A Reformulated Convex and Selective Variational Image Segmentation Model and its Fast Multilevel Algorithm*

Abdul K. Jumaat†and Ke Chen‡

Abstract

Selective image segmentation is the task of extracting one object of interest among many others in an image based on minimal user input. Two-phase segmentation models cannot guarantee to locate this object, while multiphase models are more likely to classify this object with another features in the image. Several selective models were proposed recently and they would find local minimizers (sensitive to initialization) because non-convex minimization functionals are involved. Recently, Spencer-Chen (CMS 2015) has successfully proposed a convex selective variational image segmentation model (named CDSS), allowing a global minimizer to be found independently of initialisation. However, their algorithm is sensitive to the regularization parameter μ and the area parameter θ due to nonlinearity in the functional and additionally it is only effective for images of moderate size. In order to process images of large size associated with high resolution, urgent need exists in developing fast iterative solvers. In this paper, a stabilized variant of CDSS model through primal-dual formulation is proposed and an optimization based multilevel algorithm for the new model is introduced. Numerical results show that the new model is less sensitive to parameter μ and θ compared to the original CDSS model and the multilevel algorithm produces quality segmentation in optimal 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 is a fundamental task in image processing aiming to obtain meaningful partitions of an input image into a finite number of disjoint homogeneous regions. 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. The snake algorithm proposed by Kass et al. [33] 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 [14, 49]. 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 [30, 9], watershed algorithm [30, 10], thresholding [30, 53], and fuzzy clustering [50]. The most celebrated (region-based) variational model for the images (with and without noise) is the Mumford-Shah [43] model, reconstructing the segmented image as a piecewise smooth intensity function. Since the model cannot be implemented directly and easily, the Mumford-Shah general model [43] was often approximated. The Chan-Vese (CV)

*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 email: k.chen@liverpool.ac.uk

[21] model is simplified and reduced from [43], without approximation. The simplification is to replace the piecewise smooth function by a piecewise constant function (of two constants $c_1,\ c_2$ or more) and, in the case of two phases, the piecewise constant function divides an image into the foreground and the background. A new variant of the CV model [21] has been proposed by [8] by taking the Euler's elastica as the regularization of segmentation contour that can yield to convex contours. Another interesting model named second order Mumford-Shah total generalized variation was developed by [26] for simultaneously performs image denoising and segmentation.

The segmentation models described above are for global segmentation due to the fact that all features or objects in an image are to be segmented (though identifying all objects is not guaranteed due to non-convexity). Selective image segmentation aims to extract one object of interest in an image based on some additional information of geometric constraints [28, 47, 52]. This task cannot be achieved by global segmentation. Some effective models are Badshah-Chen [7] and Rada-Chen [47] which used a mixed edge based and region based ideas, and area constraints. Both models are non-convex. A non-convex selective variational image segmentation model, though effective in capturing a local minimiser, is sensitive to initialisation where the segmentation result relies heavily on user input.

While the above selective segmentation models are formulated based on geometric constraints in [28, 29], there are another way of defining the geometric constraints that can be found in [41] where geometric points outside and inside a targeted object are given. Their model make use the Split Bregman method to speed up convergence. Although our paper based on geometric constraint defining in [28, 29], later, we shall compare our work with [41].We called their model as NCZZ model.

In 2015, Spencer-Chen [52, 51] has successfully designed a Convex Distance Selective Segmentation model (named as CDSS). This variational model allows a global minimiser to be found independently of initialisation, given knowledge of $c_1,\ c_2$. The CDSS model [52] is challenging to solve due to its penalty function $\nu\,(u)$ being highly nonlinear. Consequently, the standard addition operator splitting method (AOS) is not adequate. An enhanced version of the AOS scheme was proposed in [52] by taking the approximation of $\nu'\,(u)$ which based on its linear part [52, 51]. Another factor that affects the [52] model is how to choose the combination values of the regularization parameters μ and θ (other parameters can be fixed as suggested by [52, 51]). For a simple (synthetic) image, it is easy to get a suitable combination of parameter μ and θ which gives a good segmentation result. However, for other real life images, it is not trivial to determine a suitable combination of μ and θ simultaneously; our experiments show that high segmentation accuracy is given by the model in a small range of μ and θ and consequently the model is not ready for general use. Of course, it is known that an AOS method is not designed for processing large images.

We remark that the most recent, convex, selective, variational image segmentation model was by Liu *et al.* [35] in 2018. This work is based on [7, 12, 47]. We named their model as the CMT model. Although this paper is based on [52, 51], we shall compare our work with the CMT model [35] later.

Both the fast solvers multilevel and multigrid methods are developed using the idea of hierarchy of discretization. However, multilevel method is based on discretize-optimize scheme (algebraic) where the minimization of a variational problem is solved directly without using 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 [32].

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 are developed [34, 45, 46]. In 2008, Badshah and Chen [5] has successfully implemented a nonlinear multigrid method to solve an elliptical partial differential equation. In 2009, Badshah and Chen [6] have also developed two nonlinear multigrid algorithms for variational multiphase image segmentation. All these multigrid methods mentioned above are based on an optimize-discretize scheme where a multigrid method is used to solve the resulting Euler Lagrange partial differential equation (PDE) derived from the variational problem. While the practical performance of the latter methods (closer to this work) is good, however, the multigrid convergence is not achieved due to smoothers having a bad smoothing rate (and non-smooth coefficients with jumps near edges that separate segmented domains). Therefore the above nonlinear multigrid methods behave like the cascadic multigrids [42] where only one multigrid cycle is applied.

An optimization based multilevel method is based on a discretize-optimize scheme where minimization is solved directly (without using PDEs). The idea has been applied to image denoising and debluring problems [16, 17, 18]. 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 [16] 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 [32].

This paper investigates both the robust modeling and fast solution issues by making two contributions. Firstly, we propose a better model than CDSS. In looking for possible improvement on the selective model CDSS, we are inspired by several works [11, 3, 4, 15, 20, 13] on non-selective segmentation. The key idea that we will employ in our new model is the primal-dual formulation which allows us to "ignore" the penalty function $\nu(u)$, otherwise creating problems of parameter sensitivity. We remark that similar use of the primal-dual idea can be found in D. Chen et al. [22] to solve a variant of Mumford-Shah model which handles the segmentation of medical images with intensity inhomogeneities and also in Moreno et al. [40] for solving a four phase model for segmentation of brain MRI images by active contours. Secondly, we propose a fast optimization based multilevel method for solving the new model, which is applicable to the original CDSS [52], in order to achieve fast convergence especially for images with large size. We will consider the 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 similar work done for segmentation models in the variational convex formulation.

The rest of the paper is organized in the following way. In Section 2, we first briefly review the non-convex variant of the Spencer-Chen CDSS model [52]. This model gives foundation for the CDSS. In Section 3, we give our new primal-dual formulation of the CDSS model and in Section 4 present the optimization based multilevel algorithm. We proposed a new variant of the multilevel algorithm in Section 5 and discuss their convergence in Section 6. In Section 7 we give some experimental results before concluding in Section 8.

2 Review of existing variational selective segmentation models

As discussed, there exist many variational segmentation models in the literature on global segmentation and few on selective image segmentation models. For the latter, we will review two segmentation models below that are directly related to this work. We first review a nonconvex selective segmentation model called the Distance Selective Segmentation [52]. Then, we discuss the convex version of DSS called the Convex Distance Selective Segmentation model [52] before we introduce a new CDSS model based on primal-dual formulation and address the fast solution issue in these models.

Assume that an image $z = z(x, y)$ comprises of two regions of approximately piecewise constant intensities of distinct values (unknown) 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 value c_2. In a level set formulation, the unknown curve Γ is represented by the zero level set of the Lipschitz function such that

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

Let n_1 geometric constraints be given by a marker set

$$A = \{w_i = (x_i^*, y_i^*) \in \Omega, 1 \le i \le n_1\} \subset \Omega$$

where each point is near the object boundary Γ, not necessarily on it [47, 54]. The selective segmentation idea tries to detect the boundary of a single object among all homogeneity intensity objects in Ω close to A; here n_1 (≥ 3). The geometrical points in A define an initial polygonal contour and guide its evolution towards Γ [54].

It should be remarked that applying a global segmentation model first and selecting an object next amount provide an alternative to selective segmentation. However this approach would require a secondary binary segmentation and is not reliable because the first round of segmentation cannot guarantee to isolate the interested object often due to non-convexity of models.

2.1 Distance Selective Segmentation model

The Distance Selective Segmentation (DSS) model [52] was proposed by Spencer and Chen [52] in 2015. The formulation is based on the special case of the piecewise constant Mumford-Shah functional [43] where it is restricted to only two phase (i.e. constants), representing the foreground and the background of the given image $z(x, y)$.

Using the set A, construct a polygon Q that connects up the markers. Denote the function $P_d(x, y)$ as the Euclidean distance of each point $(x, y) \in \Omega$ from its nearest point $(x_p, y_p) \in Q$:

$$P_d(x, y) = \sqrt{(x - x_p)^2 + (y - y_p)^2} = \min_{q \in Q} \|(x, y) - (x_q, y_q)\|$$

and denote the regularized versions of a Heaviside function by

$$H_\varepsilon(\phi(x, y)) = \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan\left(\frac{\phi}{\varepsilon}\right)\right).$$

Then the DSS in a level set formulation is to minimize a cost function defined as follows

$$\begin{aligned}\min_{\phi, c_1, c_2} D(\phi, c_1, c_2) = \quad &\mu \int_\Omega g(|\nabla z|) |\nabla H_\varepsilon(\phi)| \, d\Omega + \int_\Omega H_\varepsilon(\phi)(z - c_1)^2 d\Omega \\ &+ \int_\Omega (1 - H_\varepsilon(\phi))(z - c_2)^2 d\Omega + \theta \int_\Omega H_\varepsilon(\phi) P_d \, d\Omega\end{aligned} \tag{1}$$

where μ and θ are nonnegative parameters. In this model $g(s) = \frac{1}{1+\gamma s^2}$ is an edge detector function which helps to stop the evolving curve on the edge of the objects in an image. The strength of detection is adjusted by parameter γ. The addition of new distance fitting term is weighted by the area parameter θ. Here, if the parameter θ is too strong the final result will just be the polygon P which is undesirable.

2.2 Convex Distance Selective Segmentation model

The above model from (1) was relaxed to obtain a constrained Convex Distance Selective Segmentation (CDSS) model [52]. This was to make sure that the initialisation can be flexible. The CDSS was obtained by relaxing $H_\varepsilon \to u \in [0, 1]$ to give:

$$\min_{0 \le u \le 1} CDSS(u, c_1, c_2) = \mu \int_\Omega |\nabla u|_g d\Omega + \int_\Omega ru \, d\Omega + \theta \int_\Omega P_d u \, d\Omega \tag{2}$$

and further an unconstrained minimization problem:

$$\min_u CDSS\left(u,c_1,c_2\right)=\mu\int_\Omega |\nabla u|_g \; d\Omega+\int_\Omega ru \; d\Omega+\theta\int_\Omega P_d u \; d\Omega+\alpha\int_\Omega \nu\left(u\right)\; d\Omega \qquad (3)$$

where $r=\left(c_1-z\right)^2-\left(c_2-z\right)^2$ and $|\nabla u|_g=g\left(|\nabla z|\right)|\nabla u|$, $\nu\left(u\right)=\max\left\{0,2\left|u-\frac{1}{2}\right|-1\right\}$ is an exact (non-smooth) penalty term, provided that $\alpha>\frac{1}{2}\|r+\theta P_d\|_{L^\infty}$ (see also [19]). For fixed c_1, c_2, μ, θ, and $\kappa\in[0,1]$, the minimizer u of (2) is guaranteed to be a global minimizer defining the object by $\sum=\{(x,y):u\left(x,y\right)\geq\kappa\}$ [52, 19, 11]. The parameter κ is a threshold value and usually $\kappa=0.5$.

In order to compute the associated Euler Lagrange equation for u they introduce the regularized version of $\nu\left(u\right)$:

$$\nu\left(u\right)=\left[\sqrt{\left(2u-1\right)^2+\varepsilon}-1\right]H\big(\sqrt{\left(2u-1\right)^2+\varepsilon}-1\big),\quad H\left(x\right)=\frac{1}{2}+\frac{1}{\pi}\arctan\left(\frac{x}{\varepsilon}\right).$$

Consequently, the Euler Lagrange equation for u in equation (3) is the following

$$\mu\nabla\left(g\frac{\nabla u}{|\nabla u|}\right)+f=0,\quad \text{in } \Omega,\qquad \frac{\partial u}{\partial\vec{n}}=0,\quad \text{on } \partial\Omega \qquad (4)$$

where $f=-r-\theta P_d-\alpha\nu'\left(u\right)$. When u is fixed, the intensity values c_1,c_2 are updated by

$$c_1(u)=\frac{\int_\Omega uz \; d\Omega}{\int_\Omega u \; d\Omega},\qquad c_2(u)=\frac{\int_\Omega\left(1-u\right)z \; d\Omega}{\int_\Omega\left(1-u\right)\; d\Omega}.$$

Notice that the nonlinear coefficient of equation (4) may have a zero denominator where the equation is not defined. A commonly adopted idea to deal with this is to introduce a positive parameter β to (4), so the new Euler Lagrange equation becomes

$$\mu\nabla\left(g\frac{\nabla u}{\sqrt{|\nabla u|^2+\beta}}\right)+f=0,\quad in \quad \Omega;\qquad \frac{\partial u}{\partial\vec{n}}=0,\quad on \quad \partial\Omega$$

which corresponds to minimize the following differentiable form of (3)

$$\min_u CDSS\left(u,c_1,c_2\right)=\mu\int_\Omega g\sqrt{|\nabla u|^2+\beta}\; d\Omega+\int_\Omega ru \; d\Omega+\theta\int_\Omega P_d u \; d\Omega+\alpha\int_\Omega \nu\left(u\right) d\Omega. \quad (5)$$

According to [52, 51], the standard AOS which generally assumes f is not dependent on u is not adequate to solve the model. This mainly because the term $\nu'\left(u\right)$ in f does depend on u, which can lead to stability restriction on time step size t. Moreover, the shape of $\nu'\left(u\right)$ means that changes in f between iterations are problematic near $u=0$ and $u=1$, as small changes in u produce large changes in f. In order to tackle the problem, they proposed a modified version of AOS algorithm to solve the model by taking the approximation of $\nu'\left(u\right)$ which based on its linear part.

