A fast multilevel method for selective segmentation model of 3-D digital images

Abdul Kadir Jumaat[1,2] and Ke Chen[3]

[1]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, 40450, Shah Alam, Selangor, Malaysia.
[2]Institute for Big Data Analytics and Artificial Intelligence (IBDAAI), Universiti Teknologi MARA, 40450, Shah Alam, Selangor, Malaysia.
[3]Center for Mathematical Imaging Techniques, Department of Mathematical Sciences, University of Liverpool, United Kingdom.

E-mail: abdulkadir@tmsk.uitm.edu.my[1,2], k.chen@liverpool.ac.uk[3]

Abstract

Selective segmentation is capable of extracting an object or region in a given image. Recently, a 3-D convex variational selective segmentation model has been proposed and solved using projection algorithm. For moderate size of image, the projection algorithm method is effective. However, as the image size is larger, a fast iterative solver need to be develop. This research first proposed a fast optimization based multilevel algorithm in 3-D formulation that has the optimal complexity to solve the recent 3-D selective segmentation model. To achieve faster convergence, we reformulated the recent 3-D model into a new localized model. The proposed multilevel algorithm was used to solve the new localized model. MATLAB coding was developed to implement the segmentation process. The accuracy of the segmented image was evaluated using the Jaccard similarity measure. The execution time was recorded to measure the efficiency of the models. Test results demonstrated that our model is capable of successfully segmenting a targeted object in optimal computational time. Advantages of our algorithm can be seen in processing large size of 3-D images, where magnitudes of speed-up are observed over competing algorithms.

2010 Mathematics Subject Classification. **65N22**. 65N55, 62H35, 74G65, 74G75.
Keywords. selective image segmentation, localized 3-D model, level sets, fast multilevel algorithm, optimization methods.

1 Introduction

Image segmentation is vital in imaging and computer vision applications such as object detection, image classification and medical imaging [32, 19, 18, 17, 30, 31]. The topic remains challenging because the variety of images emerged from new modalities is vast and the quality of such images varies. Given an image, often of a fairly large size, two main issues are: i) how to design or select suitable models for segmenting images to extract interested or meaningful features in an automatic way; ii) how to solve a mathematical model in a fast way. The second issue of fast solution is particularly pressing for images in three spatial dimensions (3-D) because of the huge number of voxels to be processed. This research is concerned with both of these issues in the context of designing and solving a variational model.

There exist many segmentation models for images in two spatial dimensions (2-D), falling into categories of either edge based models [20] and domain based [23]. The former category focuses on edges and its effectiveness is much affected by noise, while the latter focuses on regions with intensity similarity and it searches for edges of domains if they are present. It is not difficult to

Advanced Studies: Euro-Tbilisi Mathematical Journal Special Issue (10 - 2022), pp. 127–152.

Tbilisi Centre for Mathematical Sciences.
Received by the editors: 26 May 2021.
Accepted for publication: 28 October 2021.

see that images with low contrast or weak edges pose a challenge to both categories of models. Often a good model can merge ideas of both categories in terms of weighted edges using domain information or partitioning domains regularized with edge information. Although most of 2-D based active contours models and methods may be generalized to 3-D active surfaces [10, 29, 34], not all models have been generalised to or studied in 3-D. Incorporating extra information such priors or seed points can improve the robustness of a model. In fact, many application fields demand segmentation of a specific feature or region defined by a few geometric markers. This is the case that we consider in this research in the so-called selective segmentation.

One of the earliest 2-D selective segmentation model is [2] that is improved and extended to 3-D in [27, 33]. As with most models based on level set formulation, such models [27, 33] are non-convex which result to highly dependent on the initial solution to capture a local minimizer.

Convexification idea by [8] is widely used in the community to solve the issue of sensitive to initialization. Two models followed this idea. First, [28] proposed such a 2-D convex selective model termed CDSS. The CDSS model, though capable of capturing a global minimizer, is sensitive to area and regularization parameters. Thus, a reformulation of the model termed Primal-Dual Selective Segmentation (PDSS) is introduced in [15] and [14] to overcome the parameter sensitivity issue. Second, the convex model by Liu *et al.* [21] was proposed for selective segmentation. It is based on [2, 4, 26]. Unfortunately as reported in [15], this model requires a lot of geometric markers which is not reliable for selective segmentation modeling and the model is found to be sensitive to the threshold value in comparison to PDSS model of [15].

The main ingredient in formulating the above selective segmentation models is geometric constraints, introduce by [11, 12] which are partial differential equations (PDEs) based models for segmentation. We remark that the geometric constraints can be defined in different way as reported in [24] where the authors utilized two different markers i.e., outside and inside a targeted region. The model, though successfuly in extracting an object selectively, is not effective for fine and sophisticated images [24]. Comparison between [24] and PDSS suggests that PDSS model is more successful; see [15]. Recently, a 3-D formulation for PDSS model termed 3-D PDSS has been introduced in [16], solved by using projection algorithm that was effective for image with moderate size.

However, in order to segment images of large size especially for 3-D data, there is need in developing fast iterative solver for 3-D selective image segmentation. Therefore, this research proposes an optimization based multilevel algorithm in 3-D formulation to solve the 3-D PDSS model. The multilevel algorithm is based on discretize-optimize scheme (algebraic) where the minimization of a variational functional is solved directly without deriving the Euler Lagrange equation [13]. To increase the segmentation efficiency, this research proposes a modified 3-D PDSS model based on the local property of level set function. An associate multilevel algorithm is proposed to solve the modified 3-D PDSS model.

This research paper is structured as follows. In Section 2, we will review some 3-D selective segmentation models. We demonstrate on how to develop a 3-D multilevel method based on [13] to obtain the solution for the recent 3-D PDSS model [16] in Section 3. To achieve faster convergence, a new localized version of the 3-D PDSS model will be proposed with its fast multilevel algorithm in Section 4. In Section 5, an extended version of the algorithm will be formulated and its convergence will be discussed in Section 6. The experimental results are demonstrated in Section 7 and the conclusion of the research will be presented in Section 8.

2 Review of the related models

The extension of Chan and Vese's 2-D model [9] into 3-D version model [34, 27] will be reviewed. Following that, we will look at Jumaat and Chen's recent 3-D selective segmentation model (3-D PDSS) [16].

2.1 The 3-D Chan and Vese (3-D CV) model

Let $\mathbf{x} = (x, y, z)$ and a given 3-D grayscale image defined as $I(\mathbf{x}) : D \subset \mathbb{R}^3 \to \mathbb{R}$. In the 3-D CV model [9], it is assumed that the 3-D image $I(\mathbf{x})$ is separated by 2-D surface S into two 3-D regions of distinct intensity values a_1 and a_2. The targeted object is represented by the 3-D region D_1 with the value a_1 inside the 2-D surface S. Outside S, that is in $D_2 = D \backslash D_1$, the intensity of $I(\mathbf{x})$ is approximated by a_2. As $D = D_1 \cup D_2$, the 3-D CV model is defined by the following Equation (2.1):

$$\begin{aligned} \min_{S,a_1,a_2} CV^{3D}(S, a_1, a_2) = \mu \text{ length}(S) + \lambda_1 \int_{D_1} (I(\mathbf{x}) - a_1)^2 d\mathbf{x} \\ + \lambda_2 \int_{D_2} (I(\mathbf{x}) - a_2)^2 d\mathbf{x}. \end{aligned} \tag{2.1}$$

The constants a_1 and a_2 can be defined as the mean intensities values of $I(\mathbf{x})$ inside and outside the surface S respectively. Here, the non-negative parameters μ, λ_1, and λ_2 are introduced as weight parameters to control the length term and the fitting term respectively . The level set method [9] is applied to minimize Equation (2.1) where the zero level set of the Lipschitz function $\varphi(x, y, z)$ is used to represent the surface S such that

$$\begin{aligned} &S = \{(x, y, z) \in D : \varphi(x, y, z) = 0\}, \\ &D_1 = \text{in}(S) = \{(x, y, z) \in D : \varphi(x, y, z) > 0\}, \\ &D_2 = \text{out}(S) = \{(x, y, z) \in D : \varphi(x, y, z) < 0\}. \end{aligned}$$

