have known solutions to aid computing JSC and DSC). The markers set also are shown in Figure 4. The initial contour is defined by the markers set. We remark that for an image of size $n \times n$ the number of unknowns is $N = n^2$ which means that for $n = 256,\ 512,\ 1024,\ 2048$, the respective number of unknowns is $N = 65536,\ 262144,\ 1048576,\ 4194304$; we are solving large scale problems. Our algorithms are implemented in MATLAB R2017a on a personal computer with Intel Core i7 processor, CPU 3.60GHz, 16 GB RAM CPU. All the programs are stopped when $tol = 10^{-4}$ or when the maximum number of iterations, $maxit = 1500$ is reached.

6.1 Comparison of BC2 with BC0, BC1 and BCP

In the following experiments, we take the parameters $\lambda_1 = \lambda_2 = 1$, $\alpha = 0.01$, $\beta = 10^{-4}$ and $\kappa = 4$. Through the experiments it was observed that the parameters ε and η can be in a range between $\varepsilon = 1/n$ to 1 and $\eta = 10^{-3}$ and 10^2.

First, we compare BC1 and BC2 using test Problems 1-3. All the images are of size 256×256. We take $\bar{\mu} = 0.05n^2$ (Problem 1) and $\bar{\mu} = 0.1n^2$ (Problems $2 - 3$). For BC2, γ is between 30 to 100.

Figure 5(a) and 5(b) show the successful selective segmentation results by BC1 and BC2 respectively for capturing one object in Problems 1-3. We see that that the results from BC1 are quite similar to BC2. The times needed by BC1 and BC2 to complete the selective segmentation task are tabulated below

Problem	BC1	BC2
1	12.1	8.4
2	11.7	6.8
3	11.9	9.3

where we can observe that BC2 is about 2 times faster than BC1.

Second, against BC2, we test BC0 that is based on additive operator splitting (AOS) [6], the pyramid scheme BCP based on BC0 and BC1. For this purpose, we segment Problem 1 with different resolutions using $\mu = \bar{\mu} = 0.05n^2$. The segmentation results for image size 1024×1024

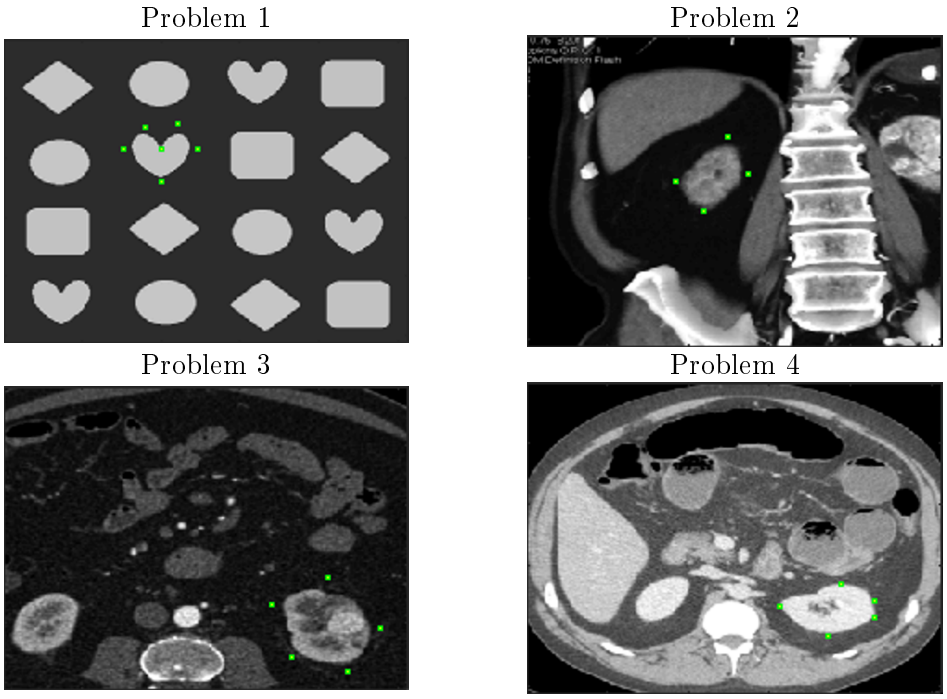


Figure 4: Test images with the markers set.

are shown in Figure 6 and the results for all sizes in terms of quality and computation time needed to complete the segmentation tasks are presented in Table 1. Columns 5 (ratios of the CPU times) show that BC0, BCP and BC1 are of complexity $O(N \log N)$ while BC2 of the 'super'-optimal efficiency $O(\sqrt{N} \log N)$.

