Electronic Transactions on Numerical Analysis.
Volume 46, pp. 474–504, 2017.

ISSN 1068–9613.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

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 at selectively segmenting one object only. We are concerned with the 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 for such problems. Here we propose an optimization-based multilevel algorithm for efficiently solving a class of selective segmentation models. It also applies to the solution of global segmentation models. In a 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 $N = n^2$. Moreover, modified localized models are proposed to exploit the local nature of the segmentation contours, and consequently, our second variant—after modifications—practically achieves super-optimal efficiency $O(\sqrt{N} \log N)$. Numerical results show that a good segmentation quality is obtained, and as expected, excellent efficiency is observed in reducing the computational time.

Key words. active contours, image segmentation, level-set function, multilevel, optimization methods, energy minimization

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

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 homogeneous 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 [25, 34], region growing [2], or edge detection and active contours [3, 15]. Of all these techniques, variational techniques [15, 30] are proven to be very efficient for extracting homogeneous areas compared with other models such as statistical methods [16, 17, 18] or wavelet techniques [26].

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 models that are able to drive the contours towards image edges by the influence of an edge detector function. The snake algorithm proposed by Kass et al. [23] was the first edge-based variational model for image segmentation. Further improvement on the algorithm with geodesic active contours and the level-set formulation led to effective models [10, 35]. 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 [7, 22], the watershed algorithm [8, 22], thresholding [22, 38], and fuzzy clustering [36]. The most celebrated and efficient variational model for images with and without noise is the Mumford-Shah [30] model, which reconstructs the segmented image as a piecewise smooth intensity function. Since the model cannot be implemented directly and easily, it is often approximated. The Chan-Vese (CV) [15] model is a simplified and reduced form of that in [30] without approximation. The

*Received November 17, 2016. Accepted October 26, 2017. Published online on December 19, 2017. Recommended by Fiorella Sgallari. Corresponding author: K. Chen.

†Center for Mathematical Imaging Techniques and Department of Mathematical Sciences, University of Liverpool, United Kingdom. ({abdulkj,k.chen}@liverpool.ac.uk).

ETNA
Kent State University and
Johann Radon Institute (RICAM)

simplification is achieved by replacing the piecewise smooth function by a piecewise constant function and, in the case of two phases, by a separation of the image into the foreground and the background.

The segmentation models described above are suited 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 the 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 methods with $O(N \log N)$ efficiency (at the level of a multigrid method) for an image sized $n \times n$ with $N = n^2$ [39, 40].

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, 33, 37]. Two effective models are Badshah-Chen [6] and Rada-Chen [33], which use a mixture of edge-based and region-based ideas in addition to imposing constraints. Recently, a convex selective variational image segmentation model referred to as Convex Distance Selective Segmentation was successfully proposed by Spencer and Chen [37]. The convex model allows a global minimiser to be found independently of the initialisation [14, 37]. 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 a hierarchy of discretizations. However, a multilevel method is based on a discretize-optimize scheme (algebraic) where the minimization of a variational problem is solved directly without using a partial differential equation (PDE). In contrast, a multigrid method is based on an optimize-discretize scheme (geometric) which solves a PDE numerically. The two methods are interconnected since both can have geometric interpretations and use similar inter-level information transfers.

The latter multigrid methods have been used for solving a few variational image segmentation models in the level-set formulation. For geodesic active contours models, linear multigrid methods have been developed [24, 31, 32]. 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 segmentations. 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 is due to jumps or edges that separate segmented domains). Therefore, the above multigrid methods behave like the cascadic multigrids [29] 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 deblurring [11, 12, 13] but not yet to segmentation problems. However, the method is found to get stuck in local minima due to a non-differentiability of the energy functional. To overcome such 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 the image size increases, the method can be slow because 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 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

ETNA
Kent State University and
Johann Radon Institute (RICAM)

in the level-set formulation. The key finding is that the resulting multilevel algorithm converges while being not very sensitive to parameter choices, unlike geometric multigrid methods [5], which are known to have convergence problems.

The rest of the paper is organized in the following way. In Section 2, we briefly review a global segmentation model, namely the Chan-Vese model [15], and two selective segmentation models, the Badshah-Chen model [6] and the Rada-Chen model [33]. In Section 3, we present an optimization-based multilevel algorithm for the selective segmentation models. In Section 4, we propose localized segmentation models and furthermore present multilevel methods for solving them in Section 5. In Section 6, we give some experimental results to test the algorithms. We compare the new methods to the previously fast methods from the literature, namely the AOS method for the Badshah-Chen [6] and Rada-Chen [33] models (since multigrid methods are not yet developed for them). However, a multiscale AOS method (for Badshah-Chen [6] and Rada-Chen [33]) based on the pyramid idea is implemented and included in the comparison. Section 7 contains concluding remarks.

2. Review of three existing models. In this section we 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 discuss two selective segmentation models by Badshah-Chen [6] and Rada-Chen [33] before addressing the issue of solving these models fast.

2.1. The Chan-Vese model. The Chan and Vese (CV) model [15] can be considered a special case of the piecewise constant Mumford-Shah functional [30] where the functional 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 of 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 by the value c_2. Then, with $\Omega = \Omega_1 \cup \Omega_2$, the Chan-Vese model minimizes the following functional:

$$\begin{aligned} &\min_{\Gamma, c_1, c_2} F_{CV}\left(\Gamma, c_1, c_2\right), \\ &F_{CV}\left(\Gamma, c_1, c_2\right) = \mu \operatorname{length}\left(\Gamma\right) + \lambda_1 \int_{\Omega_1} (z - c_1)^2 dxdy + \lambda_2 \int_{\Omega_2} (z - c_2)^2 dxdy. \end{aligned} \tag{2.1}$$

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 have to be specified. In order to minimize equation (2.1), the level-set method is applied [15], where the unknown curve Γ is represented by the zero level set of the Lipschitz function ϕ such that

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

To simplify the notation, denote the regularized versions of the Heaviside function and the Dirac delta function, respectively, by

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

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Thus, equation (2.1) becomes

$$\begin{aligned}&\min_{\phi,c_1,c_2} F_{CV}\left(\phi,c_1,c_2\right),\\ &F_{CV}\left(\phi,c_1,c_2\right)=\mu\int_\Omega\left|\nabla H(\phi)\right|dxdy+\lambda_1\int_\Omega\left(z-c_1\right)^2H(\phi)\,dxdy\\ &\qquad\qquad+\lambda_2\int_\Omega\left(z-c_2\right)^2\left(1-H(\phi)\right)\,dxdy.\end{aligned}\tag{2.2}$$

Keeping the level-set function ϕ fixed and minimizing (2.2) with respect to c_1 and c_2, we have

$$c_1(\phi)=\frac{\int_\Omega z(x,y)H(\phi)\,dxdy}{\int_\Omega H(\phi)\,dxdy},\qquad c_2(\phi)=\frac{\int_\Omega z(x,y)\left(1-H(\phi)\right)\,dxdy}{\int_\Omega\left(1-H(\phi)\right)\,dxdy}.\tag{2.3}$$

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

$$\begin{aligned}&\mu\delta\left(\phi\right)\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right)-\lambda_1\delta\left(\phi\right)\left(z-c_1\right)^2+\lambda_2\delta\left(\phi\right)\left(z-c_2\right)^2=0\quad\text{in }\Omega,\\ &\qquad\qquad\frac{\delta\left(\phi\right)}{|\nabla\phi|}\,\frac{\partial u}{\partial\vec{n}}=0\quad\text{on }\partial\Omega.\end{aligned}\tag{2.4}$$

Notice that the nonlinear coefficient in equation (2.4) 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 (2.2) and (2.4), so that the new Euler-Lagrange equation becomes

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

which corresponds to minimizing the following differentiable energy function instead of (2.2):

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

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 the intensity-fitting terms of Chan-Vese [15]. For an 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 [33, 41], the selective segmentation idea tries to detect the boundary of a single object in Ω closest to $\mathcal{A}$ among all objects with homogeneous intensity; here $n_1\ge 3$. The geometrical points in $\mathcal{A}$ define an initial polygonal contour and guide its evolution towards Γ [41].

ETNA
Kent State University and
Johann Radon Institute (RICAM)

The BC minimization [6] is given by

$$\min_{\Gamma,c_1,c_2} F_{BC}\left(\Gamma,c_1,c_2\right),$$

$$F_{BC}\left(\Gamma,c_1,c_2\right)=\mu\int_{\Gamma} d(x,y)g\left(|\nabla z(x,y)|\right)\,dxdy + \lambda_1\int_{\text{inside}(\Gamma)}\left(z-c_1\right)^2 dxdy+\lambda_2\int_{\text{outside}(\Gamma)}\left(z-c_2\right)^2 dxdy. \tag{2.5}$$

In this model, the function $g\left(|\nabla z|\right)=\frac{1}{1+\eta|\nabla z(x,y)|^2}$ 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 $g\left(|\nabla z|\right)$ is constructed such that it takes small values (close to 0) near object edges and large values (close to 1) in flat regions. The function $d(x,y)$ is a marker distance function which is close to 0 when approaching the points from the marker set and is given as

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

where κ is a positive constant. Alternative distance functions $d(x,y)$ are also possible [33, 41]. Using a level-set formulation, the functional (2.5) becomes

$$\min_{\phi,c_1,c_2} F_{BC}\left(\phi,c_1,c_2\right),$$

$$F_{BC}\left(\phi,c_1,c_2\right)=\mu\int_{\Omega} d(x,y)g\left(|\nabla z(x,y)|\right)\left|\nabla H\left(\phi\right)\right|dxdy + \lambda_1\int_{\Omega}\left(z-c_1\right)^2 H\left(\phi\right)dxdy+\lambda_2\int_{\Omega}\left(z-c_2\right)^2\left(1-H\left(\phi\right)\right)dxdy. \tag{2.6}$$