Next, the regularized versions of the Heaviside function and the Dirac delta function are introduced, respectively, as

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

Now, the Equation (2.1) can be written as the following Equation (2.2):

$$\begin{aligned} &\min_{\varphi,a_1,a_2} CV^{3D}(\varphi, a_1, a_2) = \mu \int_D |\nabla H(\varphi)| \, d\mathbf{x} + \lambda_1 \int_D (I(\mathbf{x}) - a_1)^2 H(\varphi) \, d\mathbf{x} \\ &+ \lambda_2 \int_D (I(\mathbf{x}) - a_2)^2 (1 - H(\varphi)) \, d\mathbf{x}. \end{aligned} \tag{2.2}$$

The following constants a_1 and a_2 can be obtained by fixing φ and minimizing (2.2) with respect to both constants:

$$a_1(\varphi) = \frac{\int_\Omega I(\mathbf{x}) H(\varphi) \, d\mathbf{x}}{\int_\Omega H(\varphi) \, d\mathbf{x}}, \qquad a_2(\varphi) = \frac{\int_\Omega I(\mathbf{x}) (1 - H(\varphi)) \, d\mathbf{x}}{\int_\Omega (1 - H(\varphi)) \, d\mathbf{x}}. \tag{2.3}$$

By fixing constants a_1 and a_2 in $CV^{3D}(\varphi, c_1, c_2)$, first variation with respect to φ can be computed and we arrived to the following Euler-Lagrange partial differential equation (EL-PDE):

$$\begin{cases} \mu\delta(\varphi)\nabla\cdot\left(\frac{\nabla\varphi}{|\nabla\varphi|}\right) - \lambda_1\delta(\varphi)(I-a_1)^2 + \lambda_2\delta(\varphi)(I-a_2)^2 = 0, \quad in \quad D \\ \qquad\qquad\qquad\qquad\qquad\qquad \frac{\delta(\varphi)}{|\nabla\varphi|}\frac{\partial u}{\partial\vec{n}} = 0, \quad on \quad \partial D. \end{cases} \tag{2.4}$$

To avoid singularity problem due to the total variation term $|\nabla\varphi| = 0$, a constant $\beta > 0$ was introduce to (2.2) and (2.4), so the new EL-PDE becomes

$$\begin{cases} \mu\delta(\varphi)\nabla\cdot\left(\frac{\nabla\varphi}{\sqrt{|\nabla\varphi|^2+\beta}}\right) - \lambda_1\delta(\varphi)(I-a_1)^2 + \lambda_2\delta(\varphi)(I-a_2)^2 = 0, \quad in \quad D \\ \qquad\qquad\qquad\qquad\qquad\qquad \frac{\delta(\varphi)}{|\nabla\varphi|}\frac{\partial u}{\partial\vec{n}} = 0, \quad on \quad \partial D. \end{cases}$$

This is corresponding to the following functional

$$\begin{aligned} &\min_{\varphi,a_1,a_2} CV^{3D}(\varphi, a_1, a_2) = \mu\int_\Omega \sqrt{|\nabla H(\varphi)|^2 + \beta}\, d\mathbf{x} \\ &+\lambda_1\int_D (I-a_1)^2 H(\varphi)\, d\mathbf{x} + \lambda_2\int_D (I-a_2)^2 (1-H(\varphi))\, d\mathbf{x}. \end{aligned} \tag{2.5}$$

2.2 The 3-D Zhang and Chen (3-D ZC) model

The 3-D variational selective segmentation formulation by Zhang and Chen (3-D ZC) [33] improves the model by [2]. The 3-D ZC model combines the edge based model of [11, 12] with the intensity fitting term of 3-D CV [10] in localized formulation. The selective segmentation idea aims to extract the boundary of a single target object among all homogeneity intensity object in a given image that are defined in a closed domain to the geometrical points in a set $\mathrm{B} = \{V_i = (x_i^*, y_i^*, z_i^*) \in D, 1 \le i \le n_1\} \subset D$. Here, the points $n_1\,(\ge 3)$ are placed near the target object [2]. Hence, the 3D-ZC model is defined as the following Equation (2.6):

$$\min_{S,a_1,a_2}\{F(S, a_1, a_2) = \mu F_G(S) + F_H(S, a_1, a_2)\} \tag{2.6}$$

where $F_G(S) = \int_S G(\mathbf{x})\, ds$ with $G(\mathbf{x}) = g(\mathbf{x})\, d(\mathbf{x})$ and $g(\mathbf{x}) = \frac{1}{1+q|\nabla I(\mathbf{x})|^2}$ is an edge detector function, weighted by a constant q. The function $d(\mathbf{x})$ is a marker distance function, defined as follows:

$$d(\mathbf{x}) = \text{distance}((\mathbf{x}), \mathrm{B}) = \prod_{i=1}^{n_1}\left(1 - e^{-\frac{(\mathbf{x}-^*_i)^2}{2\kappa^2}}\right), \quad \forall(\mathbf{x}) \in D$$

.

The new localized fitting energy $F_H(S, a_1, a_2)$ is given as

$$\left\{\lambda_1\int_{D_{in}(S)} b_1(\varphi(\mathbf{x}), \gamma_{in})(I(\mathbf{x})-a_1)^2 d\mathbf{x} + \lambda_2\int_{D_{out}(S)} b_2(\varphi(\mathbf{x}), \gamma_{out})(I(\mathbf{x})-a_2)^2 d\mathbf{x}\right\}.$$

The term $b_1(\varphi(\mathbf{x}), \gamma_{in}) = B(\varphi(\mathbf{x}), \gamma_{in}, 0)$ and $b_2(\varphi(\mathbf{x}), \gamma_{out}) = B(\varphi(\mathbf{x}), 0, \gamma_{out})$ with

$$B\left(\varphi\left(\mathbf{x}\right),\gamma_{in},\gamma_{out}\right)=H\left(\varphi\left(\mathbf{x}\right)+\gamma_{in}\right)\left(1-H\left(\varphi\left(\mathbf{x}\right)-\gamma_{out}\right)\right) \tag{2.7}$$

used to characterize the domain $D_{\gamma_{in},\gamma_{out}}=\left\{\mathbf{x}\in D:-\gamma_{in}\leq\varphi\left(\mathbf{x}\right)\leq\gamma_{out}\right\}=D_{\gamma_{in}}\left(S\right)\cup S\cup D_{\gamma_{out}}\left(S\right)$ which is a narrow band region surrounding the local boundary S. The value of φ is assumed to be negative inside S and positive outside S. In level set representation, the 3-D ZC model is given by the following equation:

$$\begin{aligned}&\min_{\varphi,a_1,a_2}\mu\int_D G\left(\mathbf{x}\right)\left|\nabla H\left(\varphi\right)\right|d\mathbf{x}+\lambda_1\int_D\left(I-a_1\right)^2b_1\left(\varphi\left(\mathbf{x}\right),\gamma_{in}\right)\left(1-H\left(\varphi\right)\right)\,d\mathbf{x}\\&+\lambda_2\int_D\left(I-a_2\right)^2b_2\left(\varphi\left(\mathbf{x}\right),\gamma_{out}\right)H\left(\varphi\right)\,d\mathbf{x}\end{aligned} \tag{2.8}$$

Here, $b_1\left(\varphi,\gamma_{in}\right)\left(1-H\right)=B\left(\varphi\left(\mathbf{x}\right),\gamma_{in},0\right)$ and $b_2\left(\varphi,\gamma_{out}\right)\left(H\right)=B\left(\varphi\left(\mathbf{x}\right),0,\gamma_{out}\right)$. The 3-D ZC model was solved by a multigrid method.