Clearly BC0 (the AOS method for the BC model with added balloon force) provides an effective acceleration for images of moderate size $n \leq 256$. Significant improvement can be seen by BCP which shows that the pyramid method together with AOS is better than BC0. However, we can see that our BC1 and BC2 are faster than BC0 and BCP, while BC2 is faster than the other 3 algorithms. The computation time differences between BC2 with other 3 algorithms become significant as the image size increases to $n \geq 512$. The BC0 result with " ** " indicates that a very long time is taken to complete the segmentation task. One can see for example BC0 needs almost 100 times more time compared to BC2 to complete segmentation in case of 1024×1024. We also see that all algorithms provide high segmentation quality, from JSC and DSC values.

6.2 Comparison of RC2 with RC0, RC1, RCP

In the following experiments, we fixed the parameters $\lambda_1 = \lambda_2 = 1$, $\alpha = 0.01$ and $\beta = 10^{-4}$. Through the experiments it was observed that the parameters ν, ε and η can be in a range between $\nu = 0.001$ to 0.01 $\varepsilon = 1/n$ to 1 and $\eta = 10^{-3}$ to 10^{-2}.

We first compare RC1 and RC2 using Problems 1-3. All the images are of size 256×256. We take $\bar{\mu} = 0.05n^2$ (Problem 1) and $\mu = \bar{\mu} = 0.1n^2$ (Problems 2-3). For RC2, γ is between 30 to 100. Figure 7(a) and 7(b) show the successful selective segmentation results of RC1 and RC2 respectively for capturing one object for Problems 1-3.

We then compare RC2 with RC0, RCP and RC1 using Problem 1. Here $\mu = \bar{\mu} = 0.05n^2$ for all algorithms. The segmentation results for image size 1024×1024 shown in Figure 8 and the quality measures and the computation time presented in Table 2 show that RC2 can be 100 times faster than RC0, 17 times faster than RCP and 4 times faster than RC1 for the case

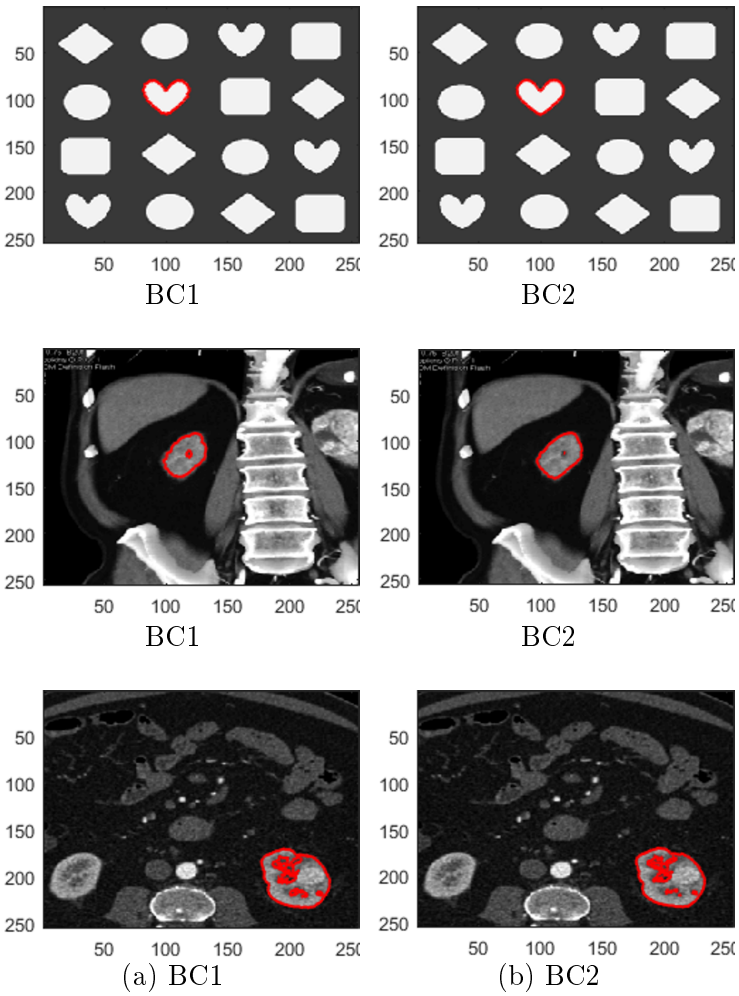


Figure 5: Segmentation of Problems 1-3: Column (a) BC1 and (b) BC2.