Keeping the level-set function ϕ fixed and minimizing (2.5) with respect to c_1 and c_2, we have

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

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

$$\begin{aligned}\mu\delta\left(\phi\right)\nabla\cdot dg\left(\frac{\nabla\phi}{\sqrt{|\nabla\phi|^2+\beta}}\right)-\lambda_1\delta\left(\phi\right)\left(z-c_1\right)^2 \\ +\lambda_2\delta\left(\phi\right)\left(z-c_2\right)^2=0 \qquad \text{in } \Omega,\\ dg\frac{\delta\left(\phi\right)}{|\nabla\phi|}\,\frac{\partial u}{\partial\vec{n}}=0 \qquad \text{on } \Omega.\end{aligned} \tag{2.7}$$

The small positive parameter β is introduced to avoid singularities in (2.7), which corresponds to minimizing the following differentiable form of the BC model in replacement of (2.6)

$$\min_{\phi,c_1,c_2} F_{BC}\left(\phi,c_1,c_2\right)$$

$$F_{BC}\left(\phi,c_1,c_2\right)=\mu\int_{\Omega} G(x,y)\sqrt{|\nabla H\left(\phi\right)|^2+\beta}\,dxdy + \lambda_1\int_{\Omega}\left(z-c_1\right)^2 H\left(\phi\right)\,dxdy+\lambda_2\int_{\Omega}\left(z-c_2\right)^2\left(1-H\left(\phi\right)\right)\,dxdy, \tag{2.8}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

where $G = d(x,y)g(x,y)$. To facilitate faster convergence, a balloon force term $\alpha G|\nabla\phi|$ is added to (2.7) as in [6].

2.3. The Rada-Chen model. The Rada-Chen (RC) model [33] 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 the area inside and outside the polygon, respectively. The areas A_1 and A_2 are computed to approximate the area of the object of interest. The RC model also incorporates a similar edge detection function as in the BC model into the regularization term. The energy minimization problem is given by

$$\begin{aligned}
&\min_{\Gamma,c_1,c_2} F_{RC}\left(\Gamma,c_1,c_2\right),\\
&F_{RC}\left(\Gamma,c_1,c_2\right) = \mu\int_{\Gamma} g\left(|\nabla z(x,y)|\right)\,dxdy + \lambda_1\int_{\Omega_1}(z-c_1)^2dxdy\\
&\qquad + \lambda_2\int_{\Omega_2}(z-c_2)^2dxdy + \nu\left(\int_{\Omega_1}dxdy - \mathrm{A}_1\right)^2 + \nu\left(\int_{\Omega_2}dxdy - \mathrm{A}_2\right)^2.
\end{aligned} \tag{2.9}$$

Rewriting (2.9) into a level-set formulation as in (2.3), we arrive at the following Euler-Lagrange equation for ϕ:

$$\begin{aligned}
&\mu\delta\left(\phi\right)\nabla\cdot g\left(\frac{\nabla\phi}{\sqrt{|\nabla\phi|^2+\beta}}\right) - \lambda_1\delta\left(\phi\right)(z-c_1)^2 + \lambda_2\delta\left(\phi\right)(z-c_2)^2\\
&-\nu\delta\left(\phi\right)\left(\left(\int_{\Omega}H\left(\phi\right)\,dxdy - \mathrm{A}_1\right) - \left(\int_{\Omega}\left(1-H\left(\phi\right)\right)dxdy - \mathrm{A}_2\right)\right) = 0 \ \text{ in } \Omega,\\
&dg\frac{\delta\left(\phi\right)}{|\nabla\phi|}\frac{\partial u}{\partial\vec{n}} = 0, \ \text{ on } \Omega.
\end{aligned} \tag{2.10}$$

As with the BC model, in the actual implementation of the RC model, the small positive parameter β is introduced to avoid singularities in (2.10), which corresponds to minimizing the following differentiable form of the RC model

$$\begin{aligned}
&\min_{\phi,c_1,c_2} F_{RC}\left(\phi,c_1,c_2\right)\\
&F_{RC}\left(\phi,c_1,c_2\right) = \mu\int_{\Omega} g\left(|\nabla z(x,y)|\right)\sqrt{|\nabla H\left(\phi\right)|^2+\beta}\,dxdy\\
&\qquad + \lambda_1\int_{\Omega}(z-c_1)^2H\left(\phi\right)\,dxdy + \lambda_2\int_{\Omega}(z-c_2)^2\left(1-H\left(\phi\right)\right)dxdy\\
&\qquad + \nu\left(\int_{\Omega}H\left(\phi\right)\,dxdy - \mathrm{A}_1\right)^2 + \nu\left(\int_{\Omega}\left(1-H\left(\phi\right)\right)\,dxdy - \mathrm{A}_2\right)^2.
\end{aligned} \tag{2.11}$$

To stimulate faster convergence, a balloon force term is added to (2.10), which is defined as $\alpha g(x,y)\,|\nabla\phi(x,y)|$.

We use the abbreviations **BC0** and **RCO** to refer to the AOS algorithm previously used to solve the BC model and the RC model in [6] and [33], respectively.

Of course, it is known that such AOS methods are not designed for processing large images. To assist AOS, a pyramid method can be used. In the process of the curve evolution, the pyramid scheme employs a decomposition of the image into different scale, and then coarse segmentation is performed on the coarse-scale image using the AOS method instead of

ETNA
Kent State University and
Johann Radon Institute (RICAM)

directly working with the original-size image. Then, the segmentation result is interpolated and adopted to an initial contour for the fine-scale image, thus gradually optimizing the contour and reaching the final segmentation result. We refer to the pyramid method for BC and RC models as **BCP** and **RCP**, respectively.

The above variational models, the BC model (2.8) and the RC model (2.11), respectively, will be conveniently solved by our new proposed multilevel scheme. The models **BC0**, **RCO**, **BCP**, and **RCP** will serve as comparisons 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 [33]. 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: $k = 1$ (the finest), $2, \ldots, L, L + 1$ (the coarsest), such that level k has $\tau_k \times \tau_k$ "superpixels" with each "superpixel" having $b_k \times b_k$ pixels, where $\tau_k = n/2^{k-1}$ and $b_k = 2^{k-1}$. Figures 3.2(a-e) illustrate 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 shape and the rest of rectangular shape.

3.1. Multilevel algorithm for the BC model. Our goal is to solve (2.8), i.e., the BC model [6], using a multilevel method in a discretize-optimize scheme. Before we proceed further, one may wonder how to discretize the total variation (TV) term

$$TV\left(u\right) = \int_{\Omega} \left|\nabla u\right| dxdy.$$

In fact, TV is most often discretized by

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

There are other ways to define the discrete TV by finite difference, but the above form is the simplest one according to [28]. In addition, the reason why this form is considered a reasonable discretization of TV relies in the notion of consistency that is well known in numerical analysis: if we consider a regular function $U : \mathbb{R}^2 \to \mathbb{R}$ and its discretization

$$\left(i, j\right) \mapsto U_h\left(i, j\right) = U\left(ih, jh\right),$$

for $h > 0$, we have

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

as $h \to 0$, $ih \to x$, 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) do

ETNA
Kent State University and
Johann Radon Institute (RICAM)

not depend on $u_{i,j}$:

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

To avoid this problem, one-sided differences can be used. More discussions on discretizing TV can be found in [19] and [28] and the references therein.

Using the above information, the discretized version of (2.8) is given by:

$$\min_{\phi,c_1,c_2} F_{BC}(\phi, c_1, c_2) \equiv \min_{\phi,c_1,c_2} F_{BC}^a(\phi_{1,1}, \phi_{2,1}, \ldots, \phi_{i-1,j}, \phi_{i,j}, \phi_{i+1,j}, \ldots, \phi_{n,n}, c_1, c_2)$$

with

$$\begin{aligned} &F_{BC}^a(\phi_{1,1}, \phi_{2,1}, \ldots, \phi_{i-1,j}, \phi_{i,j}, \phi_{i+1,j}, \ldots, \phi_{n,n}, c_1, c_2) \\ &\quad = \bar{\mu} \sum_{i,j=1}^{n-1} G_{i,j} \sqrt{(H_{i,j} - H_{i,j+1})^2 + (H_{i,j} - H_{i+1,j})^2 + \beta} \\ &\qquad + \lambda_1 \sum_{i,j=1}^{n} (z_{i,j} - c_1)^2 H_{i,j} + \lambda_2 \sum_{i,j=1}^{n} (z_{i,j} - c_2)^2 (1 - H_{i,j}), \end{aligned} \tag{3.1}$$

where ϕ denotes a row vector,

$$\bar{\mu} = \frac{\mu}{h}, \qquad h = \frac{1}{n-1}, \qquad G_{i,j} = G(x_i, y_j),$$

$$c_1 = \frac{\sum_{i,j=1}^{n} z_{i,j} H_{i,j}}{\sum_{i,j=1}^{n} H_{i,j}}, \qquad c_2 = \frac{\sum_{i,j=1}^{n} z_{i,j} (1 - H_{i,j})}{\sum_{i,j=1}^{n} (1 - H_{i,j})}, \qquad \text{and}$$

$$H_{i,j} = \frac{1}{2} + \frac{1}{\pi} \arctan \frac{\phi_{i,j}}{\varepsilon}.$$

As a prelude to multilevel methods, consider the minimization of (3.1) by the coordinate descent method on the finest level 1:

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

$$\text{(3.2)} \qquad \textit{Solve } \phi_{i,j}^{(m+1)} = \arg\min_{\phi_{i,j} \in \mathbb{R}} F_{BC}^{loc}(\phi_{i,j}, c_1, c_2) \qquad \text{for } i, j = 1, 2, \ldots, n.$$

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

ETNA
Kent State University and
Johann Radon Institute (RICAM)

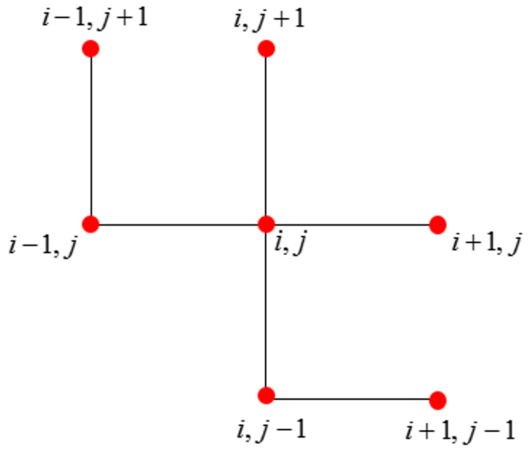


FIG. 3.1. *The interaction of $\phi_{i,j}$ at a central pixel (i,j) with neighboring pixels on the finest level 1. Clearly only 3 terms (pixels) are involved with $\phi_{i,j}$ (through regularisation).*

Here equation (3.2) is obtained by expanding and simplifying the main model in (3.1), i.e.,

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

with Neumann 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 is done as before. Clearly, it seems that this is a coordinate descent method. As such, the method will exhibit a decay of the functional $F_{BC}^{a}(\phi^{(m)}) \leq F_{BC}^{a}(\phi^{(m-1)})$ from one substep to the next. It should be remarked that the formulation in (3.2) is based on the work in [9, 11].