A successful segmentation result can be obtained depending on suitable combination of parameter μ, θ and the set of marker points defined by a user. For a simple image such as synthetic images, this task of parameters selection is easy and one can get a good segmentation result. However, for real life images, it is non-trivial to determine a suitable combination of parameters μ and θ. It becomes more challenging if a model is sensitive to μ and θ where only a small range of the values work to give high segmentation quality. Hence, a more robust model that is less dependent on the parameters needs to be developed. In addition, to process images of large size, fast iterative solvers need to be developed as well. This paper is motivated by these two problems.

We refer to the CDSS model solved by the modified AOS as **SC0**.

3 A reformulated CDSS model

We now present our work on a reformulation of the CDSS model in the primal-dual framework which allows us to “ignore” the penalty function $\nu(u)$, otherwise creating problems of parameter sensitivity. We remark that similar use of the primal-dual idea can be found in [22] and [40]. To see more background of this framework, refer to the convex regularization approach by Bresson et al. [11], Chambolle [15], and others [3, 4, 20, 13].

Our starting point is to rewrite (3) as follows:

$$\min_{u,w} J(u,w) = \mu \int_\Omega |\nabla u|_g d\Omega + \int_\Omega rw \; d\Omega + \theta \int_\Omega P_d w \; d\Omega + \alpha \int_\Omega \nu(w) \; d\Omega + \frac{1}{2\rho} \int_\Omega (u-w)^2 \; d\Omega \tag{6}$$

where w is the new and dual variable, the right-most term enforces $w \approx u$ for sufficiently small $\rho > 0$ and $|\nabla u|_g = g(|\nabla z|)|\nabla u|$. One can observe that if $w = u$, the dual formulation is reduced to the original CDSS model [52].

After introducing the term $(u-w)^2$, it is important to note that convexity still holds with respect to u and w (otherwise finding the global minimum cannot be guaranteed). This can be shown below. Write the functional (6) as the sum of two terms:

$$J(u,w) = S(u,w) + Q(u,w), \quad S(u,w) = \int_\Omega \frac{1}{2\rho}(u-w)^2 d\Omega, \quad TV_g(u) = \int_\Omega |\nabla u|_g \, d\Omega$$
$$Q(u,w) = TV_g(u) + \int_\Omega (r + \theta P_d) w d\Omega + \alpha \int_\Omega \nu(w) \, d\Omega.$$

For the functional $Q(u,w)$, we can show that the weighted total variation term $TV_g(u)$ is convex below. The remaining two terms (depending on w only) are known to be convex from [52, 51]. By definition of convex functions, showing that the weighted total variation is a convex can be done directly. Let $u_1 \neq u_2$ be two functions and $\varphi \in [0,1]$. Then

$$\begin{aligned} TV_g(\varphi u_1 + (1-\varphi) u_2) &= \int_\Omega |\nabla(\varphi u_1 + (1-\varphi) u_2)|_g \, d\Omega \\ &= \int_\Omega |\varphi \nabla u_1 + (1-\varphi) \nabla u_2|_g \, d\Omega \\ &\leq \varphi \int_\Omega |\nabla u_1|_g \; d\Omega + (1-\varphi) \int_\Omega |\nabla u_2|_g \; d\Omega \\ &= \varphi TV_g(u_1) + (1-\varphi) TV_g(u_2). \end{aligned}$$

Similarly, for the functional $S(u,w)$, let $u, w : \Omega \subseteq \mathbb{R}^2 \to \mathbb{R}$ and $u_1 \neq u_2 \neq u_3 \neq u_4$. Then

$$\begin{aligned} S[\varphi(u_1,u_2) + (1-\varphi)(u_3,u_4)] &= S[\varphi u_1 + (1-\varphi) u_3, \varphi u_2 + (1-\varphi) u_4] \\ &= \int_\Omega [\varphi u_1 + (1-\varphi) u_3 - \varphi u_2 - (1-\varphi) u_4]^2 d\Omega \\ &= \int_\Omega [\varphi(u_1-u_2) + (1-\phi)(u_3-u_4)]^2 d\Omega \\ &\leq \varphi \int_\Omega (u_1-u_2)^2 d\Omega + (1-\varphi) \int_\Omega (u_3-u_4)^2 d\Omega \\ &= \varphi S(u_1,u_2) + (1-\varphi) S(u_3,u_4). \end{aligned}$$

Alternatively, the $Hessian\left[(u-w)^2\right] = \begin{pmatrix} 2 & -2 \\ -2 & 2 \end{pmatrix}$. Clearly the principal minors are $\Delta_1 = 2, \quad \Delta_2 = 0$ which indicates that the $Hessian[(u-w)^2]$ is positive semidefinite and so $S(u,w)$ is convex.

As the sum of two convex functions Q, S is also convex, thus $J(u,w)$ is convex.

Using the property that J is differentiable, consequently, the unique minimizer can be computed by minimizing J with respect to u and w separately, iterating the process until convergence [11, 15]. Thus, the following minimization problems are considered:

i). when w is given: $\min\limits_{u} \ J_1(u,w) = \mu \int_\Omega |\nabla u|_g d\Omega + \frac{1}{2\rho}\int_\Omega (u-w)^2 d\Omega;$

ii). when u is given: $\min\limits_{w} \ J_2(u,w) = \int_\Omega rwd\Omega + \theta \int_\Omega P_d w d\Omega + \alpha \int_\Omega \nu(w)\, d\Omega + \frac{1}{2\rho}\int_\Omega (u-w)^2\, d\Omega.$

Next consider how to simplify J_2 further and drop its α term. To this end, we make use of the following proposition:

Proposition 1 *The solution of* $\min_w J_2$ *is given by:*

$$w = \min\left\{\max\left\{u(x) - \rho r - \rho\theta P_d, 0\right\}, 1\right\}. \tag{7}$$

Proof: Assume that α has been chosen large enough compared to $\|f\|_{L^\infty}$ so that the exact penalty formulation holds. We now consider the w-minimization of the form $\min\limits_{w} \int_\Omega \left(\alpha\nu(w) + \frac{1}{2\rho}(u-w)^2 + wF(x)\right) d\Omega$, where the function F is independent of w. We use the claim made by [11].

Claim [11]**:** If $u(x) \in [0,1]$ for all x, then so is $w(x)$ after the w-minimization. Conversely, if $w(x) \in [0,1]$ for all x, then so is $u(x)$ after the u-minimization.

This claim allows us to "ignore" the $\nu(w)$ terms: on one hand, its presence in the energy is equivalent to cutting off $w(x)$ at 0 and 1. On the other hand, if $w(x) \in [0,1]$, then the above w-minimization can be written in this equivalence form: $\min\limits_{w\in(0,1)} \int_\Omega \left(\frac{1}{2\rho}(u-w)^2 + wF(x)\right) d\Omega$. Consequently, the point-wise optimal $w(x)$ is found as $\frac{1}{\rho}(u-w) = F(x) \Rightarrow w = u - \rho F(x)$. Thus the w-minimization can be achieved through the following update: $w = \min\left\{\max\left\{u(x) - \rho F(x), 0\right\}, 1\right\}$. For $\min_w J_2$, let $F(x) = r + \theta P_d$. Hence, we deduce the result for w. □

Therefore, our new model is defined as

$$\min_{u,w\in(0,1)} J(u,w) = \mu\int_\Omega |\nabla u|_g d\Omega + \int_\Omega rw\, d\Omega + \theta\int_\Omega P_d w\, d\Omega + \frac{1}{2\rho}\int_\Omega (u-w)^2\, d\Omega.$$

In alternating minimization form, the new formulation is equivalent to solve the following

$$\min_{u} \ J_1(u,w) = \mu\int_\Omega |\nabla u|_g d\Omega + \frac{1}{2\rho}\int_\Omega (u-w)^2 d\Omega, \tag{8}$$

$$\min_{w\in(0,1)} \ J_2(u,w) = \int_\Omega rw\, d\Omega + \theta\int_\Omega P_d w\, d\Omega + \frac{1}{2\rho}\int_\Omega (u-w)^2\, d\Omega. \tag{9}$$

Notice that the term $\nu(w)$ is dropped in (9) and the explicit solution is given in (7) that is hopefully the new resulting model becomes less sensitive to parameter's choice. Now it only remains to discuss how to solve (8).

4 An optimization based multilevel algorithm

This section presents our multilevel formulation for two convex models: first the CDSS model (5) (for later use in comparisons) and then our newly proposed primal-dual model in (8)-(9).

For simplicity, we shall assume $n = 2^L$ for a given image z of size $n \times n$. 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 "superpixels" 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 $L = 4$, $n = 2^4$ 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.

4.1 A multilevel algorithm for CDSS

Our goal is to solve (5) using a multilevel method in discretize-optimize scheme without approximation of $\nu'(u)$. The finite difference method is used to discretize (5) as done in related works [13, 16]. The discretized version of (5) is given by

$$\begin{aligned}\min_u CDSS(u, c_1, c_2) &\equiv \min_u CDSS^a(u_{1,1}, u_{2,1}, ..., u_{i-1,j}, u_{i,j}, u_{i+1,j}, ..., u_{n,n}, c_1, c_2) \\ &= \bar{\mu} \sum_{i=1}^{n-1}\sum_{j=1}^{n-1} g_{i,j}\sqrt{(u_{i,j}-u_{i,j+1})^2 + (u_{i,j}-u_{i+1,j})^2 + \beta}\end{aligned} \tag{10}$$

$$+\sum_{i=1}^{n}\sum_{j=1}^{n}\left((c_1 - z_{i,j})^2 - (c_2 - z_{i,j})^2\right)u_{i,j} + \theta\sum_{i=1}^{n}\sum_{j=1}^{n} P_{d_{i,j}} u_{i,j} + \alpha\sum_{i=1}^{n}\sum_{j=1}^{n}\nu_{i,j}$$

where $\bar{\mu} = \frac{\mu}{h}$, $\quad c_1 = \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} z_{i,j}u_{i,j} / \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} u_{i,j}$, $\quad c_2 = \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} z_{i,j}(1-u_{i,j}) / \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}(1-u_{i,j})$,

$$h = \frac{1}{(n-1)}, \quad \nu_{i,j} = \left[\sqrt{(2u_{i,j}-1)^2 + \varepsilon} - 1\right]\left(\frac{1}{2} + \frac{1}{\pi}\arctan\frac{\sqrt{(2u_{i,j}-1)^2+\varepsilon} - 1}{\varepsilon}\right),$$

$$g_{i,j} = (x_i, y_j) \quad \text{and} \quad P_{d_{i,j}} = (x_i, y_j).$$

Here u denotes a row vector.

As a prelude to multilevel methods, minimize (10) by a coordinate descent method (also known as relaxation algorithm) on the finest level 1:

$$\begin{aligned}&\textit{Given } u^{(m)} = \left(u_{i,j}^{(m)}\right) \textit{ with } m = 0; \\ &\textit{Solve } u_{i,j}^{(m)} = \arg\min_{u_{i,j}\in\mathbb{R}} CDSS^{loc}(u_{i,j}, c_1, c_2) \textit{ for } i,j = 1,2,...,n; \\ &\textit{Set } u_{i,j}^{(m+1)} = \left(u_{i,j}^{(m)}\right) \textit{ and repeat the above steps with } m = m+1 \textit{ until stopped.}\end{aligned} \tag{11}$$

Here equation (11) is simply obtained by expanding and simplifying the main model in (10) i.e.

$$\begin{aligned}&CDSS^{loc}(u_{i,j}, c_1, c_2) \\ &\equiv CDSS^a\left(u_{1,1}^{(m-1)}, u_{2,1}^{(m-1)}, ..., u_{i-1,j}^{(m-1)}, u_{i,j}, u_{i+1,j}^{(m-1)}, ..., u_{m,n}^{(m-1)}, c_1, c_2\right) - CDSS^{(m-1)} \\ &\qquad = \bar{\mu}\Bigg[g_{i,j}\sqrt{\left(u_{i,j} - u_{i+1,j}^{(m)}\right)^2 + \left(u_{i,j} - u_{i,j+1}^{(m)}\right)^2 + \beta} \\ &\qquad + g_{i-1,j}\sqrt{\left(u_{i,j} - u_{i-1,j}^{(m)}\right)^2 + \left(u_{i-1,j}^{(m)} - u_{i-1,j+1}^{(m)}\right)^2 + \beta} \\ &\qquad + g_{i,j-1}\sqrt{\left(u_{i,j} - u_{i,j-1}^{(m)}\right)^2 + \left(u_{i,j-1}^{(m)} - u_{i+1,j-1}^{(m)}\right)^2 + \beta}\Bigg] \\ &\qquad + u_{i,j}\left((c_1 - z_{i,j})^2 - (c_2 - z_{i,j})^2\right) + \theta P_{d_{i,j}} u_{i,j} + \alpha(\nu_{i,j})\end{aligned}$$

with Neumann's boundary condition applied where $CDSS^{(m-1)}$ denotes the sum of all terms in $CDSS^a$ that do not involve $u_{i,j}$. Clearly one seems that this is a coordinate descent method. It should be remarked that the formulation in (11) is based on the work in [13] and [16].