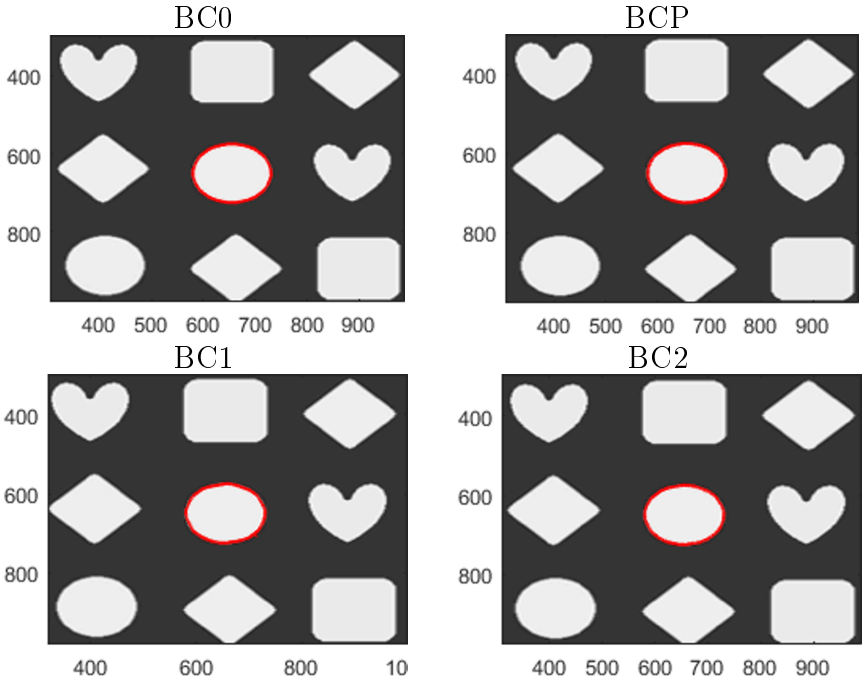


Figure 6: Segmentation of Problem 1 of size 1024×1024 for BC0, BCP, BC1 and BC2.

Table 1: Comparison of computation time (in seconds) and segmentation quality of BC0, BCP and BC1 with our BC2 for Problem 1. The ratio close to 4.4 for time indicates $O(N \log N)$ speed while a ratio of 2.2 indicates $O(\sqrt{N} \log N)$ "super-optimal" speed, where the number of unknowns $N = n^2$. Here and later, '**' means taking too long to run.

Algorithm	Size $n \times n$	Number of iteration (outer)	Time t_n	$\frac{t_n}{t_{n-1}}$	JSC	DSC
BC0	256×256	1293	227.8		1.0	1.0
	512×512	1276	898.5	3.9	1.0	1.0
	1024×1024	1234	4095.5	4.6	1.0	1.0
	2048×2048	**	**	**	**	
BCP	256×256	4	61.0		1.0	1.0
	512×512	2	180.0	3.0	1.0	1.0
	1024×1024	2	812.3	4.5	1.0	1.0
	2048×2048	2	3994.0	4.9	1.0	1.0
BC1	256×256	2	11.6		1.0	1.0
	512×512	2	43.7	3.8	1.0	1.0
	1024×1024	2	173.2	4.0	1.0	1.0
	2048×2048	2	736.9	4.3	1.0	1.0
BC2	256×256	2	10.5		1.0	1.0
	512×512	2	21.6	2.1	1.0	1.0
	1024×1024	2	42.5	2.0	1.0	1.0
	2048×2048	2	80.5	1.9	1.0	1.0

of 1024×1024: In particular, ratios of the CPU times verify that RC0, RCP and RC1 are of complexity $O(N \log N)$ while RC2 of the 'super'-optimal efficiency $O(\sqrt{N} \log N)$. Furthermore, the RC0 result with " ** " indicates that too much time is taken to complete the segmentation task. The high values of JSC and DSC show that RC0, RCP, RC1 and RC2 provide high segmentation quality.

For the benefit of readers, in Figure 9, we demonstrate a convergent plot based on Tables 1-2 of our proposed multilevel based models (BC2 and RC2) in segmenting Problem 1 of size 2048×2048. One can see that the models are fast, converging to *tol* in 2 iterations that is before the prescribed *maxit*.

Furthermore, we have extended the number of iterations for BC2 and RC2 up to 6 iterations and plot the residual history in the same Figure 9. We can observe that BC2 and RC2 keep converging.