Using (3.2), we illustrate the interaction of $\phi_{i,j}$ with its neighboring pixels on the finest level 1 in Figure 3.1. We use this basic structure to develop a multilevel method.

The one-dimensional problem in (3.2) may be solved by any suitable optimization method; here to proceed from $\phi^{(m-1)} \to \phi \to \phi^{(m)}$, we solve the first-order condition

$$
\begin{aligned}
&\frac{\bar{\mu} G_{i,j}\left(2H_{i,j} - H_{i+1,j}^{(m)} - H_{i,j+1}^{(m)}\right)}{\sqrt{\left(H_{i,j} - H_{i+1,j}^{(m)}\right)^2 + \left(H_{i,j} - H_{i,j+1}^{(m)}\right)^2 + \beta}} \\
&\qquad + \frac{\bar{\mu} G_{i-1,j}\left(H_{i,j} - H_{i-1,j}^{(m)}\right)}{\sqrt{\left(H_{i,j} - H_{i-1,j}^{(m)}\right)^2 + \left(H_{i-1,j}^{(m)} - H_{i-1,j+1}^{(m)}\right)^2 + \beta}}
\end{aligned}
$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

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

As an example, using Newton iterations, one obtains an iteration of the form

$$\phi_{i,j}^{new}=\phi_{i,j}^{old}-T^{old}/B^{old}, \tag{3.3}$$

where

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

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

To develop a multilevel version of this coordinate descent method, we may interpret (3.2) as looking for the best update $\phi_{i,j}^{(m)}$ (given 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 (3.2) in an equivalent form:

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

$$\begin{aligned}&\textit{Solve } \hat{c}=\arg\min_{c\in\mathbb{R}} F_{BC}^{loc}\left(\phi_{i,j}^{(m)}+c,c_1,c_2\right),\\ &\phi_{i,j}^{(m+1)}=\phi_{i,j}^{(m)}+\hat{c}, \qquad \text{for } i,j=1,2,\ldots,n.\end{aligned} \tag{3.4}$$

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

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 $b_k\times b_k$ pixels, e.g., the multilevel method for $k=2$ is to find the best correction constant to update this block so that the underlying merit functional, related to all four pixels (see Figure 3.2(b)), achieves a local minimum.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

For the levels $k = 1, \ldots, 5$, Figure 3.2 illustrates the multilevel partition of an image of size 16×16 pixels from (a) the finest level (level 1) to (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. Thus, from Figure 3.2(a), $b_1 = 1$ and $\tau_1 = n = 16$. On the other levels $k = 2, 3, 4, 5$, we see that the block size $b_k = 2^{k-1}$ and $\tau_k = 2^{L+1-k}$ since $n = 2^L$. In Figure 3.1, we illustrate a box $\odot$ interacting with the neighboring pixels $\bullet$ on level 3. In addition, Figure 3.2(f) illustrates the fact that a 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 the symbols $\lhd$, $\rhd$, Δ, ∇. This is important for an efficient implementation.

With the above information, we are now ready to formulate the multilevel approach for the 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 has the following structure:

(3.5)
$$\begin{array}{c|c|c|c}
\vdots & \vdots & \cdots & \vdots \\
\tilde{\phi}_{k_1-1,\ell_2+1} + c_{i-1,j+1} & \tilde{\phi}_{k_1,\ell_2+1} + c_{i,j+1} & \cdots & \tilde{\phi}_{k_2,\ell_2+1} + c_{i,j+1} \quad \tilde{\phi}_{k_2+1,\ell_2+1} + c_{i+1,j+1} \\
\hline
\tilde{\phi}_{k_1-1,\ell_2} + c_{i-1,j} & \tilde{\phi}_{k_1,\ell_2} + c_{i,j} & \cdots & \tilde{\phi}_{k_2,\ell_2} + c_{i,j} \quad \tilde{\phi}_{k_2+1,\ell_2} + c_{i+1,j} \\
\cdots & \vdots & \cdots & \vdots \quad \cdots \\
\tilde{\phi}_{k_1-1,\ell_1} + c_{i-1,j} & \tilde{\phi}_{k_1,\ell_1} + c_{i,j} & \cdots & \tilde{\phi}_{k_2,\ell_1} + c_{i,j} \quad \tilde{\phi}_{k_2+1,\ell_1} + c_{i+1,j} \\
\hline
\tilde{\phi}_{k_1-1,\ell_1-1} + c_{i-1,j-1} & \tilde{\phi}_{k_1,\ell_1-1} + c_{i,j-1} & \cdots & \tilde{\phi}_{k_2,\ell_1-1} + c_{i,j-1} \quad \tilde{\phi}_{k_2+1,\ell_1-1} + c_{i+1,j-1} \\
\vdots & \vdots & \cdots & \vdots \quad \vdots
\end{array}$$

This illustration is consistent with Figure 3.2(f), and the key point is that the interior pixels (non-boundary pixels) do not involve $c_{i,j}$ in the first nonlinear term of the formulation. 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 (3.4) is equivalent to minimizing

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

ETNA
Kent State University and
Johann Radon Institute (RICAM)

For the third term, we may note that

$$\sqrt{(c-a)^2+(c-b)^2+\beta} = \sqrt{2\left(c-\frac{a+b}{2}\right)^2+2\left(\frac{a-b}{2}\right)^2+\beta}.$$

Thus, we conclude that the local minimization problem for block (i,j) on the level k with respect to $c_{i,j}$ amounts to minimising the following equivalent functional

$$\begin{aligned}
F_{BC1}(c_{i,j}) = \;& \bar{\mu}\sum_{\ell=\ell_1}^{\ell_2} G_{k_1-1,\ell}\sqrt{(c_{i,j}-h_{k_1-1,\ell})^2+\upsilon_{k_1-1,\ell}^2+\beta} \\
&+\bar{\mu}\sum_{k=k_1}^{k_2-1} G_{k,\ell_2}\sqrt{\left(c_{i,j}-\upsilon_{k,\ell_2}\right)^2+h_{k,\ell_2}^2+\beta} \\
&+\bar{\mu}\sum_{\ell=\ell_1}^{\ell_2-1} G_{k_2,\ell}\sqrt{\left(c_{i,j}-h_{k_2,\ell}\right)^2+\upsilon_{k_2,\ell}^2+\beta} \\
&+\bar{\mu}\sum_{k=k_1}^{k_2} G_{k,\ell_1-1}\sqrt{\left(c_{i,j}-\upsilon_{k,\ell_1-1}\right)^2+h_{k,\ell_1-1}^2+\beta} \\
&+\bar{\mu}\sqrt{2}G_{k_2,\ell_2}\sqrt{\left(c_{i,j}-\bar{\upsilon}_{k_2,\ell_2}\right)^2+\bar{h}_{k_2,\ell_2}^2+\frac{\beta}{2}} \\
&+\lambda_1\sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2} H(\tilde{\phi}_{k,\ell}+c_{i,j})(z_{k,\ell}-c_1)^2 \\
&+\lambda_2\sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2}\left(1-H(\tilde{\phi}_{k,\ell}+c_{i,j})\right)(z_{k,\ell}-c_2)^2,
\end{aligned} \tag{3.6}$$

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

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

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

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

ETNA
Kent State University and
Johann Radon Institute (RICAM)

which is equivalent to

$$\begin{aligned}&\min_c \hat{F}_{BC1}(\tilde{\phi}+c),\\ &\hat{F}_{BC1}(\tilde{\phi}+c) = \lambda_1 \sum_{i,j=1}^{n} H(\tilde{\phi}_{i,j}+c)(z_{i,j}-c_1)^2\\ &\qquad\qquad + \lambda_2 \sum_{i,j=1}^{n} \big(1-H(\tilde{\phi}_{k,\ell}+c_{i,j})\big)(z_{i,j}-c_1)^2.\end{aligned} \tag{3.7}$$

In general, (3.6) 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 (3.3) 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 the level k as illustrated in (3.5). Then, we obtain a multilevel method if we cycle through all levels and all blocks on each level until the solution converges to the prescribed tolerance tol or reaches the prescribed maximum cycle (outer iteration).

So finally, our implementation of the proposed multilevel method is summarized in Algorithm 1.