Using (11), we illustrate the interaction of $u_{i,j}$ with its neighboring pixels on the finest level 1 in Figure 1. We will use this basic structure to develop a multilevel method.

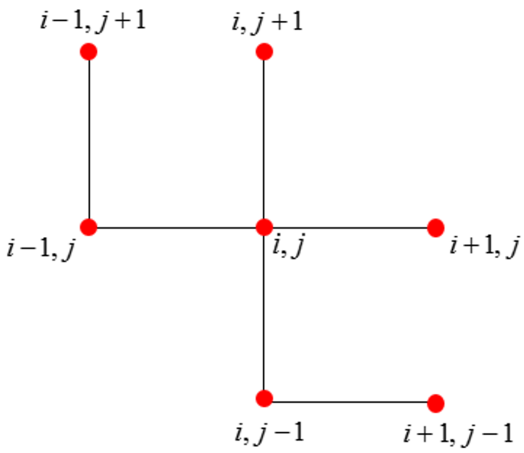


Figure 1: The interaction of $u_{i,j}$ at a central pixel (i,j) with neighboring pixels on the finest level 1. Clearly only 3 terms (pixels) are involved with $u_{i,j}$ (through regularization)

The Newton method is used to solve the one-dimensional problem from (11) by iterating $u^{(m)} \to u \to u^{(m+1)}$:

$$\begin{aligned}&\bar{\mu} g_{i,j}\frac{2u_{i,j}-u^{(m)}_{i+1,j}-u^{(m)}_{i,j+1}}{\sqrt{\left(u_{i,j}-u^{(m)}_{i+1,j}\right)^2+\left(u_{i,j}-u^{(m)}_{i,j+1}\right)^2+\beta}} + \bar{\mu} g_{i-1,j}\frac{u_{i,j}-u^{(m)}_{i-1,j}}{\sqrt{\left(u_{i,j}-u^{(m)}_{i-1,j}\right)^2+\left(u^{(m)}_{i-1,j}-u^{(m)}_{i-1,j+1}\right)^2+\beta}}\\ &+\bar{\mu} g_{i,j-1}\frac{u_{i,j}-u^{(m)}_{i,j-1}}{\sqrt{\left(u_{i,j}-u^{(m)}_{i,j-1}\right)^2+\left(u^{(m)}_{i,j-1}-u^{(m)}_{i+1,j-1}\right)^2+\beta}} + \left((c_1-z_{i,j})^2-(c_2-z_{i,j})^2\right)+\theta P_{d_{i,j}}+\alpha {\nu_{i,j}}'=0\end{aligned}$$

giving rise to the form

$$u^{new}_{i,j} = u^{old}_{i,j} - T^{old}/B^{old} \tag{12}$$

where

$$\begin{aligned}T^{old} = &\bar{\mu} g_{i,j}\frac{2u^{old}_{i,j}-u^{(m)}_{i+1,j}-u^{(m)}_{i,j+1}}{\sqrt{\left(u^{old}_{i,j}-u^{(m)}_{i+1,j}\right)^2+\left(u^{old}_{i,j}-u^{(m)}_{i,j+1}\right)^2+\beta}} + \bar{\mu} g_{i-1,j}\frac{u^{old}_{i,j}-u^{(m)}_{i-1,j}}{\sqrt{\left(u^{old}_{i,j}-u^{(m)}_{i-1,j}\right)^2+\left(u^{(m)}_{i-1,j}-u^{(m)}_{i-1,j+1}\right)^2+\beta}}\\ &+\bar{\mu} g_{i,j-1}\frac{u^{old}_{i,j}-u^{(m)}_{i,j-1}}{\sqrt{\left(u^{old}_{i,j}-u^{(m)}_{i,j-1}\right)^2+\left(u^{(m)}_{i,j-1}-u^{(m)}_{i+1,j-1}\right)^2+\beta}} + \left((c_1-z_{i,j})^2-(c_2-z_{i,j})^2\right)\\ &+\theta P_{d_{i,j}}+\alpha {\nu_{i,j}}'^{(old)}\end{aligned}$$

$$\begin{aligned}B^{old} = &\bar{\mu} g_{i,j}\frac{2}{\sqrt{\left(u^{old}_{i,j}-u^{(m)}_{i+1,j}\right)^2+\left(u^{old}_{i,j}-u^{(m)}_{i,j+1}\right)^2+\beta}} - \bar{\mu} g_{i,j}\frac{\left(2u^{old}_{i,j}-u^{(m)}_{i+1,j}-u^{(m)}_{i,j+1}\right)^2}{\sqrt{\left(\left(u^{old}_{i,j}-u^{(m)}_{i+1,j}\right)^2+\left(u^{old}_{i,j}-u^{(m)}_{i,j+1}\right)^2+\beta\right)^{\frac{3}{2}}}}\\ &+\bar{\mu} g_{i-1,j}\frac{1}{\sqrt{\left(u^{old}_{i,j}-u^{(m)}_{i-1,j}\right)^2+\left(u^{(m)}_{i-1,j}-u^{(m)}_{i-1,j+1}\right)^2+\beta}} - \bar{\mu} g_{i-1,j}\frac{\left(u^{old}_{i,j}-u^{(m)}_{i-1,j}\right)^2}{\sqrt{\left(\left(u^{old}_{i,j}-u^{(m)}_{i-1,j}\right)^2+\left(u^{(m)}_{i-1,j}-u^{(m)}_{i-1,j+1}\right)^2+\beta\right)^{\frac{3}{2}}}}\\ &+\bar{\mu} g_{i,j-1}\frac{1}{\sqrt{\left(u^{old}_{i,j}-u^{(m)}_{i,j-1}\right)^2+\left(u^{(m)}_{i,j-1}-u^{(m)}_{i+1,j-1}\right)^2+\beta}} - \bar{\mu} g_{i,j-1}\frac{\left(u^{old}_{i,j}-u^{(m)}_{i,j-1}\right)^2}{\sqrt{\left(\left(u^{old}_{i,j}-u^{(m)}_{i,j-1}\right)^2+\left(u^{(m)}_{i,j-1}-u^{(m)}_{i+1,j-1}\right)^2+\beta\right)^{\frac{3}{2}}}}\\ &+\alpha {\nu_{i,j}}''^{(old)}.\end{aligned}$$

To develop a multilevel method for this coordinate descent method, we interpret solving (11) as looking for the best correction constant $\hat{c}$ at the current approximation $u^{(m)}_{i,j}$ on level 1 (the finest level) that minimizes for c i.e.

$$\min_{u_{i,j}\in\mathbb{R}} CDSS^{loc}(u_{i,j}, c_1, c_2) = \min_{c\in\mathbb{R}} CDSS^{loc}\left(u^{(m)}_{i,j}+c, c_1, c_2\right).$$

Hence, we may rewrite (11) in an equivalent form:

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

$$\textit{Solve } \hat{c} = \arg\min_{c\in\mathbb{R}} CDSS^{loc}\left(u_{i,j}^{(m)} + c, c_1, c_2\right),\; u_{i,j}^{(m)} = u_{i,j}^{(m)} + \hat{c} \textit{ for } i,j = 1,2,...,n; \tag{13}$$

Set $u_{i,j}^{(m+1)} = \left(u_{i,j}^{(m)}\right)$ *and repeat the above steps with* $m = m+1$ *until a prescribed stopping on* m.

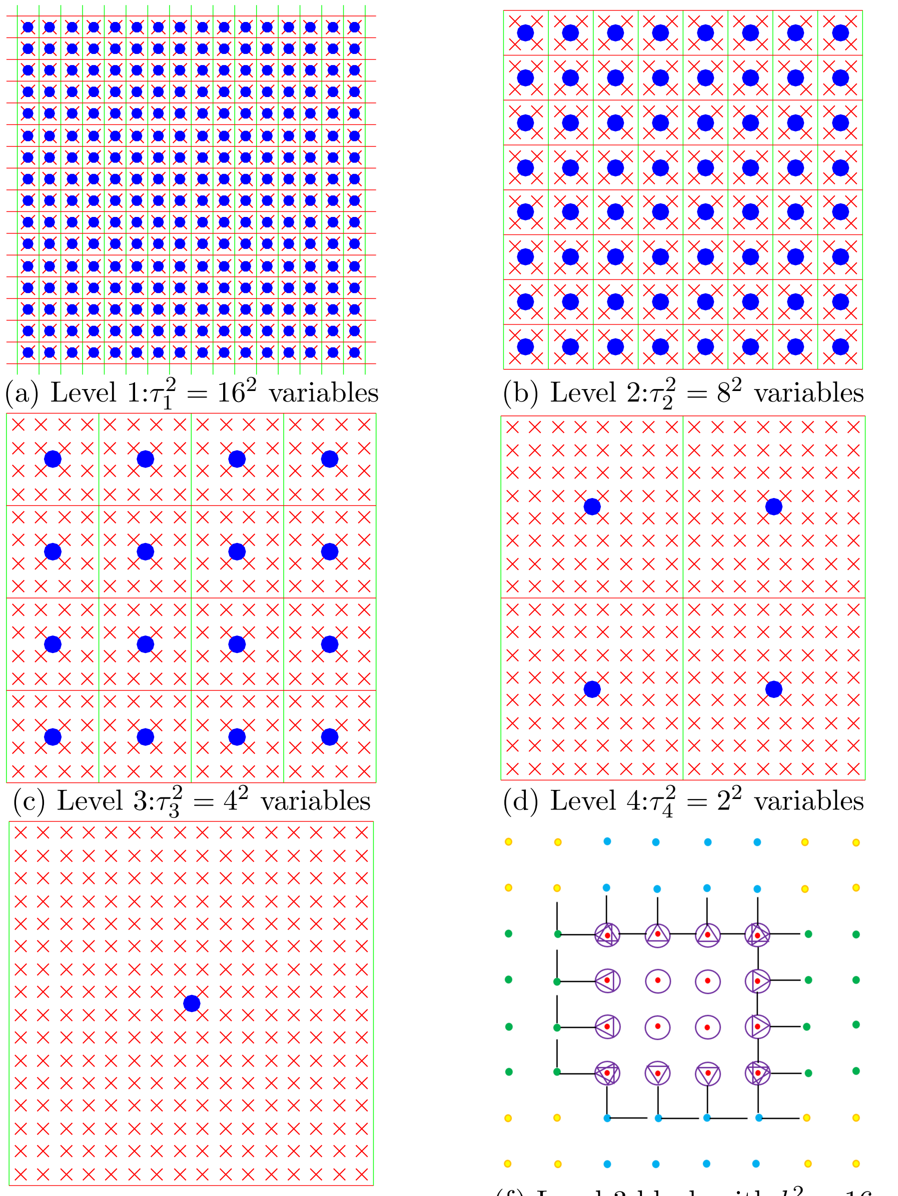
It remains to derive the simplified formulation for each of the subproblems associated with these blocks on level k e.g. the multilevel method for k=2 is to look for the best correction constant to update each 2×2 block so that the underlying merit functional, relating to all four pixels (see Fig.2(b)), 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 2(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 $\triangleleft$, $\triangleright$, 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 $c = (c_{i,j})$. Denoted the current $\tilde{u}$ then, the computational stencil involving c on level k can be shown as follows

$$\begin{array}{c|cccc|c}
\vdots & \vdots & \cdots & \vdots & & \vdots \\
\tilde{u}_{k_1-1,\ell_2+1} + c_{i-1,j+1} & \tilde{u}_{k_1,\ell_2+1} + c_{i,j+1} & \cdots & \tilde{u}_{k_2,\ell_2+1} + c_{i,j+1} & & \tilde{u}_{k_2+1,\ell_2+1} + c_{i+1,j+1} \\
\hline
\tilde{u}_{k_1-1,\ell_2} + c_{i-1,j} & \tilde{u}_{k_1,\ell_2} + c_{i,j} & \cdots & \tilde{u}_{k_2,\ell_2} + c_{i,j} & & \tilde{u}_{k_2+1,\ell_2} + c_{i+1,j} \\
\cdots & \vdots & \cdots & \vdots & & \cdots \\
\tilde{u}_{k_1-1,\ell_1} + c_{i-1,j} & \tilde{u}_{k_1,\ell_1} + c_{i,j} & \cdots & \tilde{u}_{k_2,\ell_1} + c_{i,j} & & \tilde{u}_{k_2+1,\ell_1} + c_{i+1,j} \\
\hline
\tilde{u}_{k_1-1,\ell_1-1} + c_{i-1,j-1} & \tilde{u}_{k_1,\ell_1-1} + c_{i,j-1} & \cdots & \tilde{u}_{k_2,\ell_1-1} + c_{i,j-1} & & \tilde{u}_{k_2+1,\ell_1-1} + c_{i+1,j-1} \\
\vdots & \vdots & \cdots & \vdots & & \vdots
\end{array} \tag{14}$$

The illustration shown above is consistent with Figure 2 (f) and the key point is that interior 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

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



(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 $CDSS^{loc}$

Figure 2: Illustration of partition (a)-(e). The red "×" shows image pixels, while blue • illustrates the variable c. (f) shows the difference of inner and boundary pixels interacting with neighboring pixels •. The four middle boxes ⊙ indicate the inner pixels which do not involve c, others boundary pixels denoted by symbols $\lhd$, $\rhd$, Δ, ∇ involve c as in (13) via $CDSS^{loc}$.