6.3 Sensitivity tests on the algorithmic parameters

Sensitivity is a major issue that has to be addressed and tested below. We shall pay particular attention to the regularizing parameter β that was known to be sensitive to convergence of a geometric multigrid method [4]; it turns out that our Algorithms 1-4 are more advantageous as they are not very sensitive to β.

Tests on parameter t. The inner iteration t indicates the number of iterations needed to solve the minimization problem in each level. We test several numbers of t required by BC2 and RC2 to segment heart shape in Problem 1 and record the outer iteration needed to achieve *tol*, the CPU time and the quality of segmentation. The results are tabulated in Table 3.

We can see that BC2 and RC2 work in efficiently and effectively using only 1 inner iteration i.e $t = 1$. As we increase t, the quality of segmentation for BC2 and RC2 reduce and need more CPU time and outer iteration as well.

Table 2: Comparison of computation time (in seconds) and segmentation quality of RC0, RCP and RC1 with RC2 for Problem 1. Again, the ratio close to 4.4 for time indicates $O(N \log N)$ speed while a ratio of 2.2 indicates $O(\sqrt{N} \log N)$ "super-optimal" speed, where the number of unknowns $N = n^2$.

Algorithm	Size $n \times n$	Number of iterations (outer)	Time t_n	$\frac{t_n}{t_{n-1}}$	JSC	DSC
RC0	256×256	1500	260.5		1.0	1.0
	512×512	1385	975.0	3.7	1.0	1.0
	1024×1024	1404	4735.0	4.9	1.0	1.0
	2048×2048	**	**	**	**	
RCP	256×256	4	62.5		1.0	1.0
	512×512	2	187.2	3.0	1.0	1.0
	1024×1024	2	822.3	4.4	1.0	1.0
	2048×2048	2	3996.3	4.9	1.0	1.0
RC1	256×256	2	13.0		1.0	1.0
	512×512	2	48.4	3.7	1.0	1.0
	1024×1024	2	189.9	3.9	1.0	1.0
	2048×2048	2	819.0	4.3	1.0	1.0
RC2	256×256	2	11.5		1.0	1.0
	512×512	2	24.0	2.1	1.0	1.0
	1024×1024	2	46.9	2.0	1.0	1.0
	2048×2048	2	87.6	1.9	1.0	1.0

Table 3: Dependence of BC2 and RC2 on t for heart shape in Problem 1 (Figure 4).

Algorithm	t:inner iteration	Number of iterations (outer)	CPU	JSC	DSC
BC2	1	2	8.1	1.0	1.0
	2	6	22.4	1.0	1.0
	3	7	26.7	0.9	1.0
RC2	1	2	8.8	1.0	1.0
	2	9	35.3	0.9	1.0
	3	7	28.7	0.9	1.0