2.3 The 3-D Primal Dual Selective Segmentation (3-D PDSS) model

In 2020, Jumaat and Chen [16] proposed a new 3-D Primal Dual Selective Segmentation (3-D PDSS) model which was the extension of the 2-D model in [15].

Using the marker set B, a polygon Q was constructed to connects up the markers. The 3-D PDSS model applied the Euclidean distance function $C\left(\mathbf{x}\right)$ of each point $\left(\mathbf{x}\right)\in D$ from its nearest point $\left(x_q,y_q,z_q\right)\in Q$:

$$C\left(\mathbf{x}\right)=\sqrt{\left(x-x_q\right)^2+\left(y-y_q\right)^2+\left(z-z_q\right)^2}=\min_{q\in Q}\left\|\left(x,y,z\right)-\left(x_q,y_q,z_q\right)\right\|.$$

Other distance functions in [1] are possible to consider. Thus, the 3-D PDSS model is given as the following Equation (2.9):

$$\min_{u,w\in[0,1]}PDSS^{3D}\left(u,w\right)=\int_D\left|\nabla u\right|_g d\mathbf{x}+\int_D rw\,d\mathbf{x}+\theta\int_D Cw\,d\mathbf{x}+\frac{1}{2\rho}\int_D\left(u-w\right)^2\,d\mathbf{x}. \tag{2.9}$$

Here, $r=\left(a_1-I\right)^2-\left(a_2-I\right)^2$ and $\left|\nabla u\right|_g=g\left(\left|\nabla z\right|\right)\left|\nabla u\right|$. The dual variable, w was introduced. For sufficiently small $\rho>0$, the right-most term enforces $w\approx u$. The parameter θ was the weight of the distance fitting term. If the parameter θ is too strong the final result will just be the polygon Q while the final result is potentially includes the neighboring object if it is too weak.

Following the idea from [6, 3], a unique minimizer of 3-D PDSS model can be computed using the alternating minimization method. Therefore, the formulation of Equation 2.9 was equivalent to solve the following functional alternately until convergence:

$$\min_u\;M_1\left(u,w\right)=\int_D\left|\nabla u\right|_g d\mathbf{x}+\frac{1}{2\rho}\int_D\left(u-w\right)^2 d\mathbf{x}, \tag{2.10}$$

$$\min_{w\in[0,1]}\;M_2\left(u,w\right)=\int_D rw\,d\mathbf{x}+\theta\int_D Cw\,d\mathbf{x}+\frac{1}{2\rho}\int_D\left(u-w\right)^2 d\mathbf{x}. \tag{2.11}$$

A projection algorithm was used to solve 2.10 while the solution of 2.11 can be found analytically, defined as

$$w=\min\left\{\max\left\{u(\mathbf{x})-\rho r-\rho\theta C,0\right\},1\right\}. \tag{2.12}$$

3 A 3-D optimization based multilevel algorithm

In this section, a 3-D multilevel algorithm is discussed to solve (2.10). We assume $n = 2^L$ for a given image I of size $N = n \times n \times n$. The standard coarsening defines $L+1$ levels: $s = 1$ (finest) $, 2, ..., L,\ L+1$ (coarsest) such that level s has $\tau_s \times \tau_s \times \tau_s$ "superpixels" with each "superpixels" having pixels $b_s \times b_s \times b_s$ where $\tau_s = n/2^{s-1}$ and $b_s = 2^{s-1}$. If $n \neq 2^L$, the multilevel method can still be developed with some coarse level superpixels of cube shapes and the rest of cuboid shapes.

3.1 3-D optimization based multilevel algorithm

The multilevel algorithm is developed only to solve (2.10) in discretize-optimize scheme as the solution of (2.11) is analytically defined by the equation 2.12. By refering the related works in [5, 7], (2.10) is discretized using the standard finite difference method with Neumann's boundary condition as follows

$$\begin{aligned}
&\min_u M_1(u) \equiv \min_u M_1^a(u_{1,1,1}, u_{2,1,1}, ..., u_{i-1,j,k}, u_{i,j,k}, u_{i+1,j,k}, ..., u_{n,n,n}) \\
&= \sum_{i=1}^{n-1}\sum_{j=1}^{n-1}\sum_{k=1}^{n-1} g_{i,j,k}\sqrt{(u_{i,j,k}-u_{i,j+1,k})^2 + (u_{i,j,k}-u_{i+1,j,k})^2 + (u_{i,j,k}-u_{i,j,k+1})^2 + \beta} \\
&+ \frac{1}{2\rho}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}(u_{i,j,k}-w_{i,j,k})^2.
\end{aligned} \tag{3.1}$$

Using a coordinate descent method, (3.1) is minimized on the finest level 1:

$$\begin{cases}
\text{Given} & u^{(m)} = \left(u_{i,j,k}^{(m)}\right) \ \text{ with } m = 0, \\
\text{Solve} & u_{i,j,k}^{(m+1)} = \underset{u_{i,j,k}\in\mathbb{R}}{\arg\min}\, M_1^{loc}(u_{i,j,k}, a_1, a_2) \ \text{ for } i,j,k = 1,2,...n, \\
& \text{Repeat the above steps with } m = m+1 \text{ until stopped.}
\end{cases} \tag{3.2}$$

where

$$\begin{aligned}
&M_1^{loc}(u_{i,j,k}) \\
&\equiv M_1^{loc}\left(u_{1,1,1}^{(m-1)}, u_{2,1,1}^{(m-1)}, ..., u_{i-1,j,k}^{(m-1)}, u_{i,j,k}, u_{i+1,j,k}^{(m-1)}, ..., u_{n,n,n}^{(m-1)}\right) - M_1^{(m-1)} \\
&= g_{i,j,k}\sqrt{\left(u_{i,j,k}-u_{i+1,j,k}^{(m)}\right)^2 + \left(u_{i,j,k}-u_{i,j+1,k}^{(m)}\right)^2 + \left(u_{i,j,k}-u_{i,j,k+1}^{(m)}\right)^2 + \beta} \\
&+ g_{i-1,j,k}\sqrt{\left(u_{i,j,k}-u_{i-1,j,k}^{(m)}\right)^2 + \left(u_{i-1,j,k}^{(m)}-u_{i-1,j+1,k}^{(m)}\right)^2 + \left(u_{i-1,j,k}^{(m)}-u_{i-1,j,k+1}^{(m)}\right)^2 + \beta} \\
&+ g_{i,j-1,k}\sqrt{\left(u_{i,j,k}-u_{i,j-1,k}^{(m)}\right)^2 + \left(u_{i,j-1,k}^{(m)}-u_{i+1,j-1,k}^{(m)}\right)^2 + \left(u_{i,j-1,k}^{(m)}-u_{i,j-1,k+1}^{(m)}\right)^2 + \beta} \\
&+ g_{i,j,k-1}\sqrt{\left(u_{i,j,k}-u_{i,j,k-1}^{(m)}\right)^2 + \left(u_{i,j,k-1}^{(m)}-u_{i+1,j,k-1}^{(m)}\right)^2 + \left(u_{i,j,k-1}^{(m)}-u_{i,j+1,k-1}^{(m)}\right)^2 + \beta} \\
&+ \tfrac{1}{2\rho}(u_{i,j,k}-w_{i,j,k})^2.
\end{aligned}$$

The last term $M_1^{(m-1)}$ refers to the sum of all terms in M_1^a that do not involve $u_{i,j,k}$. We applying the Newton method to solve the one-dimensional problem from (3.2) by iterating $u^{(m)} \to u \to u^{(m+1)}$.