Then, minimizing for c, the problem (13) is equivalent to minimize the following

$$\begin{aligned}
F_{SC1}\left(c_{i,j}\right) = \bar{\mu} &\sum_{\ell=\ell_1}^{\ell_2} g_{k_1-1,\ell} \sqrt{\left[c_{i,j} - \left(\tilde{u}_{k_1-1,\ell} - \tilde{u}_{k_1,\ell}\right)\right]^2 + \left(\tilde{u}_{k_1-1,\ell} - \tilde{u}_{k_1-1,\ell+1}\right)^2 + \beta} \\
&+\bar{\mu} \sum_{k=k_1}^{k_2-1} g_{k,\ell_2} \sqrt{\left[c_{i,j} - \left(\tilde{u}_{k,\ell_2+1} - \tilde{u}_{k,\ell_2}\right)\right]^2 + \left(\tilde{u}_{k,\ell_2} - \tilde{u}_{k+1,\ell_2}\right)^2 + \beta} \\
&+\bar{\mu} g_{k_2,\ell_2} \sqrt{\left[c_{i,j} - \left(\tilde{u}_{k_2,\ell_2+1} - \tilde{u}_{k_2,\ell_2}\right)\right]^2 + \left[c_{i,j} - \left(\tilde{u}_{k_2+1,\ell_2} - \tilde{u}_{k_2,\ell_2}\right)\right]^2 + \beta} \\
&+\bar{\mu} \sum_{\ell=\ell_1}^{\ell_2-1} g_{k_2,\ell} \sqrt{\left[c_{i,j} - \left(\tilde{u}_{k_2+1,\ell} - \tilde{u}_{k_2,\ell}\right)\right]^2 + \left(\tilde{u}_{k_2,\ell} - \tilde{u}_{k_2,\ell+1}\right)^2 + \beta} \\
&+\bar{\mu} \sum_{k=k_1}^{k_2} g_{k,\ell_1-1} \sqrt{\left[c_{i,j} - \left(\tilde{u}_{k,\ell_1-1} - \tilde{u}_{k,\ell_1}\right)\right]^2 + \left(\tilde{u}_{k,\ell_1-1} - \tilde{u}_{k+1,\ell_1-1}\right)^2 + \beta} \\
&+ \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \left(\tilde{u}_{k,\ell} + c_{i,j}\right) \left(\left(c_1 - z_{k,\ell}\right)^2 - \left(c_2 - z_{k,\ell}\right)^2\right) \\
&+\theta \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \left(\tilde{u}_{k,\ell} + c_{i,j}\right) P_{d_{k,\ell}} + \alpha \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \nu\left(\tilde{u}_{k,\ell} + c_{i,j}\right)
\end{aligned} \tag{15}$$

where the third term may be simplified using $(c-a)^2 + (c-b)^2 + \beta = 2\left(c - \frac{a+b}{2}\right)^2 + 2\left(\frac{a-b}{2}\right)^2 + \beta$. Further the local minimization problem for block (i,j) on level k with respect to $c_{i,j}$ amounts to minimising the following equivalent functional

$$\begin{aligned}
F_{SC1}\left(c_{i,j}\right) = \bar{\mu} &\sum_{\ell=\ell_1}^{\ell_2} g_{k_1-1,\ell} \sqrt{\left(c_{i,j} - h_{k_1-1,\ell}\right)^2 + \upsilon_{k_1-1,\ell}^2 + \beta} + \bar{\mu} \sum_{k=k_1}^{k_2-1} g_{k,\ell_2} \sqrt{\left(c_{i,j} - \upsilon_{k,\ell_2}\right)^2 + h_{k,\ell_2}^2 + \beta} \\
&+\bar{\mu} \sum_{\ell=\ell_1}^{\ell_2-1} g_{k_2,\ell} \sqrt{\left(c_{i,j} - h_{k_2,\ell}\right)^2 + \upsilon_{k_2,\ell}^2 + \beta} + \bar{\mu} \sum_{k=k_1}^{k_2} g_{k,\ell_1-1} \sqrt{\left(c_{i,j} - \upsilon_{k,\ell_1-1}\right)^2 + h_{k,\ell_1-1}^2 + \beta} \\
&+\bar{\mu}\sqrt{2} g_{k_2,\ell_2} \sqrt{\left(c_{i,j} - \bar{\upsilon}_{k_2,\ell_2}\right)^2 + \bar{h}_{k_2,\ell_2}^2 + \frac{\beta}{2}} + \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \left(c_{i,j}\right) \left(\left(c_1 - z_{k,\ell}\right)^2 - \left(c_2 - z_{k,\ell}\right)^2\right) \\
&+\theta \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \left(\tilde{u}_{k,\ell} + c_{i,j}\right) P_{d_{k,\ell}} + \alpha \sum_{k=k_1}^{k_2} \sum_{\ell=\ell_1}^{\ell_2} \nu\left(\tilde{u}_{k,\ell} + c_{i,j}\right)
\end{aligned} \tag{16}$$

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

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

For solution on the coarsest level, we look for a **single** constant update for the current approximation $\tilde{u}$ that is

$$\begin{aligned}
\min_{c} \{F_{SC1}\left(\tilde{u} + c\right) = &\sum_{i=1}^{n} \sum_{j=1}^{n} \left(\tilde{u}_{i,j} + c\right) \left(\left(c_1 - z_{i,j}\right)^2 - \left(c_2 - z_{i,j}\right)^2\right) \\
&+\bar{\mu} \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} g_{i,j} \sqrt{\left(\tilde{u}_{i,j} + c - \tilde{u}_{i,j+1} - c\right)^2 + \left(\tilde{u}_{i,j} + c - \tilde{u}_{i+1,j} - c\right)^2 + \beta} \\
&+\theta \sum_{i=1}^{n} \sum_{j=1}^{n} P_{d_{i,j}} \left(\tilde{u}_{i,j} + c\right) + \alpha \sum_{i=1}^{n} \sum_{j=1}^{n} \nu\left(\tilde{u}_{i,j} + c\right)\}
\end{aligned}$$

which is equivalent to

$$\begin{aligned}
\min_{c} \{\, F_{SC1}\left(\tilde{u} + c\right) = &\sum_{i=1}^{n} \sum_{j=1}^{n} \left(\tilde{u}_{i,j} + c\right) \left(\left(c_1 - z_{i,j}\right)^2 - \left(c_2 - z_{i,j}\right)^2\right) \\
&+\theta \sum_{i=1}^{n} \sum_{j=1}^{n} P_{d_{i,j}} \left(\tilde{u}_{i,j} + c\right) + \alpha \sum_{i=1}^{n} \sum_{j=1}^{n} \nu\left(\tilde{u}_{i,j} + c\right)\}.
\end{aligned} \tag{17}$$

The solutions of the above local minimization problems, solved by a Newton method as in (12) or a fixed point method for t iterations (inner iteration), defines the update solution $u = u + Q_k c$ 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 (14). Then we obtain a multilevel method if we cycle through all levels and all blocks on each level until the relative error in two consecutive cycles (outer iteration) is smaller than *tol* or the maximum number of cycle, *maxit* is reached.

Finally our proposed multilevel method for CDSS is summarized in Algorithm 1. We will use the term **SC1** to refer this multilevel Algorithm 1.

Algorithm 1 SC1 – Multilevel algorithm for the CDSS model

Given z, an initial guess u, the stop tolerance (tol), and maximum multilevel cycle (maxit) with $L+1$ levels,

1) *Set $\tilde{u} = u$.*

2) *Smooth for t iteration the approximation on the finest level 1 that is solve* (11) *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 (16)
   > Solve (17) *on the coarsest level $k = L+1$*
   > Add the correction $u = u + Q_k c$ 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 (14).

4) *Check for convergence using the above criteria. If not satisfied, return to Step 1. Otherwise exit with solution $u = \tilde{u}$.*

In order to get fast convergence, it is recommended to start updating our multilevel algorithm from the fine level to the coarse level. In a separate experiment we found that if we adjust the coarse structure before the fine level, the convergence is slower. In addition, we recommend the value of inner iteration $t = 1$ is used to update the algorithm in a fast manner.

4.2 A multilevel algorithm for the proposed model

We now consider our main model as expressed by (8)–(9). Minimizations of J is with respect to u in (8) and w in (9) respectively. The solution of (9) can be obtained analytically following Proposition 1. It remains to develop a multilevel algorithm to solve (8).

Similar to the last subsection, the discretized form of the functional $J_1(u, w)$ of problem (8) is as follows:

$$\min_u \{J_1(u, w) = \bar{\mu} \sum_{i=1}^{n-1} \sum_{j=1}^{n-1} g_{i,j} \sqrt{(u_{i,j} - u_{i,j+1})^2 + (u_{i,j} - u_{i+1,j})^2 + \beta} + \frac{1}{2\rho} \sum_{i=1}^{n} \sum_{j=1}^{n} (u_{i,j} - w_{i,j})^2\} \tag{18}$$

Clearly this is a much simpler functional than the CDSS model (10) so the method can be similarly developed.

Consider the minimization of (18) by the coordinate descent method on the finest level 1:

Given $u^{(m)} = \left(u_{i,j}^{(m)}\right)$ *with* $m = 0$;

$$\textit{Solve } u_{i,j}^{(m)} = \arg\min_{u_{i,j} \in \mathbb{R}} J_1^{loc}(u_{i,j}, c_1, c_2) \textit{ for } i, j = 1, 2, ..., n; \tag{19}$$

Set $u_{i,j}^{(m+1)} = \left(u_{i,j}^{(m)}\right)$ *and repeat the above steps with* $m = m+1$ *until a prescribed stopping on* m.

Here

$$\begin{aligned} J_1^{loc}\left(u_{i,j},c_1,c_2\right) = \quad J_1 - J_0 \quad &= \bar{\mu} g_{i,j}\sqrt{\left(u_{i,j}-u_{i+1,j}^{(m)}\right)^2+\left(u_{i,j}-u_{i,j+1}^{(m)}\right)^2+\beta} \\ &+\bar{\mu} g_{i-1,j}\sqrt{\left(u_{i,j}-u_{i-1,j}^{(m)}\right)^2+\left(u_{i-1,j}^{(m)}-u_{i-1,j+1}^{(m)}\right)^2+\beta} \\ &+\bar{\mu} g_{i,j-1}\sqrt{\left(u_{i,j}-u_{i,j-1}^{(m)}\right)^2+\left(u_{i,j-1}^{(m)}-u_{i+1,j-1}^{(m)}\right)^2+\beta} \\ &+\tfrac{1}{2\rho}(u_{i,j}-w_{i,j})^2. \end{aligned}$$

The term J_0 refers to a collection of all terms that are not dependent on $u_{i,j}$. For $u_{i,j}$ at the boundary, Neumann's condition is used. Note that each subproblem in (19) is only one dimensional, which is the key to the efficiency of our new method.

To introduce the multilevel algorithm, it is of interest to rewrite (19) in an equivalent form:

$$\hat{c} = \arg\min_{c\in\mathbb{R}} J_1^{loc}\left(u_{i,j}^{(m)}+c,c_1,c_2\right), \quad u_{i,j}^{(m)} = u_{i,j}^{(m)}+\hat{c} \text{ for } i,j=1,2,...,n. \tag{20}$$

Using the stencil in (14), the problem (20) is equivalent to minimize the following

$$\begin{aligned} F_2\left(c_{i,j}\right) = \bar{\mu}&\sum_{\ell=\ell_1}^{\ell_2} g_{k_1,\ell}\sqrt{\left[c_{i,j}-\left(\tilde{u}_{k_1-1,\ell}-\tilde{u}_{k_1,\ell}\right)\right]^2+\left(\tilde{u}_{k_1-1,\ell}-\tilde{u}_{k_1-1,\ell+1}\right)^2+\beta} \\ &+\bar{\mu}\sum_{k=k_1}^{k_2-1} g_{k,\ell_2}\sqrt{\left[c_{i,j}-\left(\tilde{u}_{k,\ell_2+1}-\tilde{u}_{k,\ell_2}\right)\right]^2+\left(\tilde{u}_{k,\ell_2}-\tilde{u}_{k+1,\ell_2}\right)^2+\beta} \\ &+\bar{\mu} g_{k_2,\ell_2}\sqrt{\left[c_{i,j}-\left(\tilde{u}_{k_2,\ell_2+1}-\tilde{u}_{k_2,\ell_2}\right)\right]^2+\left[c_{i,j}-\left(\tilde{u}_{k_2+1,\ell_2}-\tilde{u}_{k_2,\ell_2}\right)\right]^2+\beta} \\ &+\bar{\mu}\sum_{\ell=\ell_1}^{\ell_2-1} g_{k_2,\ell}\sqrt{\left[c_{i,j}-\left(\tilde{u}_{k_2+1,\ell}-\tilde{u}_{k_2,\ell}\right)\right]^2+\left(\tilde{u}_{k_2,\ell}-\tilde{u}_{k_2,\ell+1}\right)^2+\beta} \\ &+\bar{\mu}\sum_{k=k_1}^{k_2} g_{k,\ell_1-1}\sqrt{\left[c_{i,j}-\left(\tilde{u}_{k,\ell_1-1}-\tilde{u}_{k,\ell_1}\right)\right]^2+\left(\tilde{u}_{k,\ell_1-1}-\tilde{u}_{k+1,\ell_1-1}\right)^2+\beta} \\ &+\frac{1}{2\rho}\sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2}\left(u_{k,\ell}+c_{i,j}-w_{k,\ell}\right)^2. \end{aligned} \tag{21}$$