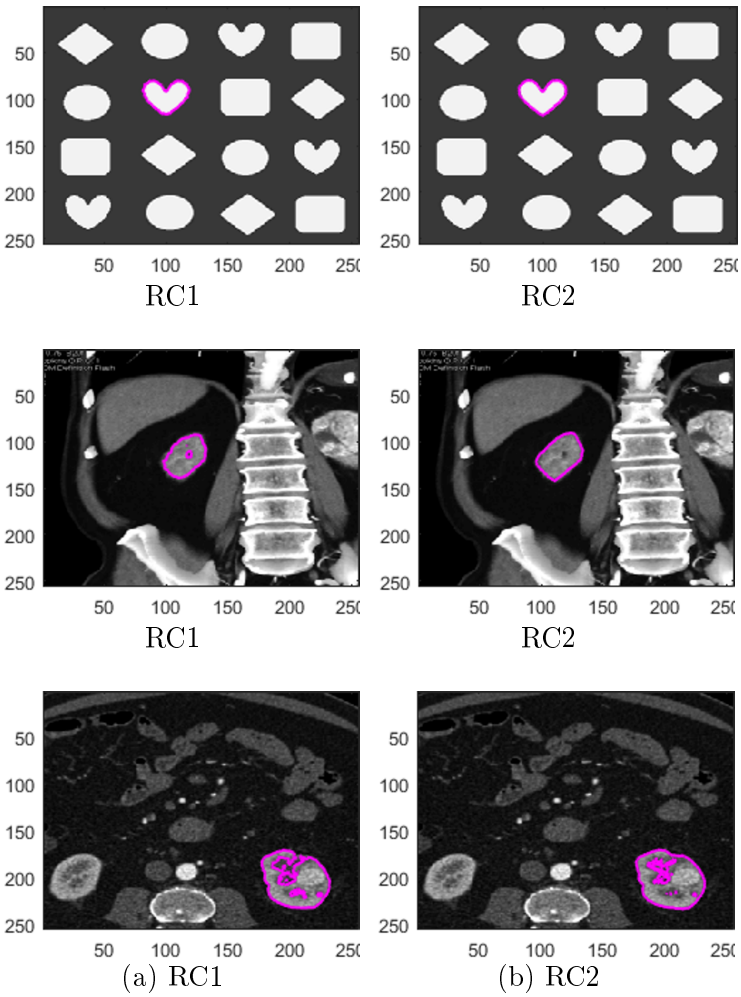


Figure 7: Segmentation of Problems 1-3. (a) and (b) show the segmentation using RC1 and RC2 respectively.

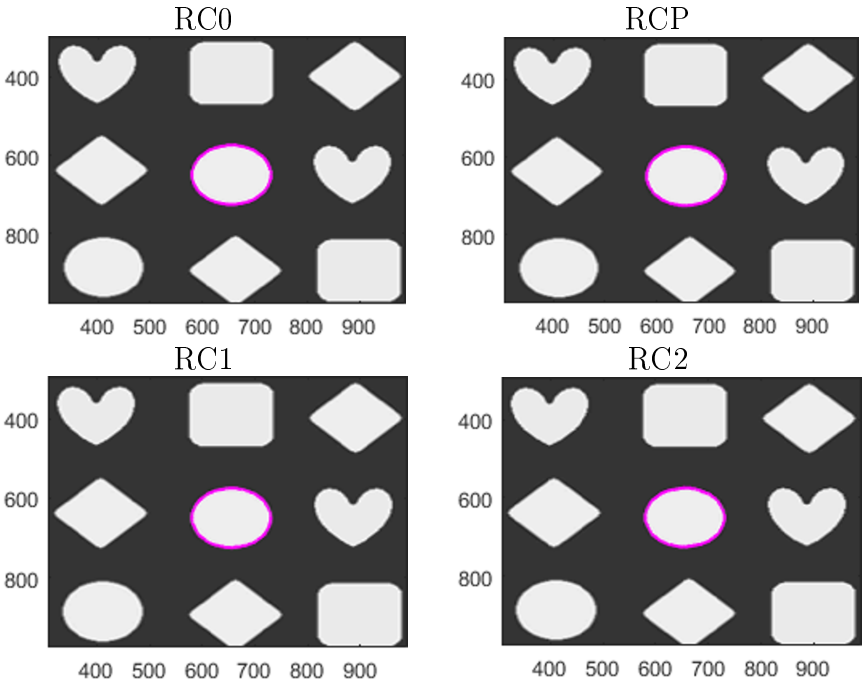


Figure 8: Segmentation of Problem 1 of size 1024×1024 for RC0, RCP, RC1 and RC2. For the same segmentation result, RC2 can be 100 times faster than RC0, 17 times faster than RCP and 4 times faster than RC1; see Table 2.

Table 4: Dependence of our new BC2 and RC2 on β for heart shape in Problem 1 (Figure 4).

β	BC2			RC2		
	$F_{BL}(\phi, c_1, c_2)$	JSC	DSC	$F_{RL}(\phi, c_1, c_2)$	JSC	DSC
1	2.461759e+09	0.6	0.7	5.177135e+10	0.6	0.7
10^{-1}	2.258762e+09	0.9	1.0	5.168056e+10	0.9	1.0
10^{-2}	2.197002e+09	1.0	1.0	5.164663e+10	1.0	1.0
10^{-4}	2.178939e+09	1.0	1.0	5.163375e+10	1.0	1.0
10^{-6}	2.177950e+09	1.0	1.0	5.163266e+10	1.0	1.0
10^{-8}	2.176280e+09	1.0	1.0	5.163252e+10	1.0	1.0
10^{-10}	2.175254e+09	1.0	1.0	5.163243e+10	1.0	1.0

Tests on parameter γ. The band width parameter γ is an important parameter to be tested. Its size determines how local the resulting segmentation will be. Below we will demonstrate the effect of applying different values of γ to BC2 and RC2.

We aim to segment an organ in Problem 4 by for BC2 and RC2 with varying γ and the results are presented in Figure 10. Columns 2 and 3 of Figure 10 show the results using 3 γ values (more spread out) for BC2 and RC2. Clearly, unless the value is too small (that results in an incorrect segmentation), in general, both BC2 and RC2 are not much sensitive to γ choice.