It is of interest to rewrite (3.2) as the following form in order to introduce the multilevel algorithm:

$$\begin{cases} \text{Given} & u^{(m)} = \left(u_{i,j,k}^{(m)}\right) \quad \text{with } m = 0, \\ \text{Solve} & \hat{c} = \arg\min_{c \in \mathbb{R}} M_1^{loc}\left(u_{i,j,k}^{(m)} + c\right), \\ \text{Update} & u_{i,j,k}^{(m+1)} = u_{i,j,k}^{(m)} + \hat{c}, \\ & \text{Repeat the above steps with } m = m+1 \text{ until stopped.} \end{cases} \tag{3.3}$$

The problem (3.3) can be viewed as finding the best correction constant $\hat{c}$ at the current approximation $u_{i,j,k}^{(m)}$ on level 1.

Next, the formulation for each of the subproblems associated with these blocks on level s is derived. We illustrate the method for level 3 of image size $16 \times 16 \times 16$ as shown in Figure 1. Figure 1(a) demonstrates one of $\tau_3^3 = 4^3$ superpixel in level 3. Each superpixel contains $b_3^3 = 4^3$ pixels. (b) shows the top surface of (a). Using equation (3.1), the interaction of a pixel with neighbouring pixel (red •) is demonstrated in (c). The interaction of pixels in (b) based on equation (3.1) and (c) is shown in (d).

To proceed, we define the following: $b = 2^{s-1}$, $k_1 = (i-1)\,b + 1$, $k_2 = ib$, $\ell_1 = (j-1)\,b + 1$, $\ell_2 = jb$, $m_1 = (k-1)\,b + 1$, $m_2 = kb$, and $c = (c_{i,j,k})$. Let $\tilde{u}$ be the current solution. Next, a general computational stencil involving c on level s can be illustrated as follows

(3.4)

Then, minimizing for c, the problem 3.3 is equivalent to minimize the following

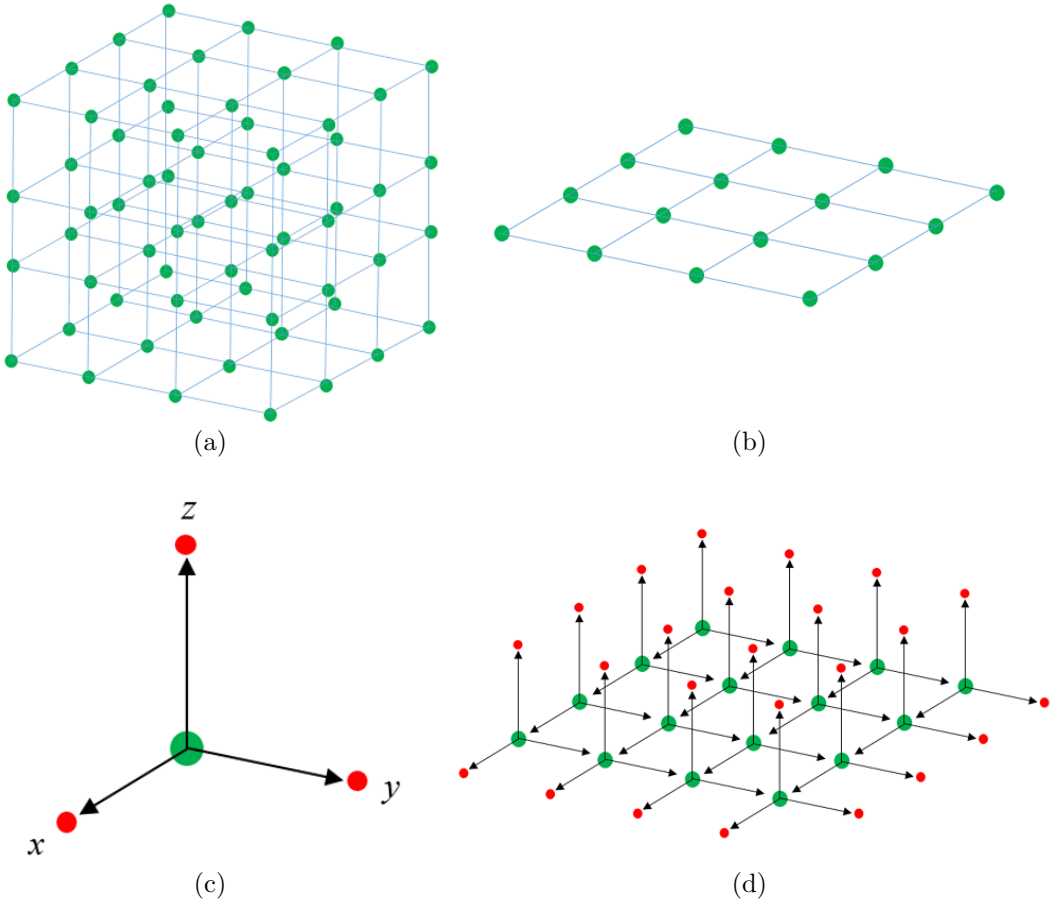


FIGURE 1. Illustration of level 3 for image size $16 \times 16 \times 16$. (a) demonstrates one of $\tau_3^3 = 4^3$ superpixel in level 3. Each superpixel contains $b_3^3 = 4^3$ pixels. (b) shows the top surface of (a). Using equation (3.1), the interaction of a pixel with neighbouring pixel (red •) is demonstrated in (c). The interaction of pixels in (b) based on equation (3.1) and (c) is shown in (d).

$$
\begin{aligned}
F_{3-DPDSS}(c_{i,j,k}) = &\sum_{k=k_1}^{k_2-1} g_{k,\ell_2,m_1}\sqrt{T_1+\beta} + \sum_{k=\ell_1}^{\ell_2-1} g_{k_2,\ell,m_1}\sqrt{T_2+\beta} + g_{k_2,\ell_2,m_1}\sqrt{T_3+\beta} \\
&+ \sum_{\ell=\ell_1}^{\ell_2-1} g_{k_1,\ell,m_2}\sqrt{T_4+\beta} + \sum_{k=k_1}^{k_2-1} g_{k,\ell_2,m_2}\sqrt{T_5+\beta} + g_{k_2,\ell_2,m_2}\sqrt{T_6+\beta} \\
&+ \sum_{\ell=\ell_1}^{\ell_2-1} g_{k_2,\ell,m_2}\sqrt{T_7+\beta} + \sum_{k=k_1}^{k_2-1} g_{k,\ell_1,m_2}\sqrt{T_8+\beta} + \sum_{m=m_1+1}^{m_2-1} g_{k_1,\ell_2,m}\sqrt{T_9+\beta} \\
&+ \sum_{m=m_1+1}^{m_2-1} g_{k_2,\ell_2,m}\sqrt{T_{10}+\beta} + \sum_{m=m_1+1}^{m_2-1} g_{k_2,\ell_1,m}\sqrt{T_{11}+\beta} \\
&+ \sum_{k=k_1+1}^{k_2-1}\sum_{\ell=\ell_1+1}^{\ell_2-1} g_{k,\ell,m_2}\sqrt{T_{12}+\beta} + \sum_{\ell=\ell_1+1}^{\ell_2-1}\sum_{m=m_1+1}^{m_2-1} g_{k_2,\ell,m}\sqrt{T_{13}+\beta} \\
&+ \sum_{k=k_1+1}^{k_2-1}\sum_{m=m_1+1}^{m_2-1} g_{k,\ell_2,m}\sqrt{T_{14}+\beta} + \sum_{k=k_1}^{k_2}\sum_{\ell=\ell_1}^{\ell_2} g_{k,\ell,m_1-1}\sqrt{T_{15}+\beta} \\
&+ \sum_{\ell=\ell_1}^{\ell_2}\sum_{m=m_1}^{m_2} g_{k_1-1,\ell,m}\sqrt{T_{16}+\beta} + \sum_{k=k_1}^{k_2}\sum_{m=m_1}^{m_2} g_{k,\ell_1-1,m}\sqrt{T_{17}+\beta} \\
&+ \tfrac{1}{2\rho}\sum_{k=k_1}^{k_2}\sum_{\ell=\ell 1}^{\ell_2}\sum_{m=m_1}^{m_2} (u_{k,\ell,m} + c_{i,j,k} - w_{k,\ell,m})^2.
\end{aligned}
\tag{3.5}
$$