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 $\min_{\phi_{i,j}} F^{loc}_{BC}(\phi_{i,j}, c_1, c_2)$ or (3.4) for $i, j = 1, 2, \ldots, n$.*
3) *Iterate for t times on each coarse level $k = 2, 3, \ldots, L, L+1$:*
   - ▶ *If $k \le L$, compute the minimiser c of (3.6) or solve $\min_{c_{i,j}} F_{BC1}(c_{i,j})$;*
   - ▶ *If $k = L+1$, solve (3.7) 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 (3.5).
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 $\phi = \tilde{\phi}$.*

Here, Steps 2–3 simply update $\tilde{\phi}$ from the finest to the coarsest level $k = 1, 2, \ldots, L, L+1$, so they might be viewed as a single step. We will use the term **BC1** to refer to the multilevel Algorithm 1. In this algorithm, we recommend to start updating our multilevel algorithm in a fast manner, i.e., 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, then convergence is slower.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

3.2. Multilevel algorithm for the RC model. The generalization of the above algorithm to other models is similar. For the RC model, the discretized version of (2.11) takes the following form

$$\begin{aligned}
&\min_{\phi,c_1,c_2} F_{RC}(\phi,c_1,c_2),\\
&F_{RC}(\phi,c_1,c_2) = \bar{\mu}\sum_{i,j=1}^{n-1} g_{i,j}\sqrt{(H_{i,j}-H_{i,j+1})^2+(H_{i,j}-H_{i+1,j})^2+\beta}\\
&\qquad + \lambda_1\sum_{i,j=1}^{n}(z_{i,j}-c_1)^2H_{i,j} + \lambda_2\sum_{i,j=1}^{n}(z_{i,j}-c_2)^2(1-H_{i,j})\\
&\qquad + \nu\Big(-\mathrm{A}_1+\sum_{i,j=1}^{n}H_{i,j}\Big)^2 + \nu\Big(-\mathrm{A}_2+\sum_{i,j=1}^{n}\Big(1-H_{i,j}\Big)\Big)^2.
\end{aligned} \tag{3.8}$$

Consider the minimization of (3.8) by the coordinate descent method on the finest level 1:

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

$$\textit{Solve}\ \ \phi_{i,j}^{(m)} = \operatorname*{arg\,min}_{\phi_{i,j},c_1,c_2\in\mathbb{R}} F_{RC}^{loc}(\phi_{i,j},c_1,c_2) \qquad \textit{for}\ i,j=1,2,\ldots,n. \tag{3.9}$$

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

Here, the functional in (3.9) is

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

The functional F_0 refers to a collection of all terms that do not depend on $\phi_{i,j}$. At the boundary, a Neumann condition for $\phi_{i,j}$ is used. In order to introduce the multilevel algorithm we first rewrite (3.9) into an equivalent form:

$$\hat{c} = \operatorname*{arg\,min}_{c\in\mathbb{R}} F_{RC}^{loc}\left(\phi_{i,j}^{(m)}+c,c_1,c_2\right), \quad \phi_{i,j}^{(m+1)} = \phi_{i,j}^{(m)}+\hat{c} \qquad \text{for } i,j=1,2,\ldots,n. \tag{3.10}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

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

$$
\begin{aligned}
F_{RC1}\left(c_{i,j}\right) = \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2} g_{k_1-1,\ell} \sqrt{\left(c_{i,j} - h_{k_1-1,\ell}\right)^2 + \upsilon_{k_1-1,\ell}^2 + \beta} \\
+ \nu \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \left(-\mathrm{A}_1 + H(\tilde{\phi}_{k,\ell} + c_{i,j})\right)^2 \\
+ \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2-1} g_{k_2,\ell} \sqrt{\left(c_{i,j} - h_{k_2,\ell}\right)^2 + \upsilon_{k_2,\ell}^2 + \beta} \\
+ \bar{\mu} \sum_{k=k_1}^{k_2} g_{k,\ell_1-1} \sqrt{\left(c_{i,j} - \upsilon_{k,\ell_1-1}\right)^2 + h_{k,\ell_1-1}^2 + \beta} \\
+ \bar{\mu}\sqrt{2} g_{k_2,\ell_2} \sqrt{\left(c_{i,j} - \bar{\upsilon}_{k_2,\ell_2}\right)^2 + \bar{h}_{k_2,\ell_2}^2 + \frac{\beta}{2}} \\
+ \lambda_1 \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} H(\tilde{\phi}_{k,\ell} + c_{i,j}) \left(z_{k,\ell} - c_1\right)^2 \\
+ \lambda_2 \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \left(1 - H(\tilde{\phi}_{k,\ell} + c_{i,j})\right) \left(z_{k,\ell} - c_2\right)^2 \\
+ \bar{\mu} \sum_{k=k_1}^{k_2-1} g_{k,\ell_2} \sqrt{\left(c_{i,j} - \upsilon_{k,\ell_2}\right)^2 + h_{k,\ell_2}^2 + \beta} \\
+ \nu \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \Big(-\mathrm{A}_2 + \left(1 - H(\tilde{\phi}_{k,\ell} + c_{i,j})\right)\Big)^2 .
\end{aligned}
\tag{3.11}
$$

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

$$
\begin{aligned}
&\min_c \hat{F}_{RC1}(\tilde{\phi} + c), \\
&\hat{F}_{RC1}(\tilde{\phi} + c) = \lambda_1 \sum_{i,j=1}^{n} H(\tilde{\phi}_{i,j} + c)(z_{i,j} - c_1)^2 + \nu \sum_{i,j=1}^{n} \Big(-\mathrm{A}_1 + H(\tilde{\phi}_{i,j} + c)\Big)^2 \\
&\qquad + \lambda_2 \sum_{i,j=1}^{n} \Big(1 - H(\tilde{\phi}_{i,j} + c)\Big) \left(z_{i,j} - c_2\right)^2 \\
&\qquad + \nu \sum_{i,j=1}^{n} \Big(-\mathrm{A}_2 + \Big(1 - H(\tilde{\phi}_{i,j} + c)\Big)\Big)^2 .
\end{aligned}
\tag{3.12}
$$

Our implementation of the proposed multilevel method is 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 a stopping 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* $\min_{\phi_{i,j}} F_{RC}^{loc}(\phi_{i,j}, c_1, c_2)$ or (3.10) *for $i, j = 1, 2, \ldots, n$.*
3) *Iterate for t times on each coarse level $k = 2, 3, \ldots, L, L+1$:*
   - ▶ *If $k \leq L$, compute the minimiser c of* (3.11) or solve $\min_{c_{i,j}} F_{RC1}(c_{i,j})$;
   - ▶ *If $k = L+1$, solve* (3.12) 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 (3.5).
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**
$h,\ \upsilon$	$4b_k\tau_k^2$
λ_1 term	$2N$
λ_2 term	$2N$
s smoothing steps	$38b_k\tau_k^2 s$

Then, the flop counts for all levels is $\xi_{BC1} = \sum_{k=1}^{L+1} \left(4N + 4b_k\tau_k^2 + 38b_k\tau_k^2 s\right)$, where $k = 1$ (the finest) and $k = L+1$ (the coarsest). Next, we compute an upper bound for **BC1**:

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

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

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

Hence, the total flop counts for **RC1** is $\xi_{RC1} = \sum_{k=1}^{L+1} \left(8N + 4b_k\tau_k^2 + 31b_k\tau_k^2 s\right)$. This gives an upper bound for **RC1**:

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

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

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

ETNA
Kent State University and
Johann Radon Institute (RICAM)

4. The new localized models. The complexity of the above 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 reductions in many cases, e.g., for image denoising. However, segmentation is a special problem because the evolution of the level-set function ϕ is always local in selective segmentation. Below we include this locality into reformulations of the problem and explore further reduction of the $O(N \log N)$ complexity, consequently achieving 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 a fast solution are proposed in this section for the BC model [6] and the RC model [33], respectively. Subsequentially, we present the corresponding multilevel algorithms to solve them. As expected, the complexity of the new models will be directly linked to the length of the 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 illustrated in Figure 4.1. Given a level-set function ϕ (intended to represent Ω_1), a local function b defined by

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

characterizes the narrow band region domain $\Omega_\gamma = \Omega_1(\gamma) \cup \Gamma \cup \Omega_2(\gamma)$ surrounding the boundary Γ, with $\Omega_1(\gamma)$ and $\Omega_2(\gamma)$ denoting the γ-band region inside and outside of Γ, respectively. A similar notation is also used by [27, 41]. Note that $b = 1$ inside Ω_γ and 0 outside, and similarly, $b(\phi(x,y),\gamma)\,H(\phi) = 1$ inside $\Omega_1(\gamma)$ and 0 outside, i.e., we have $b(\phi(x,y),\gamma)\,(1-H(\phi)) = 1$ inside $\Omega_2(\gamma)$ and 0 outside. Furthermore, after discretization, we introduce the notation for the set falling into the γ-band where $b = 1$:

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

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

$$\min_{\Gamma,c_1,c_2} F_{BL}(\Gamma,c_1,c_2), \qquad F_{BL}(\Gamma,c_1,c_2) := \mu\int_\Gamma dgds + F^\gamma_{BL}(\Gamma,c_1,c_2),$$

where a refinement of the model is achieved by

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

In the level-set formation,

$$\begin{aligned}
&\min_{\phi,c_1,c_2} F_{BL}(\phi,c_1,c_2),\\
&F_{BL}(\phi,c_1,c_2) = \mu\int_\Omega dg\sqrt{|\nabla H(\phi)|^2+\beta}\,dxdy\\
&\qquad + \lambda_1\int_\Omega (z-c_1)^2 b(\phi,\gamma)\,H(\phi)\,dxdy\\
&\qquad + \lambda_2\int_\Omega (z-c_2)^2 b(\phi,\gamma)\,(1-H(\phi))\,dxdy.
\end{aligned} \tag{4.1}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Next, we propose a localized RC model of the form

$$\begin{aligned}
&\min_{\phi,c_1,c_2} F_{RL}\left(\phi,c_1,c_2\right),\\
&F_{RL}\left(\phi,c_1,c_2\right)=\mu\int_{\Omega} g\left(\left|\nabla z(x,y)\right|\right)\sqrt{\left|\nabla H\left(\phi\right)\right|^2+\beta}\,dxdy\\
&+\lambda_1\int_{\Omega}\left(z-c_1\right)^2 b\left(\phi,\gamma\right)H\left(\phi\right)\,dxdy+\lambda_2\int_{\Omega}\left(z-c_2\right)^2 b\left(\phi,\gamma\right)\left(1-H\left(\phi\right)\right)dxdy\\
&+\nu\Big(\int_{\Omega} b\left(\phi,\gamma\right)H\left(\phi\right)\,dxdy-\mathrm{A}_1\Big)^2+\nu\Big(\int_{\Omega} b\left(\phi,\gamma\right)\left(1-H\left(\phi\right)\right)\,dxdy-\mathrm{A}_2\Big)^2.
\end{aligned}\tag{4.2}$$

5. Multilevel algorithms for localized segmentation models. We now show how to adapt the above Algorithms 1–2 to the new formulations (4.1) and (4.2).