After some algebraic manipulation to simplify (21), we arrive at the following

$$\begin{aligned} F_2\left(c_{i,j}\right) = \bar{\mu}&\sum_{\ell=\ell_1}^{\ell_2} g_{k_1-1,\ell}\sqrt{\left(c_{i,j}-h_{k_1-1,\ell}\right)^2+\upsilon_{k_1-1,\ell}^2+\beta}+\bar{\mu}\sum_{k=k_1}^{k_2-1} g_{k,\ell_2}\sqrt{\left(c_{i,j}-\upsilon_{k,\ell_2}\right)^2+h_{k,\ell_2}^2+\beta} \\ &+\bar{\mu}\sum_{\ell=\ell_1}^{\ell_2-1} g_{k_2,\ell}\sqrt{\left(c_{i,j}-h_{k_2,\ell}\right)^2+\upsilon_{k_2,\ell}^2+\beta}+\bar{\mu}\sum_{k=k_1}^{k_2} g_{k,\ell_1-1}\sqrt{\left(c_{i,j}-\upsilon_{k,\ell_1-1}\right)^2+h_{k,\ell_1-1}^2+\beta} \\ &+\bar{\mu}\sqrt{2}g_{k_2,\ell_2}\sqrt{\left(c_{i,j}-\bar{\upsilon}_{k_2,\ell_2}\right)^2+\bar{h}_{k_2,\ell_2}^2+\frac{\beta}{2}}+\frac{1}{2\rho}\sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2}\left(u_{k,\ell}+c_{i,j}-w_{k,\ell}\right)^2. \end{aligned} \tag{22}$$

On the coarsest level $(L+1)$, a **single** constant update for the current $\tilde{u}$ is given as

$$\min_c \left\{ F_2\left(\tilde{u}+c\right) = \frac{1}{2\rho}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(u_{i,j}+c-w_{i,j}\right)^2 \right\} \tag{23}$$

which has a simple and explicit solution.

Then, we obtain a multilevel method if we cycle through all levels and all blocks on each level. The process is stopped if the relative error in two consecutive cycles (outer iteration) is smaller than *tol* or the maximum number of cycle, *maxit* is reached.

The overall procedure to solve the new primal-dual model is given in Algorithm 2. We will use the term **SC2** to refer this algorithm to solve the proposed model expressed in (8) and (9).

Again, in order to update the algorithm in a fast manner, we recommend to adjust the fine level before the coarse level and to use the inner iteration $t = 1$.

Algorithm 2 SC2 – Algorithm to solve the new primal-dual model

Given image z, an initial guess u, the stop tolerance (tol), and maximum multilevel cycle (maxit) with $L+1$ levels. Set $w = u$,

1) *Solve* (8) *to update u using the following steps:*
   - i). *Set $\tilde{u} = u$.*
   - ii). *Smooth for t iteration the approximation on the finest level 1 that is solve* (19) *for $i, j = 1, 2, ...n$*
   - iii). *Iterate for t times on each coarse level $k = 2, 3, ...L, L+1$:*
     > If $k \leq L$, compute the minimizer c of (22)
     > Solve (23) *on the coarsest level $k = L+1$*
     > Add the correction $u = u + Q_k c$ where Q_k is the interpolation operator distributing $c_{i,j}$ to the corresponding $b \times b$ block on level k as illustrated in (14).
2) *Solve* (9) *to update w:*
   - i). *Set $\tilde{w} = w$.*
   - ii). *Compute w using the formula* (7).
3) *Check for convergence using the above criteria. If not satisfied, return to Step 1. Otherwise exit with solution $u = \tilde{u}$ and $w = \tilde{w}$*

---

5 A new variant of the multilevel algorithm SC2

Our above proposed method defines a sequence of search directions based in a multilevel setting for an optimization problem. We now modify it so that the new algorithm has a formal decaying property.

Denote the functional in (18) by $g(u): \mathbb{R}^{n^2} \to \mathbb{R}$ and represent each subproblem by

$$c^* = \operatorname*{argmin}_{c \in \mathbb{R}} g(u^\ell + cp^\ell), \quad u^{\ell+1} = u^\ell + c^* p^\ell, \quad p^\ell = \tilde{\mathbf{e}}^{\ell (\mathrm{mod}\ K)+1}, \quad \ell = 0, 1, 2, \ldots$$

where $K = \sum_{k=0}^{L} \frac{n^2}{4^k} = (4n^2 - 1)/3$ is the total number of search directions across all levels $1, 2, \ldots, L+1$ for this unconstrained optimization problem. We first investigate these search directions $\{\tilde{\mathbf{e}}\}$ and see that, noting $b_k = 2^{k-1},\ \tau = n/b_k$,

$$
\begin{aligned}
&\text{level } k=1, && \tilde{\mathbf{e}}^{j} = e_j, && j = 1,2,\ldots,n^2;\\
&\text{level } k=2, && \tilde{\mathbf{e}}^{n^2+j} = e_{s_j} + e_{s_j+1} + e_{s_j+n} + e_{s_j+n+1}, && j = 1,2,\ldots,\frac{n^2}{4},\\
& && \quad s_j = b_k[(j-1)/\tau_k]n + (j - \tau[(j-1)/\tau_k] - 1)b_k + 1; &&\\
&\text{level } k=3, && \tilde{\mathbf{e}}^{n^2+n^2/4+j} = \sum_{\ell=0}^{3}\sum_{m=0}^{3} e_{s_j+\ell n+m}, && j = 1,2,\ldots,\frac{n^2}{4^2},\\
& && \quad s_j = b_k[(j-1)/\tau_k]n + (j - \tau[(j-1)/\tau_k] - 1)b_k + 1; &&\\
&\quad\vdots && \quad\vdots && \quad\vdots\\
&\text{level } k=L+1, && \tilde{\mathbf{e}}^{K} = \sum_{\ell=0}^{n-1}\sum_{m=0}^{n-1} e_{s_j+\ell n+m} = \sum_{\ell=1}^{n^2} e_\ell, && j = n^2/4^L = 1,\\
& && \quad s_j = b_k[(j-1)/\tau_k]n + (j - \tau[(j-1)/\tau_k] - 1)b_k + 1 = 1, &&
\end{aligned}
$$

where e_j denotes the j-th unit (coordinate) vector in $\mathbb{R}^{n^2}$, and on a general level k, with $\tau_k \times \tau_k$ pixels, the $j-$th index corresponds to position $(j - \tau_k[(j-1)/\tau_k],\ [(j-1)/\tau_k]+1)$ which is, on level 1, the global position $([(j-1)/\tau_k]b_k + 1,\ (j - \tau_k[(j-1)/\tau_k] - 1)b_k + 1)$ which defines the sum of unit vectors in a $b_k \times b_k$ block – see Figure 2 (c-d). Clearly the sequence $\{p^\ell\}$ is essentially periodic (finitely many) and free-steering (spanning $\mathbb{R}^{n^2}$) [44].

Recall that a sequence $\{u^\ell\}$ is strongly downward (decaying) with respect to $g(u)$ i.e.

$$g(u^\ell) \geq g(v^\ell) \geq g(u^{\ell+1}), \quad v^\ell = (1-t)u^\ell + tu^{\ell+1} \in D_0, \ \ \forall\, t \in [0,1]. \tag{24}$$

This property is much stronger than the usual decaying property $g(u^\ell) \geq g(u^{\ell+1})$ which is automatically satisfied by our Algorithm **SC2**.

By [44, Thm 14.2.7], to ensure the minimizing sequence $\{u_\ell\}$ to be strongly downward, we modify the subproblem $\min J_1^{loc}(u^\ell + cp^\ell, c_1, c_2)$ to the following

$$u^{\ell+1} = u^\ell + c^* q^\ell, \qquad c^* = \mathrm{argmin}\{c \geq 0 \mid \nabla J^T q^\ell = 0\}, \qquad \ell \geq 0 \tag{25}$$

where the ℓ-th search direction is modified to

$$q^\ell = \begin{cases} \ \ p^\ell, & \text{if } \ \nabla J^T p^\ell \leq 0, \\ -p^\ell, & \text{if } \ \nabla J^T p^\ell > 0. \end{cases}$$

Here the equation $\nabla J^T q^\ell = 0$ for c and the local minimizing subproblem (20) i.e. $\min_c J_1^{loc}(\hat{u}_{i,j} + c, c_1, c_2)$ are equivalent. Now the new modification is to enforce $c \geq 0$ and the sequence $\{q^\ell\}$ is still essentially periodic.

We shall call the modified algorithm **SC2M**.

6 Convergence and complexity analysis

Proving convergence of the above algorithms **SC1-SC2** for

$$\min_{u \in \mathbb{R}} g(u)$$

would be a challenging task unless we make a much stronger assumption of uniform convexity for the minimizing functional g. However it turns out that we can prove the convergence of **SC2M** for solving problem (18) without such an assumption. For theoretical purpose, we assume that the underlying functional $g = g(u)$ is hemivariate i.e. $g(u + t(v-u)) = g(u)$ for t in $[0,1]$ and $u \neq v$.

To prove convergence of **SC2M**, we need to show that these 5 sufficient conditions are met

i) $g(u)$ is continuously differentiable in $D_0 = [0,1]^{n^2} \subset \mathbb{R}^{n^2}$;

ii) the sequence $\{q^\ell\}$ is uniformly linearly independent;

iii) the sequence $\{u^\ell\}$ is strongly downward (decaying) with respect to $g(u)$;

iv) $\lim\limits_{\ell\to\infty} g'(u^\ell)q^\ell/\|q^\ell\| = 0$,

v) the set $S = \{u \in D_0 \mid g'(u) = 0\}$ is non-empty.

Here $q'(u) = (\nabla g(u))^T$. Then we have the convergence of $\{u^\ell\}$ to a critical point u^* [44, Thm 14.1.4]

$$\lim_{\ell\to\infty} \inf_{u\in S} \|u^\ell - u^*\| = 0.$$

We now verify these conditions. Firstly condition i) is evident if $\beta \neq 0$ and condition ii) also holds since 'essentially periodic' implies 'uniformly linearly independent' [44, §14.6.3]. Condition v) requires an assumption of existence of stationary points for $g(u)$. Below we focus on verifying iii)-iv). From [44, Thm 14.2.7], the construction of $\{u^\ell\}$ via (25) ensures that the sequence $\{u^\ell\}$ is strongly downward and further $\lim_{\ell\to\infty} g'(u^\ell)q^\ell/\|q^\ell\| = 0$. Hence conditions iii)-iv) are satisfied.

Note condition iii) and the assumption of $g(u)$ being hemivariate imply that $\lim_{\ell\to\infty}\|u^{\ell+1} - u^\ell\| = 0$ from [44, Thm 14.1.3]. Further condition iv) and the fact $\lim_{\ell\to\infty}\|u^{\ell+1} - u^\ell\| = 0$ lead to the result $\lim_{\ell\to\infty} g'(u^\ell) = \mathbf{0}$. Finally by [44, Thm 14.1.4], the condition $\lim_{\ell\to\infty} g'(u^\ell) = \mathbf{0}$ implies $\lim_{\ell\to\infty}\inf_{u\in S}\|u^\ell - u^*\| = 0$. Hence the convergence is proved.

Next, we will give the complexity analysis of our **SC1**, **SC2** and **SC2M**. Let $N = n^2$ be the total number of pixels (unknowns). First, we compute the number of floating point operations (flops) for **SC1** for level k as follows:

Quantities	Flop counts for **SC1**
$h,\ v$	$4b_k\tau_k^2$
$\theta\, terms$	$2N$
$data\, terms$	$2N$
$\alpha\, terms$	$2N$
s smoothing steps	$38b_k\tau_k^2 s$

Then, the flop counts for all level is $W_{SC1} = \sum\limits_{k=1}^{L+1}\left(6N + 4b_k\tau_k^2 + 38b_k\tau_k^2 s\right)$ where $k = 1$ (finest) and $k = L+1$ (coarsest). Noting $b_k = 2^{k-1}, \tau_k = n/b_k, N = n^2$, we compute the upper bound for **SC1** as follows:

$$W_{SC1} = 6(L+1)N + \sum_{k=1}^{L+1}\left(\frac{4N}{b_k} + \frac{38Ns}{b_k}\right) = 6(L+1)N + (4+38s)N\sum_{k=0}^{L}\left(\frac{1}{2^k}\right)$$
$$< 6N\log n + 14N + 76Ns \approx O\left(N\log N\right)$$

Similarly, the flops for **SC2** is given as

Quantities	Flop counts for **SC2**
$h,\ v$	$4b_k\tau_k^2$
ρ term	$2N$
w term	$6N$
s smoothing steps	$31b_k\tau_k^2 s$

Hence, the total flop counts for SC2 is $W_{SC2} = 6N + \sum_{k=1}^{L+1} \left(2N + 4b_k\tau_k^2 + 31b_k\tau_k^2 s\right)$. This gives the upper bound for **SC2** as

$$W_{SC2} = 6N + 2(L+1)N + \sum_{k=1}^{L+1} \left(\frac{4N}{b_k} + \frac{31Ns}{b_k}\right) = 6N + 2(L+1)N + (4+31s)N \sum_{k=0}^{L} \left(\frac{1}{2^k}\right)$$
$$< 2N\log n + 16N + 62Ns \approx O\left(N \log N\right)$$

Finally, the approximate cost of an extra operation $\nabla J^T q^\ell$ in **SC2M** is $2N$ that results to the total flop counts for SC2M as $W_{SC2M} = 6N + \sum_{k=1}^{L+1} \left(4N + 4b_k\tau_k^2 + 31b_k\tau_k^2 s\right)$. This gives the upper bound for **SC2M** as

$$W_{SC2M} = 6N + 4(L+1)N + \sum_{k=1}^{L+1} \left(\frac{4N}{b_k} + \frac{31Ns}{b_k}\right) = 6N + 4(L+1)N + (4+31s)N \sum_{k=0}^{L} \left(\frac{1}{2^k}\right)$$
$$< 4N\log n + 18N + 62Ns \approx O\left(N \log N\right)$$

One can observe that both **SC1**, **SC2** and **SC2M** are of the optimal complexity $O(N \log N)$ expected of a multilevel method and $W_{SC1} > W_{SC2M} > W_{SC2}$.