Tests on parameter β. Finally, we examine the sensitivity of BC2 and RC2 on this important parameter β. Seven different values of β are tested: $\beta = 1$, 10^{-1}, 10^{-2}, 10^{-4}, 10^{-6}, 10^{-8} and 10^{-10} in segmenting the heart shape in Problem 1. For a quantitative analysis, we evaluate the energy value $F_{BL}(\phi, c_1, c_2)$ in equation (31), $F_{RL}(\phi, c_1, c_2)$ in equation (34) and the indexes JSC and DSC. The values of $F_{BL}(\phi, c_1, c_2)$, $F_{RL}(\phi, c_1, c_2)$, JSC and DSC are tabulated in Table 4. One can see that as β decreases, the functional $F_{BL}(\phi, c_1, c_2)$ and $F_{RL}(\phi, c_1, c_2)$ gets closer to each other. The segmentation quality measured by JSC and DSC increases as β decreases. This finding shows that BC2 and RC2 are only sensitive to (unrealistic) large β but less to a very small β. In separate experiments, we found that BC2 and RC2 algorithms are not much sensitive to η, α, ε, and ν (involve in RC2 only), although there exist choices what give the optimal quality of segmentation.

Acknowledgements

The first author would like to thank to Universiti Teknologi MARA and Ministry of Higher Education of Malaysia for funding a scholarship to support this research. The second author is grateful to the support from the UK EPSRC grants EP/K036939/1 and EP/N014499/1.

7 Conclusions

In this work, we presented an optimization based multilevel method to solve two variational and selective segmentation models (BC and RC), though the idea is applicable to other global and variational models.

In Part 1, we presented 2 algorithms (BC1, RC1) for solving the respective models with each algorithm having the expected optimal complexity of $O(N \log N)$ for segmentation of an image of size $n \times n$ or $N = n^2$ unknowns (pixels). These algorithms can be adapted to solve other segmentation models. In Part 2, we reformulated the models so that they became localized versions that operate within a banded region of an active level set contour, and consequently obtained 2 further algorithms (BC2, RC2) with each algorithm having the 'super'-optimal complexity of approximately $O(\sqrt{N} \log N)$ depending on the objects to be segmented. These algorithms

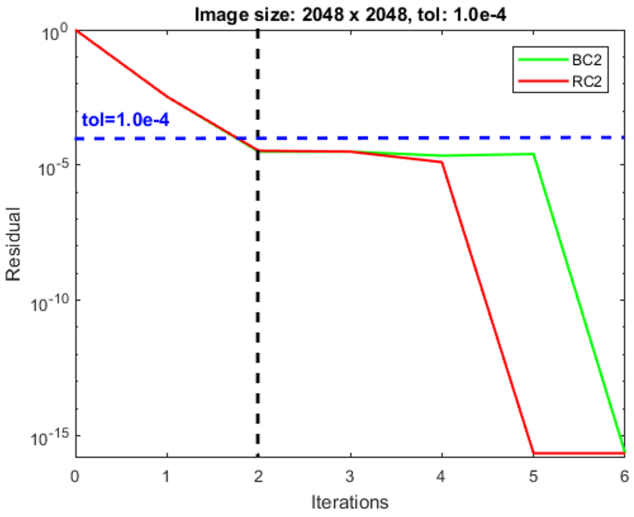
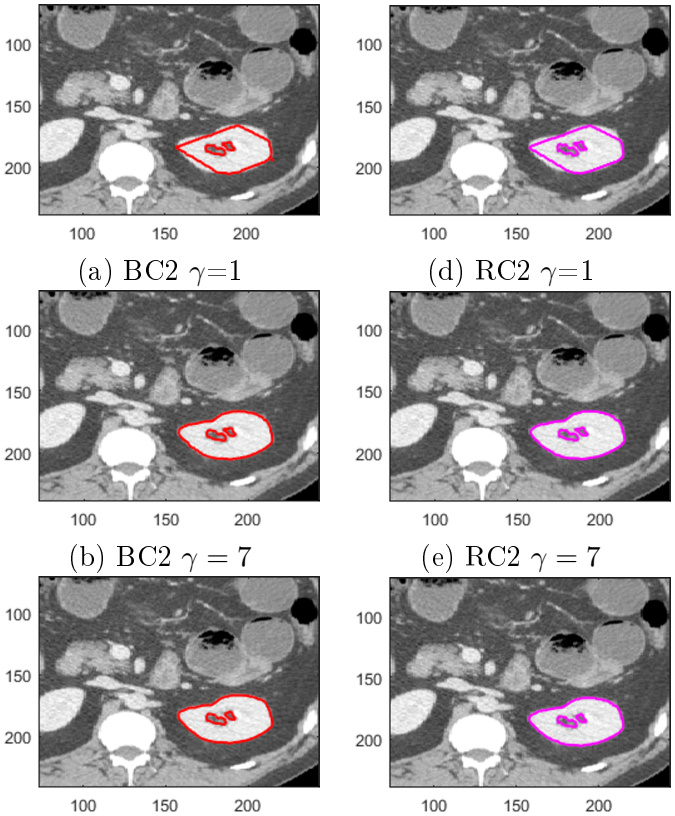