where

$$
\begin{aligned}
T_1 &= (\tilde{u}_{k,\ell_2,m_1} + c_{i,j,k} - \tilde{u}_{k,\ell_2+1,m_1})^2 + (\tilde{u}_{k,\ell_2,m_1} - \tilde{u}_{k+1,\ell_2,m_1})^2 + (\tilde{u}_{k,\ell_2,m_1} - \tilde{u}_{k,\ell_2,m_1+1})^2 \\
T_2 &= (\tilde{u}_{k_2,\ell,m_1} + c_{i,j,k} - \tilde{u}_{k_2+1,\ell,m_1})^2 + (\tilde{u}_{k_2,\ell,m_1} - \tilde{u}_{k_2,\ell+1,m_1})^2 + (\tilde{u}_{k_2,\ell,m_1} - \tilde{u}_{k_2,\ell,m_1+1})^2 \\
T_3 &= (\tilde{u}_{k_2,\ell_2,m_1} + c_{i,j,k} - \tilde{u}_{k_2+1,\ell_2,m_1})^2 + (\tilde{u}_{k_2,\ell_2,m_1} + c_{i,j,k} - \tilde{u}_{k_2,\ell_2+1,m_1})^2 \\
&+(\tilde{u}_{k_2,\ell_2,m_1} - \tilde{u}_{k_2,\ell_2,m_1+1})^2 \\
T_4 &= (\tilde{u}_{k_1,\ell,m_2} + c_{i,j,k} - \tilde{u}_{k_1,\ell,m_2+1})^2 + (\tilde{u}_{k_1,\ell,m_2} - \tilde{u}_{k_1+1,\ell,m_2})^2 + (\tilde{u}_{k_1,\ell,m_2} - \tilde{u}_{k_1,\ell+1,m_2})^2 \\
T_5 &= (\tilde{u}_{k,\ell_2,m_2} + c_{i,j,k} - \tilde{u}_{k,\ell_2+1,m_2})^2 + (\tilde{u}_{k,\ell_2,m_2} + c_{i,j,k} - \tilde{u}_{k,\ell_2,m_2+1})^2 + (\tilde{u}_{k,\ell_2,m_2} - \tilde{u}_{k+1,\ell_2,m_2})^2 \\
T_6 &= (\tilde{u}_{k_2,\ell_2,m_2} + c_{i,j,k} - \tilde{u}_{k_2+1,\ell_2,m_2})^2 + (\tilde{u}_{k_2,\ell_2,m_2} + c_{i,j,k} - \tilde{u}_{k_2,\ell_2+1,m_2})^2 \\
&+(\tilde{u}_{k_2,\ell_2,m_2} + c_{i,j,k} - \tilde{u}_{k_2,\ell_2,m_2+1})^2 \\
T_7 &= (\tilde{u}_{k_2,\ell,m_2} + c_{i,j,k} - \tilde{u}_{k_2,\ell,m_2+1})^2 + (\tilde{u}_{k_2,\ell,m_2} + c_{i,j,k} - \tilde{u}_{k_2+1,\ell,m_2})^2 + (\tilde{u}_{k_2,\ell,m_2} - \tilde{u}_{k_2,\ell+1,m_2})^2 \\
T_8 &= (\tilde{u}_{k,\ell_1,m_2} + c_{i,j,k} - \tilde{u}_{k,\ell_1,m_2+1})^2 + (\tilde{u}_{k,\ell_1,m_2} - \tilde{u}_{k,\ell_1+1,m_2})^2 + (\tilde{u}_{k,\ell_1,m_2} - \tilde{u}_{k+1,\ell_1,m_2})^2 \\
T_9 &= (\tilde{u}_{k_1,\ell_2,m} + c_{i,j,k} - \tilde{u}_{k_1,\ell_2+1,m})^2 + (\tilde{u}_{k_1,\ell_2,m} - \tilde{u}_{k_1,\ell_2,m+1})^2 + (\tilde{u}_{k_1,\ell_2,m} - \tilde{u}_{k_1+1,\ell_2,m})^2 \\
T_{10} &= (\tilde{u}_{k_2,\ell_2,m} + c_{i,j,k} - \tilde{u}_{k_2+1,\ell_2,m})^2 + (\tilde{u}_{k_2,\ell_2,m} + c_{i,j,k} - \tilde{u}_{k_2,\ell_2+1,m})^2 + (\tilde{u}_{k_2,\ell_2,m} - \tilde{u}_{k_2,\ell_2,m+1})^2 \\
T_{11} &= (\tilde{u}_{k_2,\ell_1,m} + c_{i,j,k} - \tilde{u}_{k_2+1,\ell_1,m})^2 + (\tilde{u}_{k_2,\ell_1,m} - \tilde{u}_{k_2,\ell_1+1,m})^2 + (\tilde{u}_{k_2,\ell_1,m} - \tilde{u}_{k_2,\ell_1,m+1})^2 \\
T_{12} &= (\tilde{u}_{k,\ell,m_2} + c_{i,j,k} - \tilde{u}_{k,\ell,m_2+1})^2 + (\tilde{u}_{k,\ell,m_2} - \tilde{u}_{k+1,\ell,m_2})^2 + (\tilde{u}_{k,\ell,m_2} - \tilde{u}_{k,\ell+1,m_2})^2 \\
T_{13} &= (\tilde{u}_{k_2,\ell,m} + c_{i,j,k} - \tilde{u}_{k_2+1,\ell,m})^2 + (\tilde{u}_{k_2,\ell,m} - \tilde{u}_{k_2,\ell+1,m})^2 + (\tilde{u}_{k_2,\ell,m} - \tilde{u}_{k_2,\ell,m+1})^2 \\
T_{14} &= (\tilde{u}_{k,\ell_2,m} + c_{i,j,k} - \tilde{u}_{k,\ell_2+1,m})^2 + (\tilde{u}_{k,\ell_2,m} - \tilde{u}_{k+1,\ell_2,m})^2 + (\tilde{u}_{k,\ell_2,m} - \tilde{u}_{k,\ell_2,m+1})^2 \\
T_{15} &= (\tilde{u}_{k,\ell,m_1} + c_{i,j,k} - \tilde{u}_{k,\ell,m_1-1})^2 + (\tilde{u}_{k,\ell,m_1-1} - \tilde{u}_{k+1,\ell,m_1-1})^2 + (\tilde{u}_{k,\ell,m_1-1} - \tilde{u}_{k,\ell+1,m_1-1})^2 \\
T_{16} &= (\tilde{u}_{k_1,\ell,m} + c_{i,j,k} - \tilde{u}_{k_1-1,\ell,m})^2 + (\tilde{u}_{k_1-1,\ell,m} - \tilde{u}_{k_1-1,\ell+1,m})^2 + (\tilde{u}_{k_1-1,\ell,m} - \tilde{u}_{k_1-1,\ell,m+1})^2 \\
T_{17} &= (\tilde{u}_{k,\ell_1,m} + c_{i,j,k} - \tilde{u}_{k,\ell_1-1+1,m})^2 + (\tilde{u}_{k,\ell_1-1,m} - \tilde{u}_{k,\ell_1-1,m+1})^2 + (\tilde{u}_{k,\ell_1-1,m} - \tilde{u}_{k+1,\ell_1-1,m})^2
\end{aligned}
$$

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

$$\min_{c} \{ F_{3-DPDSS}(\tilde{u}+c) = \frac{1}{2\rho} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} (u_{i,j,k} + c - w_{i,j,k})^2 \} \tag{3.6}$$

Algorithm 1 summarized the proposed method to solve the 3-D PDSS termed 3-D MPDSS.