7 Numerical experiments

This section will demonstrate the performance of the developed multilevel methods through several experiments. The algorithms to be compared are:

Name	Algorithm		Description
CMT	Old	:	The selective segmentation model proposed by Liu *et al.* [35] solved by a multilevel algorithm.
NCZZ	Old	:	The interactive image segmentation model proposed by Nguyen *et al.* [41] solved by a Split Bregman method.
BC	Old	:	The selective segmentation model proposed by Badshah and Chen [7] solved by an AOS algorithm.
RC	Old	:	The selective segmentation model proposed by Rada and Chen [47] solved by an AOS algorithm.
SC0	Old	:	The modified AOS algorithm [52] for the CDSS model [52].
SC1	New	:	The multilevel Algorithm 1 for the CDSS model [52].
SC2	New	:	The multilevel Algorithm 2 for the new primal-dual model (8)–(9).
SC2M	New	:	The modified multilevel algorithm for SC2.

There are five sets of tests carried out. In the **first set**, we will choose the best multilevel algorithm among SC1, SC2 and SC2M by comparing their segmentation performances in terms of CPU time (in seconds) and quality. The segmentation quality is measured based on the Jaccard similarity coefficient (JSC):

$$JSC = \frac{|S_n \cap S_*|}{|S_n \cup S_*|}$$

where S_n is the set of the segmented domain u and S_* is the true set of u (which is only easy to obtain for simple images). The similarity functions return values in the range $[0, 1]$. The value 1 indicates perfect segmentation quality while the value 0 indicates poor quality.

In the **second set**, we will perform the speed, quality, and parameter sensitivity test for the chosen multilevel algorithm (from set 1) and compare its performance with SC0. In the **third, fourth**, and **fifth set**, we will perform the segmentation quality comparison of the chosen

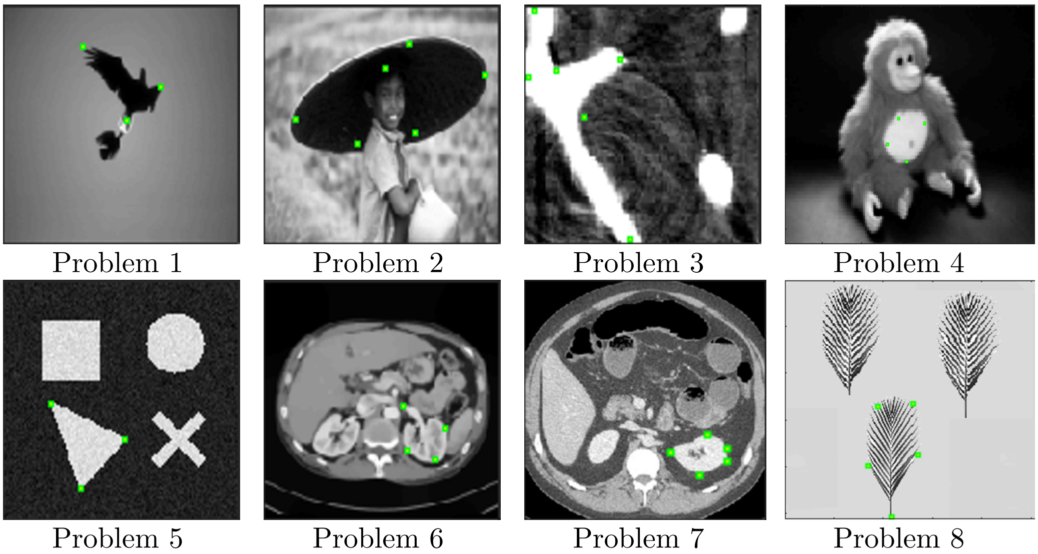


Figure 3: Segmentation test images and markers.

multilevel algorithm (from set 1) with CMT model [35], NCZZ model [41], and BC model [7] and RC model [47] respectively.

The test images used in this paper are listed in Figure 3. We remark that Problems 1-2 are obtained from the Berkeley segmentation dataset and benchmark [38], while Problems 3-4 are obtain from database provided by [25]. All algorithms are implemented in MATLAB R2017a on a computer with Intel Core i7 processor, CPU 3.60GHz, 16 GB RAM CPU.

As a general guide to choose suitable parameters for different images, our experimental results recommend the following. The parameters $\bar{\mu} = \mu$ can be between 10^{-5} and 5×10^5, $\beta = 10^{-4}$, ρ in between 10^{-5} and 10^{-1}, and γ in between $1/255^2$ and 10. Tuning the parameter θ depends on the targeted object. If the object is too close to a nearby boundary then θ should be large. Segmenting a clearly separated object in an image needs just a small θ.

7.1 Test Set 1: Comparison of SC1, SC2, and SC2M

In the first experiment, we compare the segmentation speed and quality for SC1, SC2 and SC2M using test Problem 1-4 with size of 128×128. Here, we take $\bar{\mu} = 1$, $\beta = 10^{-4}$, $\rho = 10^{-3}$, $\theta = 1000$ (Problem 1-3), $\theta = 2000$ (Problem 4), $\varepsilon = 0.12$, $\gamma = 10$, $tol = 10^{-2}$ and $maxit = 10^4$.

Figure 4 shows successful selective segmentation results by SC1, SC2 and SC2M for Problem 4. The segmentation quality for all algorithms is the same (JSC=0.96). However, SC2 performs faster (4.9 seconds) than SC1 (10.5 seconds) and SC2M (6.3 seconds).

The remaining results are tabulated in Table 1. We can see for all four test problems, SC2 gives the highest accuracy and performs the fastest compared to SC1 and SC2M.

Next, we test the performance of all the multilevel algorithms to segment Problem 5 in different resolutions. We take $\bar{\mu} = 1$, $\beta = 10^{-4}$, $\rho = 10^{-5}$, $\theta = 5000$, $\varepsilon = 0.12$, $\gamma = 10$, $tol = 10^{-3}$ and $maxit = 10^4$. The segmentation results for image size 1024×1024 are shown in Figure 5. The CPU times needed by SC2 to complete the segmentation of image size 1024×1024 is 413.2s while SC1 and SC2M need 690.6s and 636.1s respectively which implies that SC2 can be 277s faster than SC1 and 222s faster than SC2M. All the algorithms reach equal quality of segmentation.

The remaining result in terms of quality and CPU time are tabulated in Table 2. Column 6 (ratios of the CPU times) shows that SC1, SC2 and SC2M are of complexity $O(N \log N)$.

Table 1: Test Set 1 – Comparison of computation time (in seconds) and segmentation quality of SC1, SC2, and SC2M for Problem 1- 4. Clearly, for all four test problems, SC2 gives the highest accuracy and performs fast segmentation process compared to SC1 and SC2M.

Algorithm	Problem	Iteration	CPU time (s)	JSC
SC1	1	6	7.0	0.82
	2	12	20.0	0.82
	3	15	24.4	0.91
	4	6	10.5	0.96
SC2	1	5	5.9	0.82
	2	8	8.7	0.82
	3	4	4.9	0.91
	4	4	4.9	0.96
SC2M	1	5	7.9	0.79
	2	8	11.7	0.82
	3	5	7.9	0.85
	4	4	6.3	0.96

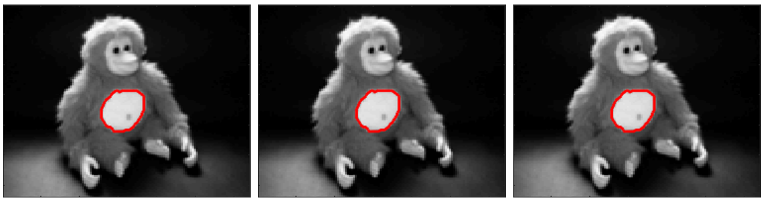


Figure 4: Test Set 1 – Segmentation of Problem 4 using our multilevel algorithms SC1, SC2, and SC2M with same quality (JSC=0.96) achieved. However, SC2 performs faster (4.9 seconds) compared to SC1 (10.5 seconds) and SC2M (6.3 seconds).

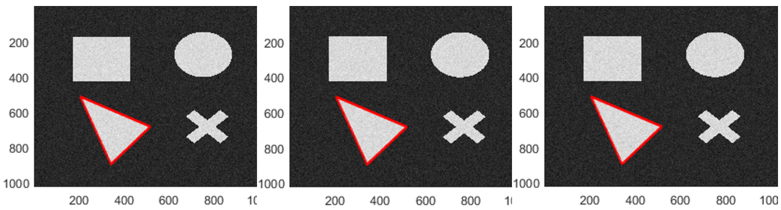
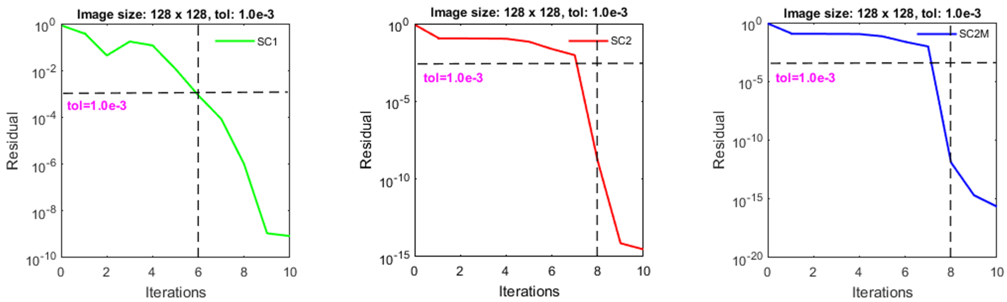


Figure 5: Test Set 1 – Segmentation of Problem 5 of size 1024x1024 for SC1, SC2, and SC2M. SC2 can be 277 seconds faster than SC1 and 222 seconds faster than SC2M : see Table 2. All algorithms give similar segmentation quality.

Table 2: Test Set 1 – Comparison of computation time (in seconds) and segmentation quality of SC1, SC2 and SC2M for Problem 5. The time ratio, t_n/t_{n-1} close to 4.4 indicates $O(N \log N)$ speed. Clearly, all algorithms have similar quality but SC2 is faster than SC1 and SC2M for all image sizes.

Algorithm	Size $N = n \times n$	Unknowns N	Iteration	Time, t_n	$\frac{t_n}{t_{n-1}}$	JSC
SC1	128×128	16384	6	10.6		1.0
	256×256	65536	7	43.5	4.1	1.0
	512×512	262144	7	173.7	4.0	1.0
	1024×1024	1048576	7	690.6	4.0	1.0
SC2	128×128	16384	8	8.7		1.0
	256×256	65536	7	23.7	2.7	1.0
	512×512	262144	8	103.9	4.4	1.0
	1024×1024	1048576	8	413.2	4.0	1.0
SC2M	128×128	16384	8	11.6		1.0
	256×256	65536	7	36.5	3.1	1.0
	512×512	262144	8	156.7	4.3	1.0
	1024×1024	1048576	8	636.1	4.1	1.0



SC1 SC2 SC2M

Figure 6: Test Set 1 – The residual plots for SC1, SC2, and SC2M to illustrate the convergence of the algorithms. The extension up to 10 iterations shows that the residual of the algorithms keep reducing. The residual for SC2 and SC2M decrease rapidly compared to SC1.

Again, we can see that for all image sizes, all algorithms have equal quality but SC2 is faster than other algorithms.

To illustrate the convergence of our multilvel algorithms, we plot in Figure 6 the residuals of SC1, SC2 and SC2M in segmenting Problem 5 for size 128×128 based on Table 2. There we extend the iterations up to 10. As we can see, the residuals of the algorithms keep reducing. The residuals for SC2 and SC2M decrease more rapidly than SC1.

Based on the experiments above, we observe that SC2 performs faster than the other two multilevel algorithms. In addition, for all problems tested, SC2 gives the higher segmentation quality than SC1 and SC2M. Therefore in practice, we recommend SC2 as the better multilevel algorithm for our convex selective segmentation method.

7.2 Test Set 2: Comparison of SC2 with SC0

The second set starts with the speed and quality comparison of SC2 with SC0 in segmenting Problem 5 with multiple resolutions. We take $\bar{\mu} = \mu = 1$, $\beta = 10^{-4}$, $\rho = 10^{-5}$, $\theta = 5000$, $\varepsilon = 0.01$, $\gamma = 10$, $tol = 10^{-6}$ and $maxit = 5000$.