Figure 9: The number of iterations needed by BC2 and RC2 to achieve a set *tol* (residual) in segmenting an image of size 2048×2048. With $tol = 10^{-4}$, BC2 and RC2 need 2 iterations. The extension up to 6 iterations shows that residuals of BC2 and RC2 keep reducing.



(a) BC2 γ=1 (d) RC2 γ=1

(b) BC2 $\gamma = 7$ (e) RC2 $\gamma = 7$

(c) BC2 $\gamma = 100$ (f) RC2 $\gamma = 100$

Figure 10: Dependence of algorithms BC2, RC2 on parameter γ for Problem 4.

are only applicable to our selective segmentation models. Numerical experiments have verified the complexity claims, and comparisons with related algorithms (BC0, BCP, RC0, RCP for the standard models) show that the new algorithms are many times faster than BC0, BCP, RC0, RCP, in achieving comparable quality of segmentation.

Future works will address convexified selective variational models such as [39] especially in high dimensions and other image processing tasks such as image registration and joint registration and segmentation models. There is much scope to explore the presented algorithms on parallel platforms, especially for 3D problems.

References

[1] R. Acar and C. R. Vogel, Analysis of bounded variation penalty methods for ill-posed problems, Inverse Problems, 10 (6), pp:1217-1229, 1994.

[2] R. Adams and L. Bischof. Seeded region growing. IEEE Trans. Pattern Anal. Mach. Intell., 16(6), pp:641–647, June 1994.

[3] G. Aubert and P. Kornprobst. Mathematical Problems In Image Processing: Partial differential Equations and the Calculus of Variations. Springer, 2001.

[4] N. Badshah and K. Chen. Multigrid method for the Chan-Vese model in variational segmentation. Communications in Computational Physics(CiCP),4(2),pp:294–316, 2008.

[5] N. Badshah and K. Chen. On Two multigrid algorithms for modelling variational multiphase image segmentation. IEEE Transaction on Image Processing, Vol. 18(5), pp:1097–1106,2009.

[6] N. Badshah and K. Chen. Image selective segmentation under geometrical constraints using an active contour approach. Commun. Comput. Phys, 7(4), pp:759–778, 2010.

[7] I. N. Bankman, T. Nizialek, I. Simon, O. B. Gatewood, I. N. Weinberg, and W. R. Brody. Segmentation algorithms for detecting microcalcifications in mammograms. IEEE Trans. Inform. Techn. Biomed. 1(2), pp:161–173, 1993.

[8] S. Bernsen. Segmentation tools in mathematical morphology. SPIE 1350, Image Algebra and Morphological Image Processing, pp:70–84, 1990.

[9] J. L. Carter. Dual Method for Total Variation-Based Image Restoration, CAM Report 02-13, University of California, Los Angeles, CA, 2002.

[10] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. International Journal of Computer Vision, 22(1), pp:61–79, 1997.

[11] T. F. Chan and K. Chen. An optimization-based multilevel algorithm for total variation image denoising, Multiscale Model. Simul., 5, pp: 615–645, 2006.

[12] R. H. Chan and K. Chen. Multilevel algorithm for a Poisson noise removal model with total variation regularisation, Int. J. Comput. Math., 84, pp: 1183–1198, 2007.

[13] R. H. Chan and K. Chen. A Multilevel algorithm for simultaneously denoising and deblurring images, SIAM J. Sci. Comput., 32, pp:1043–1063, 2010.

[14] T.F. Chan, S. Esedoglu, and M. Nikilova. Algorithm for finding global minimizers of image segmentation and denoising models. SIAM J. Appl. Math., 66, pp:1932–1648, 2006.

[15] T. F. Chan and L. A. Vese. Active contours without edges. IEEE Transactions on Image Processing, 10(2), pp:266–277, 2001.