Algorithm 1 3-D MPDSS – Multilevel Algorithm to solve the 3-D PDSS

Set $w = u$,

1) *Solve* (2.10) *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* (3.2) *for* $i, j, k = 1, 2, ...n$

   iii). *Iterate for* t *times on each coarse level* $s = 2, 3, ...L, L+1$ *:*
   $>$ *If* $s \leq L$, *compute the minimizer* c *of* (3.5)
   $>$ *Solve* (3.6) *on the coarsest level* $s = L+1$
   $>$ *Add the correction* $u = u + Q_s c$ *where* Q_s *is the interpolation operator distributing* $c_{i,j}$ *to the corresponding* $b_s \times b_s \times b_s$ *block on level* s *as illustrated in* (3.4).

2) *Solve* (2.11) *to update* w*:*

   i). *Set* $\tilde{w} = w$.

   ii). *Compute* w *using the formula* (2.12).

3) *Repeat the steps until the solution converges.*

4 A new 3-D localized model

To further speed up the convergence, we propose to compute the solution in a narrow band region. We first defined two local functions b_1 and b_2 that characterize the domain of narrow band region $D_\Gamma = D_1(\Gamma) \cup S \cup D_2(\Gamma)$ around the 2-D surface S such that

$$b_1(\varphi(\mathbf{x}), \Gamma) = 1 - H(\varphi(\mathbf{x}) - \Gamma)$$
$$b_2(\varphi(\mathbf{x}), \Gamma) = H(\varphi(\mathbf{x}) + \Gamma)$$

We remark that similar notation can be found in [22, 35, 33].

Then, the fitting term of the 3-D PDSS model is modified as

$$\int_{D_1} (I - a_1)^2 \, d\mathbf{x} \to \int_{D} (I - a_1)^2 b_1(\varphi, \Gamma) H(\varphi) \, d\mathbf{x}$$
$$\int_{D_2} (I - a_2)^2 \, d\mathbf{x} \to \int_{D} (I - a_2)^2 b_2(\varphi, \Gamma)(1 - H(\varphi)) \, d\mathbf{x}$$

By applying the relaxation of the Heaviside function to $u \in [0,1]$, the final solution $u(\mathbf{x})$ defining the targeted object by $\sum = \{(x) : u(x) \geq \eta\}$, normally $\eta = 0.5$. We set $\varphi = u - \eta$, thus the functions b_1 and b_2 are modified as

$$b_1(u(\mathbf{x}), \Gamma) = 1 - H\left(u(\mathbf{x}) - \eta - \Gamma\right)$$
$$b_2(u(\mathbf{x}), \Gamma) = H\left(u(\mathbf{x}) - \eta + \Gamma\right)$$

Finally, the new 3-D localized version of 3-D PDSS model is given as

$$\begin{aligned} \min_{u,w\in[01]} PDSS_L^{3D}(u,w) = &\int_D |\nabla u|_g d\mathbf{x} + \int_D (I-a_1)^2 w b_1 \, d\mathbf{x} + \theta \int_D C w d\mathbf{x} \\ &+ \int_D (I-a_2)^2 (1-w) b_2 \, d\mathbf{x} + \frac{1}{2\rho} \int_D (u-w)^2 d\mathbf{x} \end{aligned} \tag{4.1}$$

which is equivalent to minimizing the following functional alternately

$$\begin{aligned} \min_u M_{L1}(u,w) = &\int_D |\nabla u|_g d\mathbf{x} + \int_D (I-a_1)^2 w b_1 \, d\mathbf{x} + \frac{1}{2\rho} \int_D (u-w)^2 d\mathbf{x} \\ &+ \int_D (I-a_2)^2 (1-w) b_2 \, d\mathbf{x} \end{aligned} \tag{4.2}$$

$$\begin{aligned} \min_{w\in[01]} M_{L2}(u,w) = &\int_D (I-a_1)^2 w b_1 \, d\mathbf{x} + \int_D (I-a_2)^2 (1-w) b_2 \, d\mathbf{x} \\ &+ \theta \int_D C w d\mathbf{x} + \frac{1}{2\rho} \int_D (u-w)^2 d\mathbf{x} \end{aligned} \tag{4.3}$$

The analytical solution for For (4.3) is defined as

$$w = \min\left\{\max\left\{u - \left((I-a_1)^2 b_1 - (I-a_2)^2 b_2\right)\rho - \rho\theta C, 0\right\}, 1\right\} \tag{4.4}$$

To solve (4.2) using multilevel method, we first introduce the set falling into the Γ-band where $b_1 = b_2 = 1$:

$$V(u) = \{(i,j,k) \,|-\Gamma \leq u_{i,j,k} \leq \Gamma\}$$

All the steps to solve (4.2) using multilevel algorithm are identical except $u = u + c_{i,j,k}$ only needs an updated if the set $[k_1,k_2] \times [\ell_1,\ell_2] \times [m_1,m_2] \cap V(u)$ is non-empty. We will call 3-D MPDSS2 to refer the multilevel algorithm to solve localized version of 3-D PDSS model.

5 New variant of the multilevel algorithms 3-D MPDSS and 3-D MPDSS2

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

Denote the functional in ((3.1)) by $g(u) : \mathbb{R}^{n^3} \to \mathbb{R}$ and represent each subproblem by

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

where $\tilde{\mathbf{e}}$ and K will be defined below, $\mod(\cdot)$ denotes modulo operator. Noting that $b_s = 2^{s-1}$ and $\tau_s = n/b_s$.

Now we investigate the search direction $\{\tilde{\mathbf{e}}\}$;

$$
\begin{aligned}
&\text{level } 1 \quad : \quad 1\times 1 \text{ block}'\text{s index } (i,j,k), \quad 1 \le i,j,k \le n \\
&\qquad\qquad \tilde{\mathbf{e}}^f = \mathbf{e}_f, \quad f = 1,2,\ldots n^3 \\
&\text{level } 2 \quad : \quad 2^1\times 2^1 \text{ block}'\text{s indices } (k_1:k_2, \ell_1:\ell_2, m_1:m_2), \quad 1 \le i,j,k \le \tau_2 \\
&\qquad\qquad \tilde{\mathbf{e}}^{K_0+(k-1)\tau_2^2+(j-1)\tau_2+i} = \sum_{i_1=k_1}^{k_2}\sum_{j_1=\ell_1}^{\ell_2}\sum_{q_1=m_1}^{m_2} \mathbf{e}_{n^2(q_1-1)+n(j_1-1)+i_1} \\
&\qquad\qquad k_1 = 2^1(i-1)+1,\ k_2 = 2^1 i;\ \ell_1 = 2^1(j-1)+1,\ \ell_2 = 2^1 j; \\
&\qquad\qquad m_1 = 2^1(k-1)+1,\ m_2 = 2^1 k \\
&\text{level } 3 \quad : \quad 2^2\times 2^2 \text{ block}'\text{s indices } (k_1:k_2, \ell_1:\ell_2, m_1:m_2), \quad 1 \le i,j,k \le \tau_3 \\
&\qquad\qquad \tilde{\mathbf{e}}^{K_0+K_1+(k-1)\tau_3^2+(j-1)\tau_3+i} = \sum_{i_1=k_1}^{k_2}\sum_{j_1=\ell_1}^{\ell_2}\sum_{q_1=m_1}^{m_2} \mathbf{e}_{n^2(q_1-1)+n(j_1-1)+i_1} \\
&\qquad\qquad k_1 = 2^2(i-1)+1,\ k_2 = 2^2 i;\ \ell_1 = 2^2(j-1)+1,\ \ell_2 = 2^2 j; \\
&\qquad\qquad m_1 = 2^2(k-1)+1,\ m_2 = 2^2 k \\
&\qquad\qquad \vdots \\
&\text{level } L+1 \quad : \quad 2^L\times 2^L \text{ block}'\text{s indices } (1:n, 1:n, 1:n) \\
&\qquad\qquad \tilde{\mathbf{e}}^K = \sum_{i_1=1}^{n}\sum_{j_1=1}^{n}\sum_{q_1=1}^{n} \mathbf{e}_{n^2(q_1-1)+n(j_1-1)+i_1} = \sum_{V=1}^{n^3} \mathbf{e}_V
\end{aligned}
$$

where $\mathbf{e}_\iota \in {}^{n^3}$ is the ι-th unit (coordinate) vector,

$$
\begin{aligned}
K = \sum_{Y=0}^{L} K_Y &= \sum_{Y=0}^{L} \tau_Y^2 \\
&= \sum_{Y=0}^{L} \frac{n^3}{8^Y} = \frac{8n^3-1}{7}
\end{aligned}
$$