Table 3: Test Set 2 – Comparison of computation time (in seconds) and segmentation quality of SC0 and SC2 for Problem 5 with different resolutions. Again, the time ratio, $t_n/t_{n-1} \approx 4.4$ indicates $O(N \log N)$ speed since $N_L = n_L^2 = (2^L)^2 = 4^L$ and $kN_L \log N_L/(kN_{L-1} \log N_{L-1}) = 4L/(L-1) \approx 4.4$. Clearly, all algorithms have similar quality but SC2 is faster than SC0 for all image sizes. Here, (**) means taking too long to run. For image size 512×512, SC2 performs 33 times faster than SC0.

Algorithm	Size $N = n \times n$	Time, t_n	$\frac{t_n}{t_{n-1}}$	JSC
SC0	128×128	243.5		1.0
	256×256	872.7	3.6	1.0
	512×512	3803.1	4.4	1.0
	1024×1024	**	**	**
SC2	128×128	8.6		1.0
	256×256	27.2	3.2	1.0
	512×512	112.0	4.1	1.0
	1024×1024	453.6	4.1	1.0

The segmentation results are tabulated in Table 3. The ratios of the CPU times in column 4 show that SC0 and SC1 are of complexity $O(N \log N)$. The symbols (**) indicates that too much time is taken to complete the segmentation task. For all image sizes, SC0 and SC2 give the same high quality.

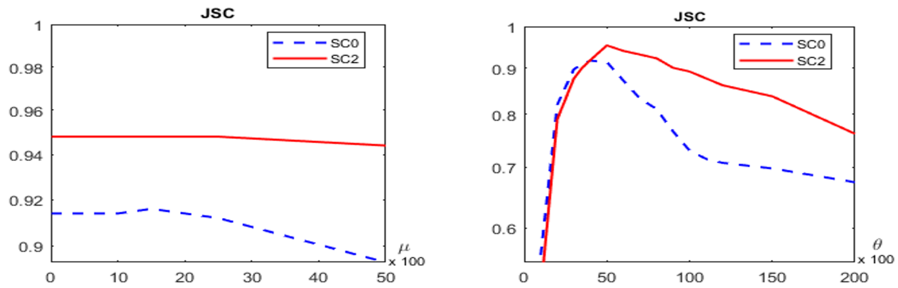
Next, we shall test parameter sensitivity for our recommended SC2. We focus on three important parameters: the regularization parameter μ, the regularising parameter β and the area parameter θ. The SC2 results are compared with SC0.

Test on parameter μ. The regularization parameter μ in a segmentation model not only controls a balance of the terms but also implicitly defines the minimal diameter of detected objects among a possibly noisy background [54]. Here, we test sensitivity of SC2 for different regularization parameters μ in segmenting an object in Problem 6 and compare with SC0 in terms of segmentation quality. We set $\beta = 10^{-4}$, $\rho = 10^{-5}$, $\varepsilon = 0.01$, $\gamma = 1/255^2$, $\theta = 5000$, $tol = 10^{-5}$ and $maxit = 10^4$.

Figure 7a shows the value of JSC for SC0 and SC2 respectively for different values of μ. Clearly, SC2 is successful for larger range of μ than SC0. This finding implies that SC2 is less dependent to parameter μ than SC0.

Test on area parameter θ. As a final comparison of SC0 and SC2, we will test how the area parameter θ effects the segmentation quality of SC0 and SC2. For this comparison, we use Problem 6 and set $\bar{\mu} = \mu = 100$, $\beta = 10^{-4}$, $\rho = 10^{-3}$, $\varepsilon = 0.01$, $\gamma = 1/255^2$, $tol = 10^{-5}$ and $maxit = 10^4$. Figure 7b shows the value of JSC for SC0 and SC2 respectively for different values of θ. We observe that SC2 is successful for a larger range of θ than SC0. This finding implies that SC2 is less sensitive to parameter θ than SC0.

Test on parameter β. Finally, we examine the sensitivity of our proposed SC2 on parameter β. The parameter β is used to avoid singularity or to ensure the original cost function is differentiable and it should be as small as possible (close to 0) so that the modified cost function (having β) in (18) is close to the original cost function in (8). We have chosen to segment an object (organ) in Problem 6. Six different values of β are tested: $\beta = 1, \quad 10^{-1}, \quad 10^{-5}, \quad 10^{-10}$, and 10^{-15}. Here, $\bar{\mu} = 100$, $\rho = 10^{-3}$, $\theta = 5500$, $\gamma = 1/255^2$, $tol = 10^{-3}$ and $maxit = 10^4$. For quantitative analysis, we compute the energy value in equation (6) (that has no β) and the JSC value. Both values are tabulated in Table 4. One can see that as β decreases, the energy value gets closer to each other. The segmentation quality measured by JSC values remain the same as β decreases. This result indicates that SC2 is not sensitive to β; large energy values for large β are expected.



(a) (b)

Figure 7: Test Set 2 – The segmentation accuracy for SC0 and SC2 in segmenting Problem 6 using different values of parameter μ in (a) and parameter θ in (b). The results demonstrate that SC2 is successful for a much larger range for both parameters.

Table 4: Test Set 2 – Dependence of our SC2 on β for segmenting Problem 6 in Figure 3.

β	JSC	Energy
1	0.95	-5.326416e+04
10^{-1}	0.95	-5.325908e+04
10^{-5}	0.95	-5.326213e+04
10^{-10}	0.95	-5.326153e+04
10^{-15}	0.95	-5.326122e+04

7.3 Test Set 3: Comparison of SC2 with CMT model [35]

In this test set 3, we investigate how the number of markers and threshold values will effect the segmentation quality for CMT model [35] and our SC2. For this purpose, we use the test Problem 4. We set $\bar{\mu} = 10^{-5}$, $\beta = 10^{-4}$, $\rho = 20$, $\theta = 3.5$, $\gamma = 20$, $tol = 10^{-3}$ and $maxit = 10^4$. The first row in Figure 8 shows the Problem 4 with different number of markers. There are 4 markers in (a1), 6 markers in (b1) and 9 markers used in (c1). The results given by CMT and SC2 using the markers with different threshold value are plotted respectively in the second row.

We observe that CMT performs well only when the number of markers used is large while our SC2 is less sensitive to the number of markers used. In addition, it is clearly shown that the range of threshold values that work for SC2 is wider than CMT. Consequently, our SC2 is more reliable than CMT.

7.4 Test Set 4: Comparison of SC2 with NCZZ model [41]

For almost all of the test images in Figure 3, we see that the NCZZ model [41] gives same satisfactory results as our SC2. For brevity, we will not show too many cases where both models give satisfactory results; Figure 9 shows the successful segmentation of an organ in Problem 7 of size 256 × 256 by NCZZ model. There two types of markers are used to label foreground region (red) and background region (blue) for the NCZZ model [41] as shown in Figure 9(a). Successful segmentation results (zoom in) by NCZZ model [41] and our SC2 for Problem 7 are shown in (b) and (c) respectively using the following parameters; $\bar{\mu} = 0.01$, $\beta = 10^{-4}$, $\rho = 10^{-3}$, $\theta = 3000$, $\gamma = 10$, $tol = 10^{-2}$ and $maxit = 10^4$.

However, according to the authors [41], the model unable to segment semi-transparent boundaries and sophisticated shapes (such as bush branches or hair in a clean way. In Figure 10, we demonstrate the limitation of NCZZ model using Problems 1 and 8. The set of parameters

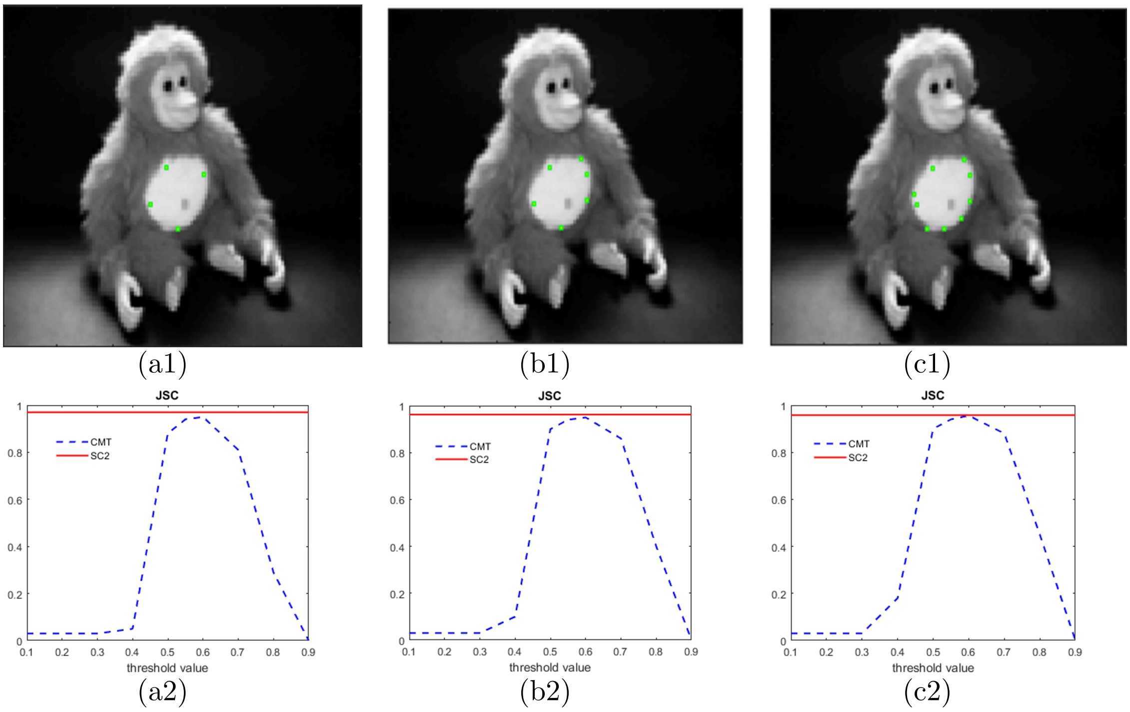


Figure 8: Test Set 3 – Comparison of SC2 with CMT model [35]. First row shows different numbers of markers used for Problem 4. Second row demonstrates the respective results (a2), (b2) and (c2) for (a1), (b1) and (c1) with different threshold values. Clearly, CMT performs well only when the number of markers used is large while our SC2 seems less sensitive to the number of markers used. Furthermore, the range of threshold value that works for SC2 is wider than CMT.

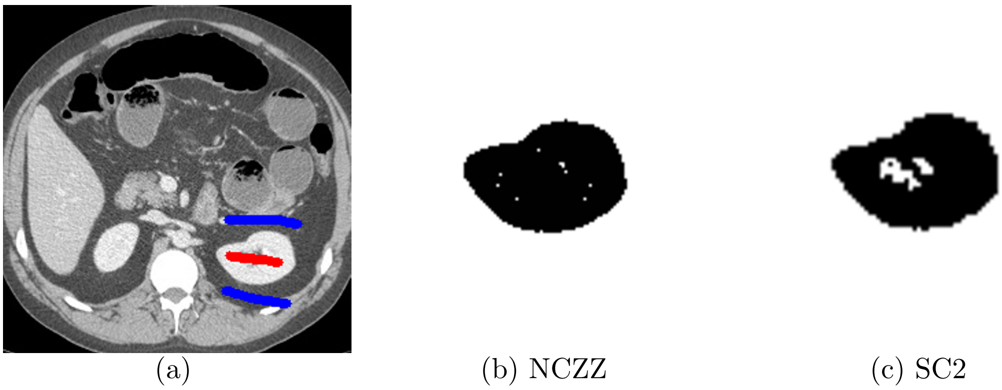
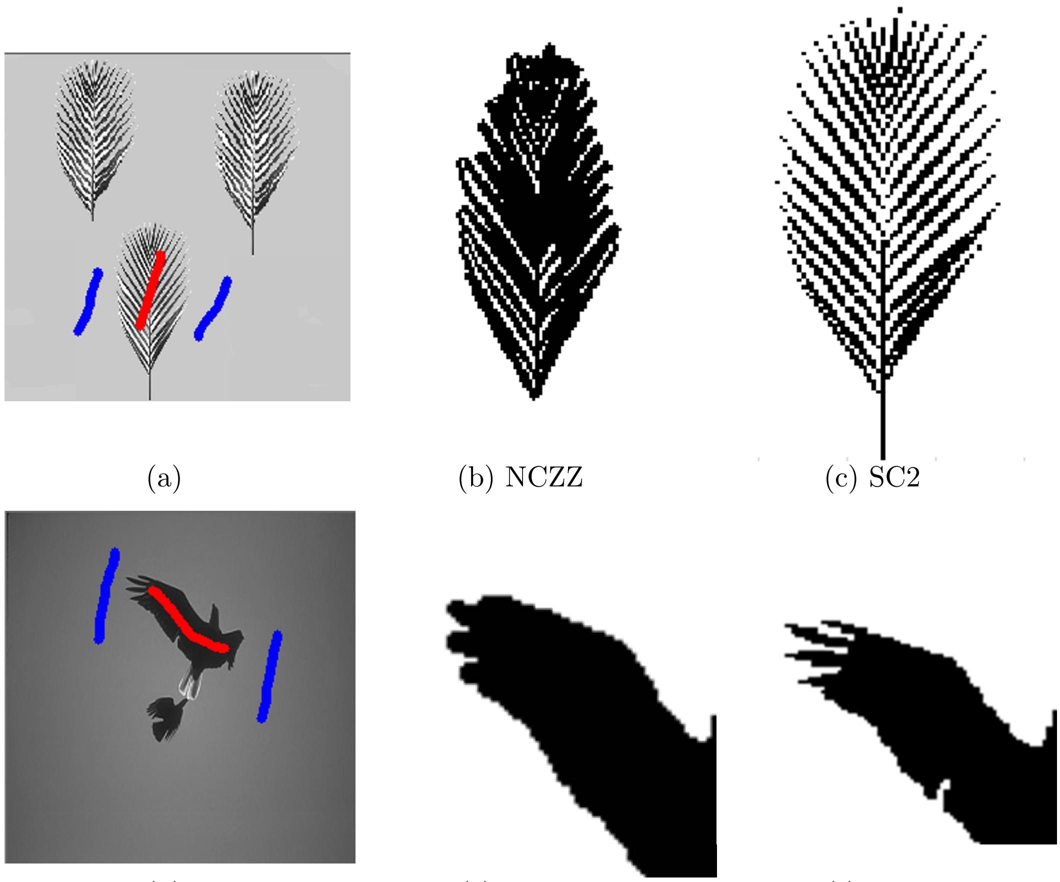


Figure 9: Problem 7 in Test Set 4 – Two types of markers used to label foreground region (red) and background region (blue) for NCZZ model [41] in (a). Successful segmentation result (zoom in): (b) by NCZZ model [41] and (c) by our SC2 (only using foreground markers).