Multilevel algorithm for the localized BC model. Discretize the functional (4.1) as

$$\begin{aligned}
F_{BL}\left(\phi,c_1,c_2\right)&=\bar{\mu}\sum_{i,j=1}^{n-1} G_{i,j}\sqrt{\left(H_{i,j}-H_{i,j+1}\right)^2+\left(H_{i,j}-H_{i+1,j}\right)^2+\beta}\\
&+\lambda_1\sum_{i,j=1}^{n}\left(z_{i,j}-c_1\right)^2 H_{i,j}b_{i,j}+\lambda_2\sum_{i,j=1}^{n}\left(z_{i,j}-c_2\right)^2\left(1-H_{i,j}\right)b_{i,j},
\end{aligned}\tag{5.1}$$

where $G=dg$, $G_{i,j}=G(x_i,y_j)$, $(i,j)\in B(\phi)$. Minimization of (5.1) by the coordinate descent method on the finest level 1 leads to the following local minimization for only $(i,j)\in B(\phi^{(m)})$:

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

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

Further, the multilevel method for the localized BC model (4.1) at a general level for updating the block $[k_1,k_2]\times[\ell_1,\ell_2]$ amounts to minimizing the following local functional

$$\begin{aligned}
F_{BC2}\left(c_{i,j}\right)=&\bar{\mu}\sum_{\ell=\ell_1}^{\ell_2} G_{k_1-1,\ell}\sqrt{\left(c_{i,j}-h_{k_1-1,\ell}\right)^2+\upsilon_{k_1-1,\ell}^2+\beta}\\
&+\bar{\mu}\sum_{k=k_1}^{k_2-1} G_{k,\ell_2}\sqrt{\left(c_{i,j}-\upsilon_{k,\ell_2}\right)^2+h_{k,\ell_2}^2+\beta}\\
&+\bar{\mu}\sum_{\ell=\ell_1}^{\ell_2-1} G_{k_2,\ell}\sqrt{\left(c_{i,j}-h_{k_2,\ell}\right)^2+\upsilon_{k_2,\ell}^2+\beta}\\
&+\bar{\mu}\sum_{k=k_1}^{k_2} G_{k,\ell_1-1}\sqrt{\left(c_{i,j}-\upsilon_{k,\ell_1-1}\right)^2+h_{k,\ell_1-1}^2+\beta}
\end{aligned}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

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 by replacing*

$$\min_{\phi_{i,j}} F_{BC}^{loc}(\phi_{i,j}, c_1, c_2) \quad \textit{on the finest level by} \quad \min_{\phi_{i,j}} F_{BL}^{loc}(\phi_{i,j}, c_1, c_2);$$
$$\min_{c_{i,j}} F_{BC1}(c_{i,j}) \quad \textit{on coarse levels by} \quad \min_{c_{i,j}} F_{BC2}(c_{i,j}).$$

All other steps are identical.

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

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

Multilevel algorithm for the localized RC model. The functional (4.2) is discretized as

$$\begin{aligned}
F_{RL}(\phi, c_1, c_2) = \bar{\mu} &\sum_{i,j=1}^{n-1} g_{i,j} \sqrt{\left(H_{i,j} - H_{i,j+1}\right)^2 + \left(H_{i,j} - H_{i+1,j}\right)^2 + \beta} \\
&+ \lambda_1 \sum_{i,j=1}^{n} \left(z_{i,j} - c_1\right)^2 H_{i,j} b_{i,j} + \lambda_2 \sum_{i,j=1}^{n} \left(z_{i,j} - c_2\right)^2 \left(1 - H_{i,j}\right) b_{i,j} \\
&+ \nu\Big(-\mathrm{A}_1 + \sum_{i,j=1}^{n} H_{i,j} b_{i,j}\Big)^2 + \nu\Big(-\mathrm{A}_2 + \sum_{i,j=1}^{n} \left(1 - H_{i,j}\right) b_{i,j}\Big)^2.
\end{aligned} \tag{5.2}$$

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

$$\begin{aligned}
&F_{RC2}(c_{i,j}) \\
&= \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2} g_{k_1-1,\ell} \sqrt{\left(c_{i,j} - h_{k_1-1,\ell}\right)^2 + \upsilon_{k_1-1,\ell}^2 + \beta} \\
&\quad + \bar{\mu} \sum_{k=k_1}^{k_2-1} g_{k,\ell_2} \sqrt{(c_{i,j} - \upsilon_{k,\ell_2})^2 + h_{k,\ell_2}^2 + \beta} \\
&\quad + \bar{\mu} \sum_{\ell=\ell_1}^{\ell_2-1} g_{k_2,\ell} \sqrt{\left(c_{i,j} - h_{k_2,\ell}\right)^2 + \upsilon_{k_2,\ell}^2 + \beta} \\
&\quad + \bar{\mu}\sqrt{2} g_{k_2,\ell_2} \sqrt{\left(c_{i,j} - \bar{v}_{k_2,\ell_2}\right)^2 + \bar{h}_{k_2,\ell_2}^2 + \frac{\beta}{2}}
\end{aligned}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

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*

$$\min_{\phi_{i,j}} F_{RC}^{loc}(\phi_{i,j}, c_1, c_2) \quad \textit{on the finest level by} \quad \min_{\phi_{i,j}} F_{RL}^{loc}(\phi_{i,j}, c_1, c_2);$$
$$\min_{c_{i,j}} F_{RC1}(c_{i,j}) \quad \textit{on coarse levels by} \quad \min_{c_{i,j}} F_{RC2}(c_{i,j}).$$

All other steps are identical.

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

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

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 for a single object extraction. However, they are only applicable to our selective models; for global models such as the CV model, the band idea promotes local minimisers 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 have 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 [33] for the original RC model [33];
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, i.e., if an algorithm is faster or slower than or of the same magnitude as $O(N \log N)$, where $N = n^2$;

ETNA
Kent State University and
Johann Radon Institute (RICAM)

ii) to compare the quality, we use the so-called the Jaccard similarity coefficient (JSC) and the Dice similarity coefficient (DSC):

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

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 6.1. These are four images, which include 3 real medical images and 1 synthetic image (Problem 1 has a known solution, which helps computing JSC and DSC). The markers set also are shown in Figure 6.1. 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 $N = 65536, 262144, 1048576, 4194304$, i.e., we are solving large-scale problems. Our algorithms are implemented in MATLAB R2017a on a computer with an Intel Core i7 processor with CPU 3.60GHz and 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 $\kappa = 4$. During the experiments it was observed that the parameters ε and η can be in a range between $\varepsilon \in [1/n, 1]$ and $\eta \in [10^{-3}, 10^2]$.

First, we compare BC1 and BC2 using 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 6.2(a) and 6.2(b) displays the successful selective segmentation results by BC1 and BC2, respectively, for capturing one object in the Problems 1–3. We see that that the results from BC1 are quite similar to BC2. The computation times required by BC1 and BC2 to complete the selective segmentation task are tabulated below, where we observe that BC2 is about 2 times faster than BC1.

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

Second, against BC2, we test the algorithm BC0 based on additive operator splitting (AOS) [6] and 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 an image of size 1024×1024 are presented in Figure 6.3, and the results for all sizes in terms of quality and computation times needed to complete the segmentation tasks are given in Table 6.1. Columns 5 (ratios of the CPU times) show that BC0, BCP, and BC1 are of complexity $O(N \log N)$ while BC2 is of 'super'-optimal efficiency $O(\sqrt{N} \log N)$.

Clearly BC0 (the AOS method for the BC model with an added balloon force) provides an effective acceleration for images of moderate size $n \leq 256$. Significant improvement can be seen for 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 differences in the computation time of BC2 and the three other algorithms become significant as the image size increases to $n \geq 512$. The BC0 result marked with " ** " indicates that a very long time is required to complete the segmentation task. For example, one can observe that BC0 needs almost 100 times more computation time

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 6.1
*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 is $N = n^2$. 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

compared to BC2 to complete the segmentation in case of an image of size 1024×1024. We also see from the JSC and DSC values that all algorithms provide high segmentation quality.

6.2. Comparison of RC2 with RC0, RC1, and RCP. In the following experiments, we fixed the parameters $\lambda_1 = \lambda_2 = 1$, $\alpha = 0.01$, and $\beta = 10^{-4}$. During the experiments it was observed that the parameters ν, ε, and η can be in a range of $\nu \in [0.001, 0.01]$, $\varepsilon \in [1/n, 1]$, and $\eta \in [10^{-3}, 10^{-2}]$.

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 6.4(a) and 6.4(b) display 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 an image of size 1024×1024 illustrated in Figure 6.5, and the quality measures and the computation time presented in Table 6.2 show that RC2 can be 100 times faster than RC0, 17 times faster than RCP, and 4 times faster than RC1 for the case of an image of size 1024×1024. In particular, the ratios of the CPU times verify that RC0, RCP, and RC1 are of complexity $O(N \log N)$ while RC2 is of 'super'-optimal efficiency $O(\sqrt{N} \log N)$. Furthermore, the RC0 result with " ** " indicates that too much time is required 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 the readers, in Figure 6.6 we demonstrate a convergent plot based on Tables 6.1 and 6.2 of our proposed multilevel-based models (BC2 and RC2) for segmenting Problem 1 with an image of size 2048×2048. One can see that the models are fast, converging to tol in 2 iterations, that is, before the prescribed $maxit$.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 6.2
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 is* $N = n^2$.