Here K represents the total number of search direction across all levels $1, 2, \ldots L, L+1$ for this unconstrained type optimization problem. As shown above, the sequence $\{p^\ell\}$ is clearly essentially periodic (finitely many) and free-steering (spanning $\mathbb{R}^{n^3}$) [25].

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

$$
g(u^\ell) \ge g(v^\ell) \ge g(u^{\ell+1}), \quad v^\ell = (1-t)u^\ell + tu^{\ell+1} \in D_0, \quad \forall\ t \in [0,1]. \tag{5.1}
$$

This property is much stronger than the usual decaying property $g(u^\ell) \ge g(u^{\ell+1})$ which is automatically satisfied by our Algorithm 3-D MPDSS and 3-D MPDSS2.

By [25, Theorem 14.2.7], to ensure the minimizing sequence $\{u_\ell\}$ to be strongly downward, we modify the subproblem $\min M_1^{loc}(u^\ell + cp^\ell)$ to the following

$$
u^{\ell+1} = u^\ell + c^* q^\ell, \qquad c^* = \arg\min\{c \ge 0 \mid \nabla M^T q^\ell = 0\}, \qquad \ell \ge 0 \tag{5.2}
$$

where the ℓ-th search direction is modified to

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

Here the equation $\nabla M^T q^\ell = 0$ for c and the local minimizing subproblem (3.3) that is $\min_c M_1^{loc}(\hat{u}_{i,j} + c)$ are equivalent. Now the new modification is to enforce $c \ge 0$ and the sequence $\{q^\ell\}$ is still essentially periodic.

We shall call the modified algorithm 3-D MPDSS and 3-D MPDSS2 as 3-D MPDSSM and 3-D MPDSS2M respectively.

6 Convergence and complexity analysis

The convergence of the algorithms 3-D MPDSS-3-D MPDSS2

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

is challenging to prove unless a stronger assumption of uniform convexity for the minimization of functional g is made. However there is another approach to prove the convergence of 3-D MPDSSM-3-D MPDSS2M for solving problem ((3.1)) without such an assumption. We first assume the functional $g = g(u)$ is hemivariate i.e. $g(u+t(v-u)) = g(u)$ for t in $[0,1]$ and $u \ne v$ (for theoretical purpose).

There are five sufficient conditions must be met in order to prove the convergence of 3-D MPDSSM-3-D MPDSS2M,

i) $g(u)$ is continuously differentiable in $D_0 = [0,1]^{n^3} \subset \mathbb{R}^{n^3}$;

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

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

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

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^* [25, Theorem 14.1.4]

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

The condition i) is met since we set $\beta \ne 0$ and condition ii) also holds since 'essentially periodic' implies 'uniformly linearly independent' [25, §14.6.3]. We make an assumption of existence of stationary points for $g(u)$ to verify condition v). Next, we now focus on verifying condition iii)-iv). From [25, Theorem 14.2.7], the construction of $\{u^\ell\}$ via (5.2) 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 $\lim_{\ell \to \infty} \|u^{\ell+1} - u^\ell\| = 0$ from [25, Theorem 14.1.3]. Further condition iv) and the fact $\lim_{\ell \to \infty} \|u^{\ell+1} - u^\ell\| = 0$ lead to the result $\lim_{\ell \to \infty} g'(u^\ell) = 0$.

Finally by [25, Theorem 14.1.4], the condition $\lim_{\ell \to \infty} g'(u^\ell) = 0$ 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 3-D MPDSS, 3-D MPDSS2, 3-D MPDSSM and 3-D MPDSS2M. Let $N = n^3$ be the total number of pixels (unknowns). First, we compute the number of floating point operations (flops) for 3-D MPDSS for level s as follows:

Quantities	Flop counts for 3-D MPDSS
ρ term	$2N$
w term	$6N$
r smoothing steps	$125b_s\tau_s^3 r$

Then, the flop counts for all level is $W_{3-DMPDSS} = 6N + \sum_{s=1}^{L+1}\left(2N + 125b_s\tau_s^3 r\right)$ where $s = 1$ (finest) and $s = L+1$ (coarsest). Noting $b_s = 2^{s-1}, \tau_s = n/b_s, N = n^3$, we compute the upper bound for 3-D MPDSS as follows:

$$W_{3-DMPDSS} = 6N + 2(L+1)N + \sum_{s=1}^{L+1}\left(\frac{125Nr}{b_s}\right) = 6N + 2(L+1)N + (125r)N\sum_{s=0}^{L}\left(\frac{1}{2^s}\right)$$
$$< 2N\log n + 8N + 250Nr \approx O\left(N\log N\right)$$

The complexity of the new 3-D MPDSS2 is expected will be directly linked to the length of the segmented objects at each iteration; at the discrete level, this length is usually $O\left(\sqrt[3]{N}\right)$. The upper bound for 3-D MPDSS2 is computed as follows:

$$W_{3-DMPDSS2} = 6\sqrt[3]{N} + 2(L+1)\sqrt[3]{N} + \sum_{s=1}^{L+1}\left(\frac{135r\sqrt[3]{N}}{b_s}\right) = 6\sqrt[3]{N} + 2(L+1)\sqrt[3]{N}$$
$$+(135r)\sqrt[3]{N}\sum_{s=0}^{L}\left(\tfrac{1}{2^s}\right) < 2\sqrt[3]{N}\log n + 8\sqrt[3]{N} + 270r\sqrt[3]{N} \approx O\left(\sqrt[3]{N}\log N\right)$$

The approximate cost of an extra operation $\nabla M^T q^\ell$ in 3-D MPDSSM and 3-D MPDSS2M is $2N$ that results to the total flop counts for 3-D MPDSSM as $W_{3-DMPDSSM} = 6N + \sum_{s=1}^{L+1}\left(4N + 125b_s\tau_s^3 r\right)$. This gives the upper bound for 3-D MPDSSM as

$$W_{3-DMPDSSM} = 6N + 4(L+1)N + \sum_{s=1}^{L+1}\left(\frac{125Nr}{b_s}\right) = 6N + 4(L+1)N + (125r)N\sum_{s=0}^{L}\left(\frac{1}{2^s}\right)$$
$$< 4N\log n + 10N + 250Nr \approx O\left(N\log N\right)$$

.

Finally, for 3-D MPDSS2M, the upper bound for 3-D MPDSS2M is computed as follows:

$$W_{3-DMPDSS2M} = 6\sqrt[3]{N} + 4(L+1)\sqrt[3]{N} + \sum_{s=1}^{L+1}\left(\frac{135r\sqrt[3]{N}}{b_s}\right) = 6\sqrt[3]{N} + 4(L+1)\sqrt[3]{N}$$
$$+(135r)\sqrt[3]{N}\sum_{s=0}^{L}\left(\tfrac{1}{2^s}\right) < 2\sqrt[3]{N}\log n + 10\sqrt[3]{N} + 270r\sqrt[3]{N} \approx O\left(\sqrt[3]{N}\log N\right)$$

Notice that the complexity is optimal for all multilevel algorithm: 3-D MPDSS, 3-D MPDSS2, 3-D MPDSSM and 3-D MPDSS2M and $W_{3-DMPDSS2} < W_{3-DMPDSS2M} < W_{3-DMPDSS} < W_{3-DMPDSSM}$.