(a) (b) NCZZ (c) SC2

(d) (e) NCZZ (f) SC2

Figure 10: Problems 1,8 in Test Set 4 – (a) and (d) show the foreground markers (red) and background markers (blue) for NCZZ model [41]. Zoomed segmentation results in (b) and (e) demonstrate the limitation of NCZZ model [41] that is unable to segment semi-transparent boundaries and sophisticated shapes (such as bush branches or hair as explained in [41]) in a clean way. Our SC2 gives cleaner segmentation for the same problems as illustrated in (c) and (f).

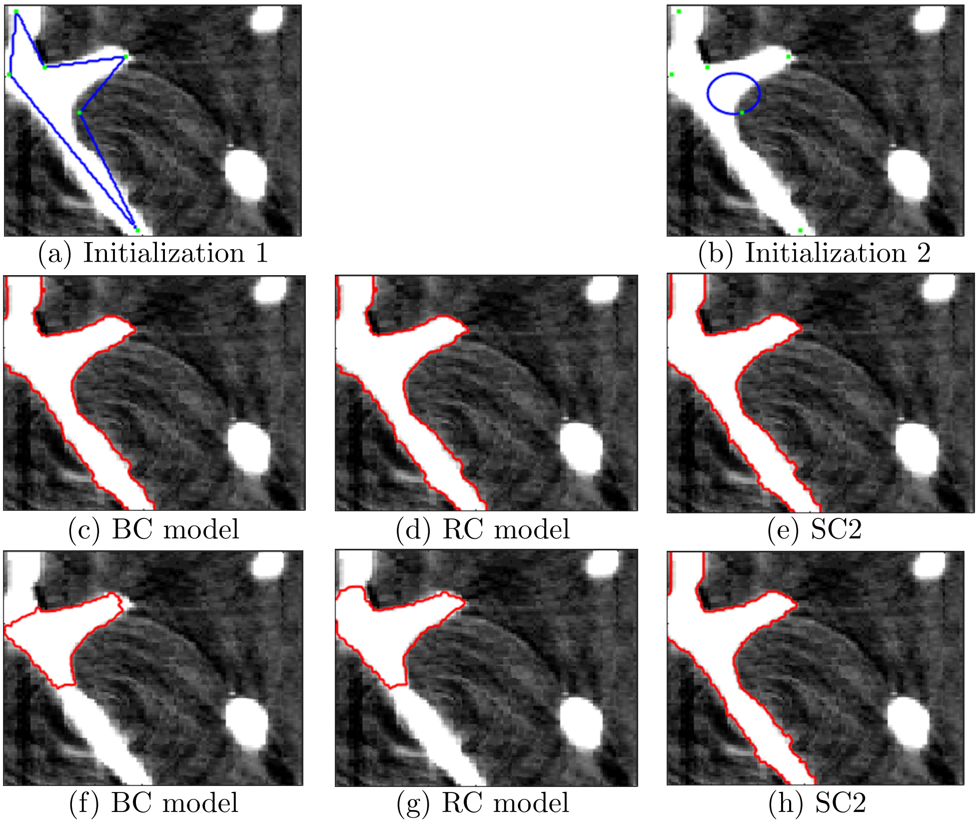
are $\bar{\mu} = 0.01$, $\beta = 10^{-4}$, $\rho = 10^{-3}$, $\theta = 2000$ (Figure 10(a)),$\theta = 400$ (Figure 10(d)), $\gamma = 10$, $tol = 10^{-2}$ and $maxit = 10^4$.

Zoomed segmentation results in Figure 10(b) and (e) demonstrate the limitation of NCZZ model [41]. As comparison, our SC2 gives cleaner segmentation as illustrated in Figure 10(c) and (f) for the same problems.

7.5 Test Set 5: Comparison of SC2 with BC [7] and RC [47]

Finally, we compare the performance of SC2 with two non-convex models namely BC model [7] and RC model [47] for different initializations in segmenting Problem 3. We set $\bar{\mu} = 128 \times 128 \times 0.05$, $\beta = 10^{-4}$, $\rho = 10^{-4}$, $\theta = 1000$, $\gamma = 5$, $tol = 10^{-4}$ and $maxit = 10^4$. Figures 11(a) and 11(b) show two different initializations with fixed markers.

The second row shows the results for all three models using the first initialization in (a) and the third row using the second initialization in (b). It can be seen that under different initializations, our SC2 will result in the same, consistent segmentation curves (hence independent of initializations) showing the advantage of a convex model. However, the segmentation results for BC and RC models are heavily dependent on the initialization; a well known drawback of non-



(a) Initialization 1 (b) Initialization 2

(c) BC model (d) RC model (e) SC2

(f) BC model (g) RC model (h) SC2

Figure 11: Test Set 5 – Performance comparison of BC, RC and SC2 models using 2 different initializations. With Initialization 1 in (a), the segmentation results for BC, RC, and SC2 models are illustrated on second row (c-e) respectively. With Initialization 2 in (b), the results are shown on third row (f-h). Clearly, SC2 gives a consistent segmentation result indicating that our SC2 is independent of initializations while BC and RC are sensitive to initializations due to different results obtained.

convex models. In addition, the segmentation result of non-convex models is not guaranteed to be a global solution.

8 Conclusions

In this work, we present a new primal-dual formulation for CDSS model [52] and propose an optimization based multilevel algorithm SC2 to solve the new formulation. In order to get a stronger decaying property than SC2, a new variant of SC2 named as SC2M is proposed. We also have developed a multilevel algorithm for the original CDSS model [52] called as SC1.

Five sets of tests are presented to compare eight models. In **Test Set 1** of the experiment, we find that all the multilevel algorithms have the expected optimal complexity $O(N \log N)$. However, SC2 converges faster than SC1 and SC2M. In addition, for all tested images, SC2 gives high accuracy compared to SC1 and SC2M. Practically, we recommend SC2 as the better multilevel algorithm for convex and selective segmentation method. In **Test Set 2**, we have performed the speed and quality comparisons of SC2 with SC0. Results show that SC2 performs much faster than SC0. Both algorithms deliver same high quality for the tested problem. We also have run the sensitivity test for our recommended algorithm SC2 towards parameters μ and θ. Comparison of SC2 with SC0 shows that SC2 is less sensitive to the regularization parameters μ and θ. Moreover, SC2 is also less sensitive for parameter β. In **Test Set 3**, we compare the segmentation quality of SC2 with the recent model CMT. The result demonstrates that SC2 performs better than CMT even for few markers. Moreover, the range of threshold values that work for SC2 is wider than CMT. In **Test Set 4**, the segmentation quality of SC2 is compared with NCZZ model. For the tested problem, it is clear that SC2 has successfully reduced the difficulty of NCZZ model that is unable to segment semi-transparent boundaries and sophisticated shapes. The final **Test Set 5** demonstrates the advantage of SC2 being a convex model (independent of initializations) compared to two non-convex models (BC and RC).

In future work, we will extend SC2 to 3D formulation and develop an optimization based multilevel approach for higher order selective segmentation models.

Acknowledgements

The first author would like to thank to Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Shah Alam 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 for the grant EP/N014499/1.

References

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

[2] G. Aubert and P. Kornprobst. Mathematical problems in image processing: Partial differential Equations and the Calculus of Variations. Springer, 2001.

[3] J. F. Aujol, and A. Chambolle. Dual norms and image decomposition models. International Journal of Computer Vision, 63(1), pp:85–104, 2005.

[4] J. F. Aujol, G. Gilboa, T. F. Chan, and S. Osher. Structure-texture image decomposition-modelling, algorithms, and parameter selection. International Journal of Computer Vision, 67(1), pp:111–136, 2006.

[5] N. Badshah and K. Chen. Multigrid method for the Chan-Vese model in variational segmentation. Commun. Comput. Phys., 4(2), pp:294–316, 2008.

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

[7] 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.

[8] E. Bae, X.C. Tai and W. Zhu. Augmented lagrangian method for an euler's elastica based segmentation model that promotes convex contours. Inverse Problems adn Imaging, 11(1), pp:1–23, 2017.

[9] 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. Iomed. 1(2), pp:161–173, 1993.

[10] S. Beucher. Segmentation Tools in Mathematical Morphology. SPIE 1350, Image Algebra and Morphological Image Processing, pp:70–84, 1990.

[11] X. Bresson, S. Esedoglu, P. Vandergheynst, J. P. thiran, and S. Osher. Fast global minimization of the active contour/snake model. Journal of Mathematical Imaging Vision, 28, pp:151–167, 2007.

[12] X. Cai, R. Chan, and T. Zeng. A two-stage image segmentation method using a convex variant of the Mumford-Shah model and thresholding. SIAM Journal of Imaging Sciences, 6, pp:368–390, 2013.

[13] J. Carter. Dual methods for total variation-based image restoration. PhD thesis, UCLA,2001.

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

[15] A. Chambolle. An algorithm for total variation minimization and applications. Journal of Mathematical Imaging and Vision, 20(1-2), pp:89–97, 2004.

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

[17] 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.

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

[19] T.F. Chan, S. Esedoglu, and M. Nikilova. Algorithm for finding global minimizers of image segmentation and denoising models. SIAM Journal on Applied Mathematics, 66, pp:1932–1648, 2006.

[20] T.F. Chan, G.Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. SIAM Journal on Scientific Computing, 20(6), pp:1964–1977, 1999.

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

[22] D. Chen, M. Yang, and L. D. Cohen. Global minimum for a variant mumford-shah model with application to medical image segmentation. Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization,1(1), pp:48–60, 2013.

[23] 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.

[24] M. Comer, C. Bouman, and J. Simmons. Statistical Methods for Image Segmentation and Tomography Reconstruction. Microscopy and Microanalysis, 16(S2), pp:1852-1853, 2010.

[25] T. Dietenbeck, M. Alessandrini, D. Friboulet, O. Bernard. CREASEG: a free software for the evaluation of image segmentation algorithms based on level-set. In IEEE International Conference On Image Processing, Hong Kong, China, pp:665–668, 2010.

[26] J. Duan, W. Lu, Z. Pan and L. Bai. New second order Mumford-Shah model based on Γ-convergence approximation for image processing. Infrared Physics and Technology, 76, pp:641–647, 2016.

[27] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 6(6), pp:721–741, 1984.

[28] 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.

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

[30] 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.

[31] S. Hanov. Wavelets and edge detection (preprint).stevehanov.ca/cs698_ wavelet, April 2006.

[32] A. K. Jumaat and K. Chen. An optimization-based multilevel algorithm for variational image segmentation models. Electronic Transactions on Numerical Analysis, 46, pp: 474-504, 2017.

[33] M. Kass, M. Witkin, and D. Terzopoulos.Snakes: Active Contour Models. International Journal of Vision, 1, pp:321–331, 1987.

[34] 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.

[35] C. Liu, M. K. P. Ng, and T. Zeng. Weighted variational model for selective image segmentation with application to medical images. Pattern Recognition, 16, pp:367–379, 2018.

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

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

[38] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Proc. 8th Int. Conf. Computer Vision, pp: 416–423, 2001.

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

[40] J. C. Moreno, V. B. S. Prasath, H. Proenca, and K. Palaniappan. Fast and globally convex multiphase active contours for brain MRI. Computer Vision and Image Undrstanding, 125, pp:237–250, 2014.

[41] T. N. A. Nguyen, J. Cai, J. Zhang, and J. Zheng. Robust interactive image segmentation using convex active contours. IEEE Trans. on Image Processing, 21(8), pp:3734–3743, 2012.

[42] 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.

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

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

[45] 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.

[46] G. Papandreou and P. Maragos. Multigrid geometric active contour models. IEEE Transactions on Image Processing, 16(1), pp: 229–240, Jan. 2007.

[47] L. Rada and K. Chen. Improved Selective Segmentation Model using One Level set. Journal of Algorithm & Computational Technology, 7(4), pp: 509-541, 2013.

[48] 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.

[49] J. A. Sethian. Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. Cambridge University Press, 1999.

[50] A. I. Shihab. Fuzzy Clustering Algorithms and Their Application to Medical Image Analysis. PhD dissertation, Dept. of Computing, Univ. of London, 2000.

[51] J. Spencer. Variational methods for image segmentation. PhD thesis, University of Liverpool, 2016.

[52] J. Spencer and K. Chen. A convex and selective variational model for image segmentation. Communication in Mathematical Sciences, 13(6), pp: 1453–1472, 2015.

[53] J. S. Weszka. A Survey of Threshold Selection Techniques. Computer Graphics and Image Proc. 7, pp: 259–265, 1978.

[54] 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.