Algorithm	Size $n \times n$	Number of iteration (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

Furthermore, we have extended the number of iterations for BC2 and RC2 up to 6 iterations and plotted the residual history in the same Figure 6.6. 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 is tested below. We shall pay particular attention to the regularizing parameter β that is known to be a sensitive parameter for the 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 for the 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 the 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 6.3.

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

Tests for the parameter γ. The bandwidth parameter γ is an important parameter to be tested. Its size determines how local the resulting segmentation will be. Below, we demonstrate the effect of setting different values of γ in BC2 and RC2. We aim to segment an organ in Problem 4 by applying BC2 and RC2 with varying γ. The results are presented in Figure 6.7. Columns 2 and 3 of Figure 6.7 display the results using three γ-values (with increasingly spread out bandwidth) for BC2 and RC2. Clearly, unless the value is too small (which results in an incorrect segmentation), in general, both BC2 and RC2 are not much sensitive to the choice of γ.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

TABLE 6.3
Dependence of BC2 and RC2 on t for the heart shape in Problem 1 (Figure 6.1).

Algorithm	t:inner iteration	Number of iteration (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

TABLE 6.4
Dependence of our new BC2 and RC2 on β for the heart shape in Problem 1 (Figure 6.1).

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
10e-1	2.258762e+09	0.9	1.0	5.168056e+10	0.9	1.0
10e-2	2.197002e+09	1.0	1.0	5.164663e+10	1.0	1.0
10e-4	2.178939e+09	1.0	1.0	5.163375e+10	1.0	1.0
10e-6	2.177950e+09	1.0	1.0	5.163266e+10	1.0	1.0
10e-8	2.176280e+09	1.0	1.0	5.163252e+10	1.0	1.0
10e-10	2.175254e+09	1.0	1.0	5.163243e+10	1.0	1.0

Tests for the parameter β. Finally, we examine the sensitivity of BC2 and RC2 with respect to the 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} for 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 (5.1), $F_{RL}(\phi, c_1, c_2)$ in equation (5.2), 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 6.4. One can see that as β decreases, the functional $F_{BL}(\phi, c_1, c_2)$ and $F_{RL}(\phi, c_1, c_2)$ get closer to each other. The segmentation quality measured by JSC and DSC increases as β decreases. This finding indicates that BC2 and RC2 are only sensitive to (unrealistic) large β but to a lesser extent to a very small β. In separate experiments, we found that the BC2 and RC2 algorithms are not much sensitive with respect to η, α, ε, and ν (involved in RC2 only), although there exist choices which give the optimal quality of segmentation.

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 as well.

In Part 1, we presented two algorithms (BC1, RC1) for solving the respective models with each algorithm having the expected optimal complexity of $O(N \log N)$ for the 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 two 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 are only applicable to our selective segmentation models. Numerical experiments have verified the complexity claims, and comparisons with related

ETNA
Kent State University and
Johann Radon Institute (RICAM)

algorithms (BC0, BCP, RC0, RCP for the standard models) show that the new algorithms are many times faster than BC0, BCP, RC0, RCP while achieving a comparable quality of segmentation.

Future works will address convexified selective variational models, such as [37], 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.

Acknowledgements. The first author acknowledges valuable supports from the Faculty of Computer and Mathematical Sciences (FSKM) Shah Alam, Universiti Teknologi MARA (Malaysia) and the Ministry of Higher Education of Malaysia. The second author is grateful to the support from the UK EPSRC grants EP/K036939/1 and EP/N014499/1. All the three anonymous reviewers are acknowledged for their helpful comments and suggestions that lead to improvements of this research.

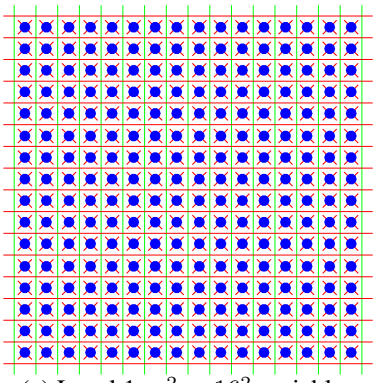
REFERENCES

[1] R. ACAR AND C. R. VOGEL, *Analysis of bounded variation penalty methods for ill-posed problems*, Inverse Problems, 10 (1994), pp. 1217–1229.
[2] R. ADAMS AND L. BISCHOF, *Seeded region growing*, IEEE Trans. Pattern Anal. Mach. Intell., 16 (1994), pp. 641–647.
[3] G. AUBERT AND P. KORNPROBST, *Mathematical Problems in Image Processing. Partial Differential Equations and the Calculus of Variations*, Springer, New York, 2002.
[4] N. BADSHAH AND K. CHEN, *Multigrid method for the Chan-Vese model in variational segmentation*, Commun. Comput. Phys., 4 (2008), pp. 294–316.
[5] ———, *On two multigrid algorithms for modelling variational multiphase image segmentation*, IEEE Trans. Image Process., 18 (2009), pp. 1097–1106.
[6] ———, *Image selective segmentation under geometrical constraints using an active contour approach*, Commun. Comput. Phys, 7 (2010), pp. 759–778.
[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 (1997), pp. 141–149.
[8] S. BEUCHER, *Segmentation tools in mathematical morphology*, in Proceedings of the SPIE on Image Algebra and Morphological Image Processing, Proceedings of SPIE 1350, SPIE, Bellingham, 1990, pp. 70–84.
[9] J. L. CARTER, *Dual Method for Total Variation-Based Image Restoration*, Ph.D. Thesis, Department of Mathematics, University of California, Los Angeles, 2002.
[10] V. CASELLES, R. KIMMEL, AND G. SAPIRO, *Geodesic active contours*, Int. J. Comput Vis., 22 (1997), pp. 61–79.
[11] T. F. CHAN AND K. CHEN, *An optimization-based multilevel algorithm for total variation image denoising*, Multiscale Model. Simul., 5 (2006), pp. 615–645.
[12] R. H. CHAN AND K. CHEN, *Multilevel algorithm for a Poisson noise removal model with total variation regularisation*, Int. J. Comput. Math., 84 (2007), pp. 1183–1198.
[13] ———, *A multilevel algorithm for simultaneously denoising and deblurring images*, SIAM J. Sci. Comput., 32 (2010), pp. 1043–1063.
[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 (2006), pp. 1632–1648.
[15] T. F. CHAN AND L. A. VESE, *Active contours without edges*, IEEE Trans. Image Process., 10 (2001), pp. 266–277.
[16] D. COMANICIU AND P. MEER, *Mean shift: a robust approach toward feature space analysis*, IEEE Trans. Pattern Anal. Mach. Intell., 24 (2002), pp. 603–619.
[17] M. COMER, C. BOUMAN, AND J. SIMMONS, *Statistical methods for image segmentation and tomography reconstruction*, Microsc. Microanalysis, 16 (2010), pp. 1852-1853.
[18] S. GEMAN AND D. GEMAN, *Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images*, IEEE Trans. Pattern Anal. Machine Intell., 6 (1984), pp. 721–741.
[19] P. GETREUER, *Rudin-Osher-Fatemi total variation denoising using split Bregman*, Image Processing On Line, 2 (2012), pp. 74–95.
[20] C. GOUT, C. LE GUYADER, AND L. A. VESE, *Segmentation under geometrical conditions with geodesic active contour and interpolation using level set methods*, Numer. Algorithms, 39 (2005), pp. 155–173.

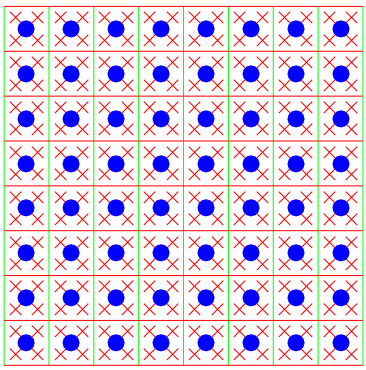
ETNA
Kent State University and
Johann Radon Institute (RICAM)