[16] D. Comaniciu, and P. Meer. Mean shift: a robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24, pp:603–619, 2002.

[17] M. Comer, C. Bouman, and J. Simmons. Statistical methods for image segmentation and tomography reconstruction. Microscopy and Microanalysis, 16(S2), pp:1852-1853, 2010.

[18] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. IEEE Trans. Pattern Anal. Machine Intell., 6(6), pp:721–741, 1984.

[19] P. Getreuer. Rudin-Osher-Fatemi total variation denoising using split bregman. Image Processing On Line., 2, pp:74–95, 2012.

[20] C. Gout, C. Le Guyader, and L.A. Vese. Segmentation under geometrical conditions with geodesic active contour and interpolation using level set methods. Numerical Algorithms, 39, pp:155-173, 2005.

[21] C.L. Guyader and C. Gout. Geodesic active contour under geometrical conditions theory and 3D applications. Numerical Algorithms, 48, pp:105-133, 2008.

[22] M. Hadhoud, M. Amin, and W. Dabbour. Detection of breast cancer tumor algorithm using mathematical morphology and wavelet analysis. In Proceedings of GVIP 05 Conference, CICC. Cairo, Egypt, 2005.

[23] S. Hanov. Wavelets and edge detection (preprint). `http://stevehanov.ca/cs698_wavelet_project.pdf`, April 2006.

[24] M. Kass, M. Witkin, and D. Terzopoulos. Snakes: active contour models. Inter. J. Comput. Vision, 1, pp:321–331, 1987.

[25] A. Kenigsberg, R. Kimmel, and I. Yavneh. A multigrid approach for fast geodesic active contours, Comput. Sci. Depart., The Technion—Israel Int. Technol., Haifa, Tech. Rep. CIS-2004-06, 2004.

[26] J. Malik, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. International Journal of Computer Vision, 43, pp:7–27, 2001.

[27] S. Mallat. A Wavelet Tour Of Signal Processing. Academic Press, USA, 1998.

[28] J. Mille, R. Bone, P. Makris, and H. Cardot. Narrow band region-based active contours and surfaces for 2D / 3D segmentation. Computer Vis. Image Underst., 113, pp:946–965, 2009.

[29] L. Moisan. How to discretize the Total Variation of an image?. Proc. Appl. Math. Mech, 2007. John Wiley and Sons, Ltd.

[30] S. Morigi, L. Reichel and F. Sgallari, Noise-reducing cascadic multilevel methods for linear discrete ill-posed problems. Numerical Algorithms, 53, pp:1-22, 2010.

[31] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. Commu. Pure Applied Mathematics, 42, pp:577–685, 1989.

[32] J. M. Ortega and W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, London, 1970.

[33] G. Papandreou and P. Maragos. A fast multigrid implicit algorithm for the evolution of geodesic active contours. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Washington DC, 2, pp: 689–694, 2004.

[34] G. Papandreou and P. Maragos. Multigrid geometric active contour models. IEEE Trans. Image Process., 16(1), pp: 229–240, Jan. 2007.

[35] L. Rada and K. Chen. Improved selective segmentation model using one level set. Journal of Algorithm & Computational Technology, 7(4), pp: 509-541, 2013.

[36] D. Sen and S. K. Pal. Histogram thresholding using fuzzy and rough measures of association error. IEEE Transactions on Image Processing, 18(4), pp:879–888, 2009.

[37] J. A. Sethian. Level Set Methods and Fast Marching Methods. Cambridge University Press, 1999.

[38] A. I. Shihab. Fuzzy Clustering Algorithms and Their Application to Medical Image Analysis. Ph.D. Diss. Dept. of Computing, Univ. of London, 2000.

[39] J. Spencer and K. Chen. A convex and selective variational model for image segmentation. Communication in Mathematical Sciences, 13(6), pp: 1453–1452, 2015.

[40] J. S. Weszka. A survey of threshold selection techniques. Computer Graphics and Image Proc., 7, pp: 259–265, 1978.

[41] J. Yuan, E. Bae and X. C. Tai. A study on continuous max-flow and min-cut approaches. In CVPR, USA, San Francisco, 2010.

[42] J. Yuan, E. Bae, X. C. Tai and Y. Boycov. A study on continuous max-flow and min-cut approaches. Technical report CAM, UCLA, 2010.

[43] J. Zhang, K.Chen, B.Yu, and D.Gould. A local information based variational model for selective image segmentation. J. Inverse Problems and Imaging, 8, pp: 293–320, 2014.