[21] C. L. GUYADER AND C. GOUT *Geodesic active contour under geometrical conditions theory and 3D applications*, Numer. Algorithms, 48 (2008), pp. 105–133.
[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, 2005, pp. 75–80.
[23] M. KASS, M. WITKIN, AND D. TERZOPOULOS, *Snakes: active contour models*, Int. J. Comput. Vis., 1 (1987), pp. 321–331.
[24] A. KENIGSBERG, R. KIMMEL, AND I. YAVNEH, *A multigrid approach for fast geodesic active contours*, Tech. Rep. CIS-2004-06, Computer Science Department, The Technion-Israel Int. Technol., Haifa, 2004.
[25] J. MALIK, T. LEUNG, AND J. SHI, *Contour and texture analysis for image segmentation*, Int. J. Comput. Vis., 43 (2001), pp. 7–27.
[26] S. MALLAT, *A Wavelet Tour Of Signal Processing*, Academic Press, San Diego, 1998.
[27] J. MILLE, R. BONE, P. MAKRIS, AND H. CARDOT, *Narrow band region-based active contours and surfaces for 2D / 3D segmentation*, J. Comput. Vis. Image Underst., 113 (2009), pp. 946–965.
[28] L. MOISAN, *How to discretize the Total Variation of an image?*, Proc. Appl. Math. Mech., 7 (2007), pp. 1041907–1041908.
[29] S. MORIGI, L. REICHEL, AND F. SGALLARI, *Noise-reducing cascadic multilevel methods for linear discrete ill-posed problems*, Numer. Algorithms, 53 (2010), pp. 1–22.
[30] D. MUMFORD AND J. SHAH, *Optimal approximation by piecewise smooth functions and associated variational problems*, Comm. Pure Appl. Math., 42 (1989), pp. 577–685.
[31] G. PAPANDREOU AND P. MARAGOS, *A fast multigrid implicit algorithm for the evolution of geodesic active contours*, in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2004, IEEE Conference Proceedings, Los Alamitos, 2004, pp. 689–694.
[32] ———, *Multigrid geometric active contour models*, IEEE Trans. Image Process., 16 (2007), pp. 229–240.
[33] L. RADA AND K. CHEN, *Improved selective segmentation model using one level set*, J. Algorithms Comput. Technol., 7 (2013), pp. 509–541.
[34] D. SEN AND S. K. PAL, *Histogram thresholding using fuzzy and rough measures of association error*, IEEE Trans. Image Process., 18 (2009), pp. 879–888.
[35] J. A. SETHIAN, *Level Set Methods and Fast Marching Methods*, 2nd ed., Cambridge University Press, Cambridge, 1999.
[36] A. I. SHIHAB, *Fuzzy Clustering Algorithms and Their Application to Medical Image Analysis*, Ph.D. Thesis, Department of Computing, University of London, London, 2000.
[37] J. SPENCER AND K. CHEN, *A convex and selective variational model for image segmentation*, Commun. Math. Sci., 13 (2015), pp. 1453–1472.
[38] J. S. WESZKA, *A survey of threshold selection techniques*, Comput. Vision Graphics Image Process., 7 (1978), pp. 259–265.
[39] J. YUAN, E. BAE, AND X. C. TAI, *A study on continuous max-flow and min-cut approaches*, in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Conference Proceedings, Los Alamitos, 2010, pp. 2217–2224.
[40] J. YUAN, E. BAE, X. C. TAI, AND Y. BOYCOV, *A study on continuous max-flow and min-cut approaches*, CAM Report 10-61, Department of Mathematics, University of California, Los Angeles, 2010.
[41] J. ZHANG, K. CHEN, B. YU, AND D. GOULD, *A local information based variational model for selective image segmentation*, Inv. Probl. Imaging, 8 (2014), pp. 293–320.

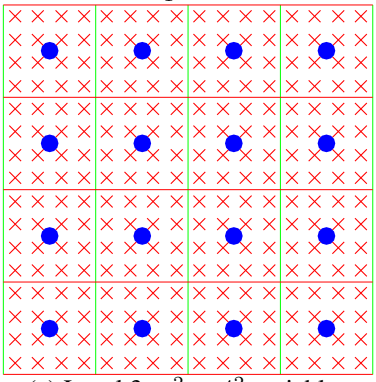
ETNA
Kent State University and
Johann Radon Institute (RICAM)



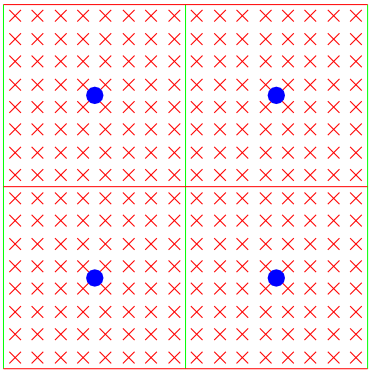
(a) Level 1: $\tau_1^2 = 16^2$ variables



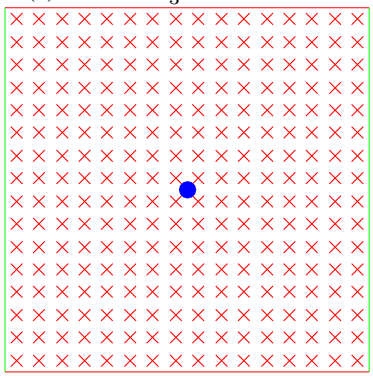
(b) Level 2: $\tau_2^2 = 8^2$ variables



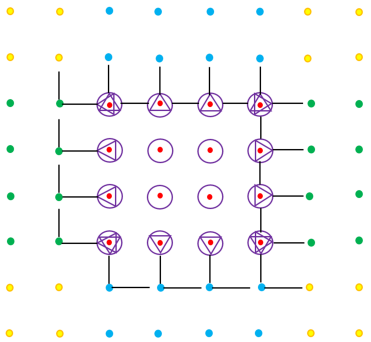
(c) Level 3: $\tau_3^2 = 4^2$ variables



(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}

FIG. 3.2. *Illustration of multilevel coarsening. Partitions (a)-(e): the red* $\times$ *shows image pixels, while the blue* $\bullet$ *illustrates the variable* c*. (f) shows on coarse level* 3 *the difference of inner and boundary pixels interacting with neighboring pixels* $\bullet$*. The middle boxes* $\odot$ *indicate the inner pixels which do not involve* c*, others boundary pixels denoted by symbols* $\lhd$, $\rhd$, Δ, ∇ *involve* c *as in* (3.4) *via* F_{BC}^{loc}.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

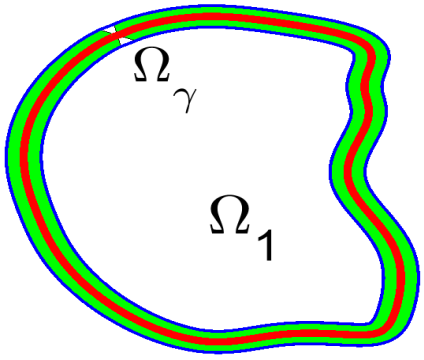


FIG. 4.1. *New modelling setup: replacement of domain Ω_1 by a smaller domain Ω_γ.*

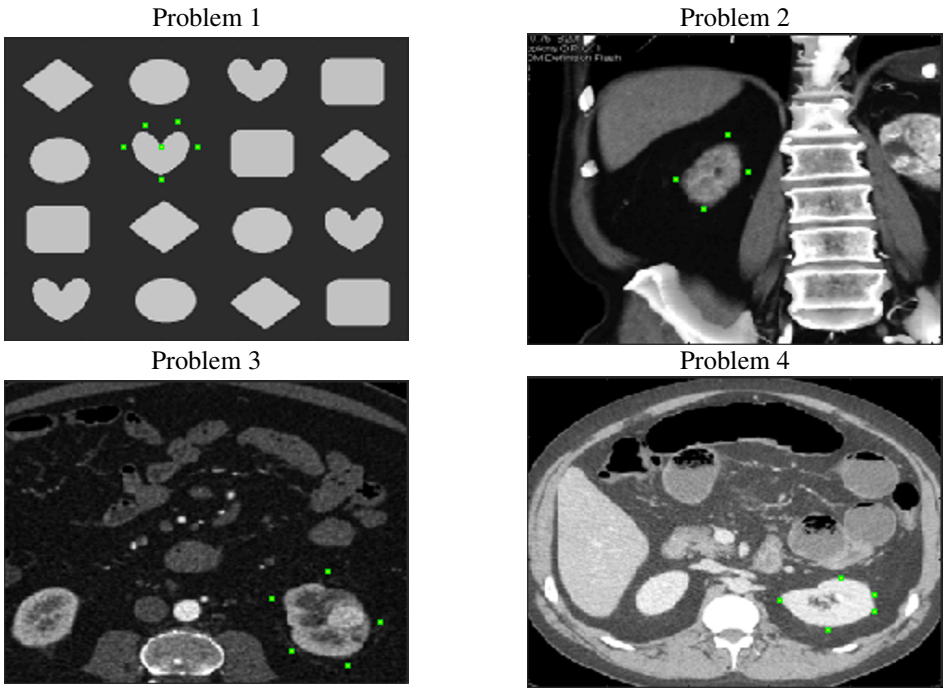


FIG. 6.1. *Test images with the markers set.*

ETNA
Kent State University and
Johann Radon Institute (RICAM)

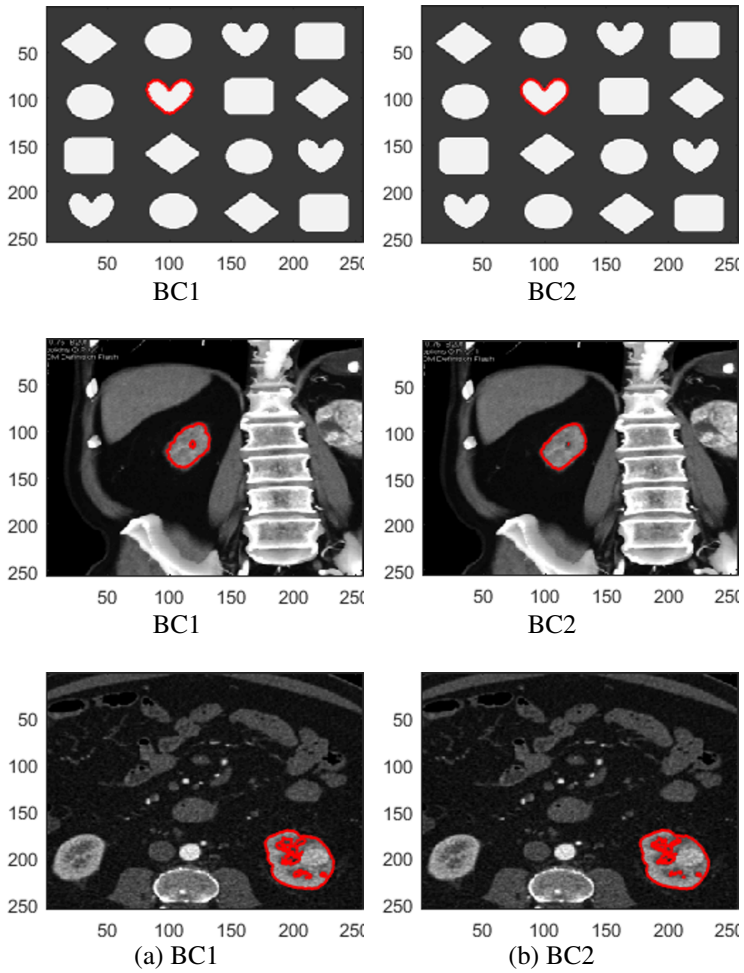


FIG. 6.2. *Segmentation of Problems 1–3: Column (a) BC1 and (b) BC2.*

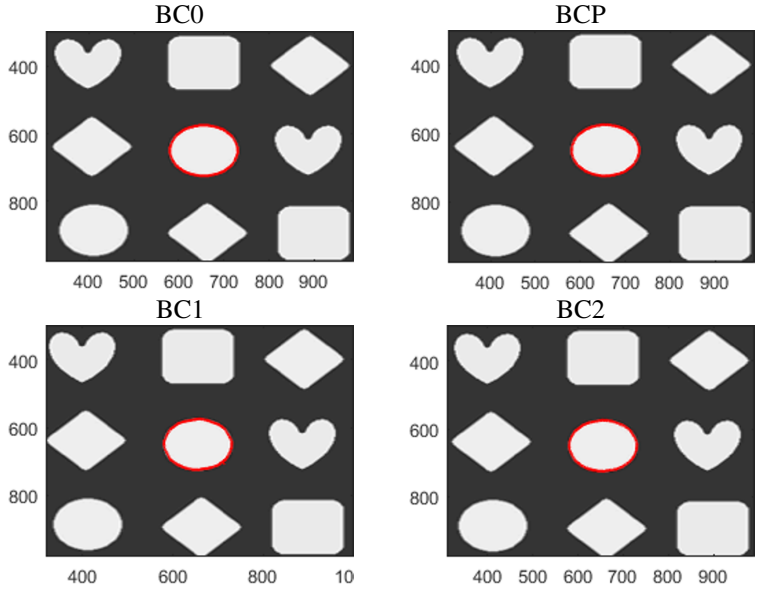


FIG. 6.3. *Segmentation of Problem 1 of size* 1024×1024 *for BC0, BCP, BC1, and BC2.*

ETNA
Kent State University and
Johann Radon Institute (RICAM)

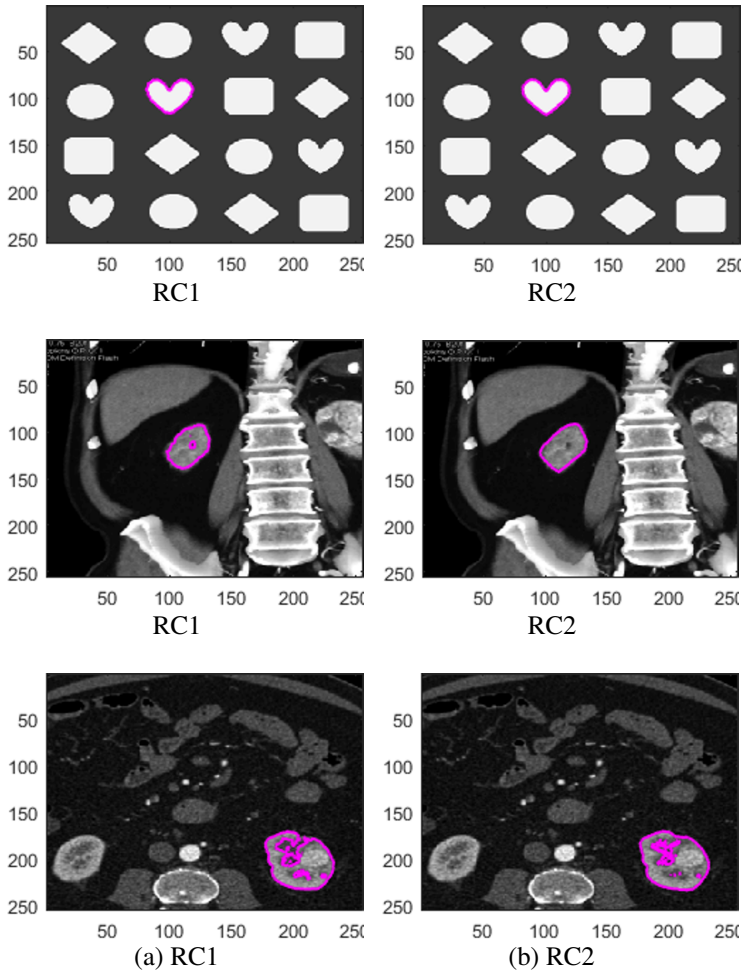


FIG. 6.4. *Segmentation of Problems 1–3. (a) and (b) show the segmentation using RC1 and RC2, respectively.*

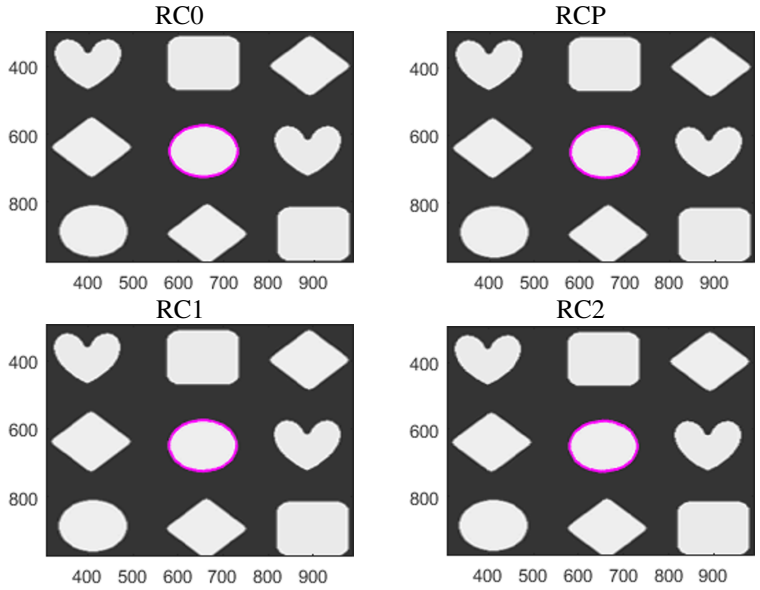


FIG. 6.5. *Segmentation of Problem 1 of size* 1024×1024 *for RC0, RCP, RC1, and RC2. For the same segmentation result, RC2 can be 100 times faster than RC0, 17 times faster than RCP and 4 times faster than RC1; see Table 6.2.*

ETNA
Kent State University and
Johann Radon Institute (RICAM)

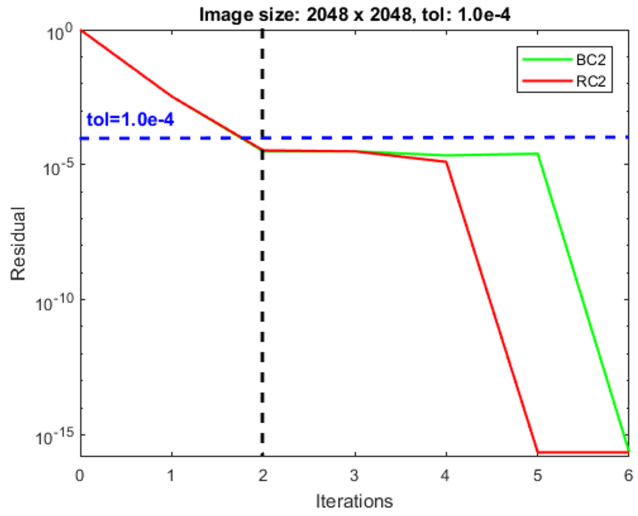


FIG. 6.6. *The number of iterations needed by BC2 and RC2 to achieve a set tol (residual) in segmenting an image of size* 2048×2048 *with* $tol = 10^{-4}$. *BC2 and RC2 need 2 iterations. The extension up to* 6 *iterations shows that the residuals of BC2 and RC2 keep decreasing.*

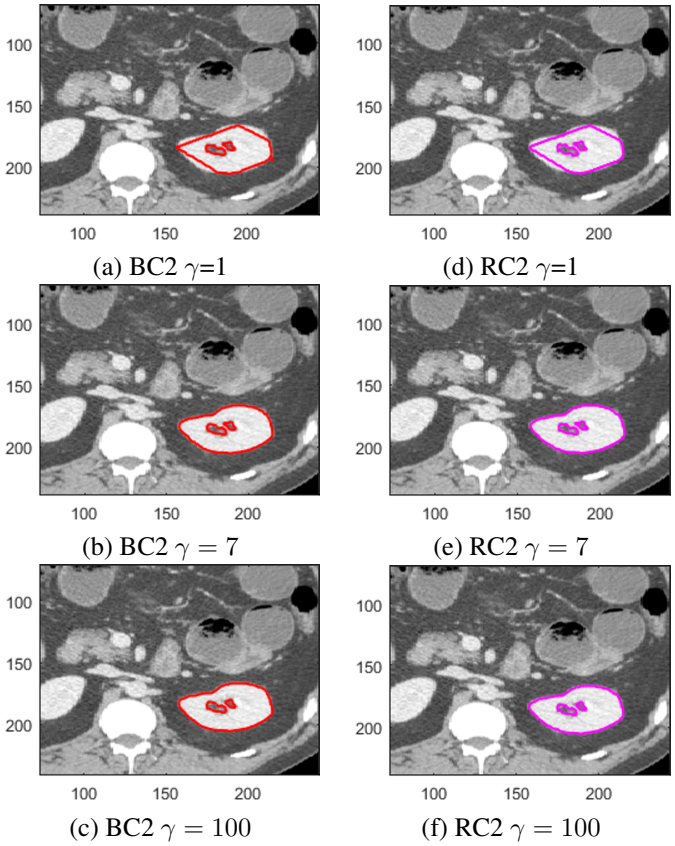


FIG. 6.7. *Dependence of algorithms BC2, RC2 on the parameter* γ *for Problem 4.*