7 Numerical experiment

In this research, the performance of all algorithms were compared. The list of the algorithms are listed as follows:

Name	Algorithm		Description
3-D MPDSS	New	:	The multilevel Algorithm 1 for the 3-D PDSS model.
3-D MPDSS2	New	:	The localized model of 3-D MPDSS.
3-D MPDSSM	New	:	The modified multilevel algorithm for 3-D MPDSS.
3-D MPDSS2M	New	:	The modified multilevel algorithm for 3-D MPDSS2.
3-D PDSS	Old	:	The projection method for the 3-D PDSS model [16].
3-D ZC	Old	:	The proposed model by [33].

Four different types of experiments were conducted. In Experiment 1, we segment 3-D synthetic images with different size using 3-D MPDSS, 3-D MPDSS2, 3-D MPDSSM, and 3-D MPDSS2M. Each image was added with noise. To measure the segmentation performance, we record the computation time. A similarity coefficient namely Jaccard (JSC) was computed to measure the segmentation accuracy such that

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

Here, the set of segmented domain D_1 is denoted by S_n while the true set of D_1 is denoted by S_*. The return values of JSC was in the range of $[0, 1]$. A perfect segmentation quality is indicated by a value of 1 while a poor quality of segmentation is indicated by a value of 0.

This is followed by Experiment 2 that demonstrates the performance of 3-D MPDSS, 3-D MPDSS2, 3-D MPDSSM, and 3-D MPDSS2M in segmenting 3-D medical images which are more challenging. The best multilevel algorithm will be selected based on the results from Experiment 1 and 2. In Experiment 3, the chosen method was compared with 3-D PDSS while in Experiment 4, the chosen method was compared with 3-D ZC model. The list of test images are shown in Figure 2. The markers are shown in red, while the built polyhedral surface in green indicates the first (initial) solution based on the markers' locations. MATLAB R2017a was used to implement all experiments, installed in a 16 GB RAM computer with a 3.60GHz processor of Intel Core i7.

For the parameters setting, we fixed the value of $\beta = 10^{-4}$. The value of $\rho \in [7 \times 10^{-4}, 10^{-3}]$, $\Gamma \in [1, 10]$, $\varepsilon \in [1/n^2, 0.01]$, $q \in [1, 10]$ and the inner iteration $t = 1$. The program stop when the

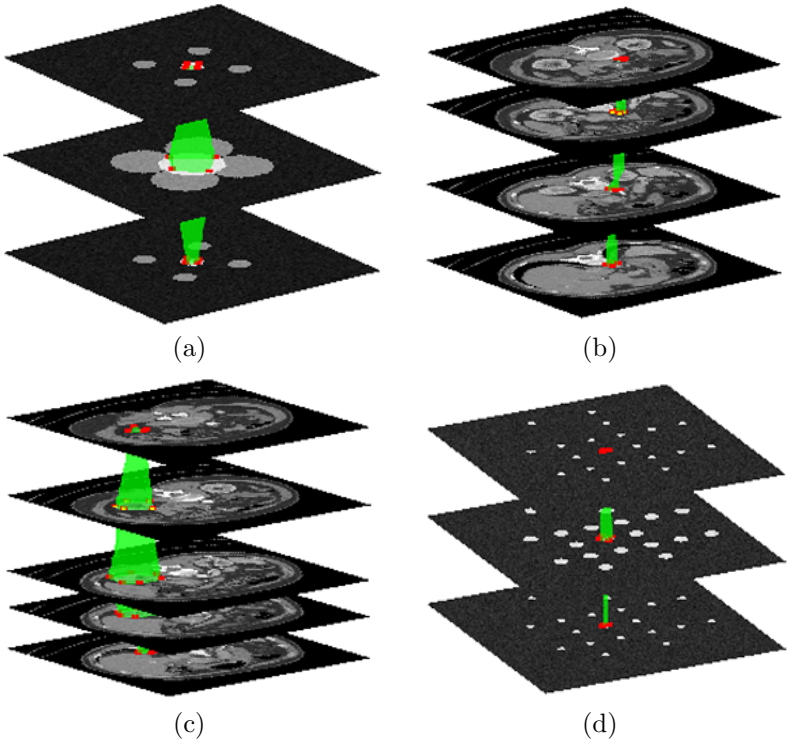


FIGURE 2. List of test images (a) is a synthetic image used in Experiment 1, (b) and (c) are medical images used in Experiment 2 and (d) is a synthetic image used in Experiment 3. The markers are shown in red, while the built polyhedral surface in green indicates the first solution based on the marker's locations.

TABLE 1. Experiment 1–The computation time, t_n and the JSC values for each algorithm with different image resolutions.

Algorithm	Size $N = n^3$	JSC	Time, t_n	$\frac{t_n}{t_{n-1}}$
3-D MPDSS	32^3	1.0	17.2	
	64^3	1.0	67.2	3.9
	128^3	1.0	481.5	7.2
	256^3	1.0	3690.6	7.8
3-D MPDSS2	32^3	1.0	15.3	
	64^3	1.0	39.1	2.6
	128^3	1.0	70.8	1.8
	256^3	1.0	158.2	2.2
3-D MPDSSM	32^3	1.0	24.3	
	64^3	1.0	84.7	3.5
	128^3	1.0	565.1	6.7
	256^3	1.0	4668.5	8.3
3-D MPDSS2M	32^3	1.0	28.3	
	64^3	1.0	61.7	2.2
	128^3	1.0	108.5	1.8
	256^3	1.0	234.9	2.2

outer iteration $maxit = 8000$ or $\max\left(\frac{\|\tilde{u}-u\|_2}{\|\tilde{u}\|_2}, \frac{\|\tilde{w}-w\|_2}{\|\tilde{w}\|_2}\right) < tol$. In this research, $tol = 10^{-2}$ unless stated otherwise. The value of θ varies depending on the images. Basically, the value of θ is large for an object that is close to the nearby object, while a smaller value of θ is needed for clearly separated object.

7.1 Experiment 1: Segmentation on 3-D synthetic image.

In Experiment 1, the performance of 3-D MPDSS, 3-D MPDSS2, 3-D MPDSSM, and 3-D MPDSS2M were compared in segmenting 3-D synthetic image in Figure 2(a). Here, multiple size of images were used and each image was added with additive noise. We use the value of $\theta = 30/(n \times 10^{-4})$.

Figure 3 demonstrates that the algorithms 3-D MPDSS, 3-D MPDSS2, 3-D MPDSSM, and 3-D MPDSS2M successfully segmenting the targeted object in an image of size $128 \times 128 \times 128$ in the first column. Some samples of slice representation results are displayed in the second and third columns. The computation time of 3-D MPDSS2 is 70.8s (1.2 min) which is faster than 3-D MPDSS, 3-D MPDSSM and 3-D MPDSS2M about 6.8, 8 and 1.5 times respectively. Table 1 shows the overall result of computing time for each algorithm in Experiment 1. We also evaluate the computing times ratio in column 5 using the formula t_n/t_{n-1}. The ratio value close to 8.8 shows that the complexity of 3-D MPDSS and 3-D MPDSSM are $O(N \log N)$. Moreover, both 3-D MPDSS2 and 3-D MPDSS2M having $O\left(\sqrt[3]{N} \log N\right)$ complexity as indicated by the ratio 2.2. The computation time for 3-D MPDSS2 is faster than other algorithms for all tested images. We remark that high segmentation accuracy are delivered by all algorithms as indicated by the JSC

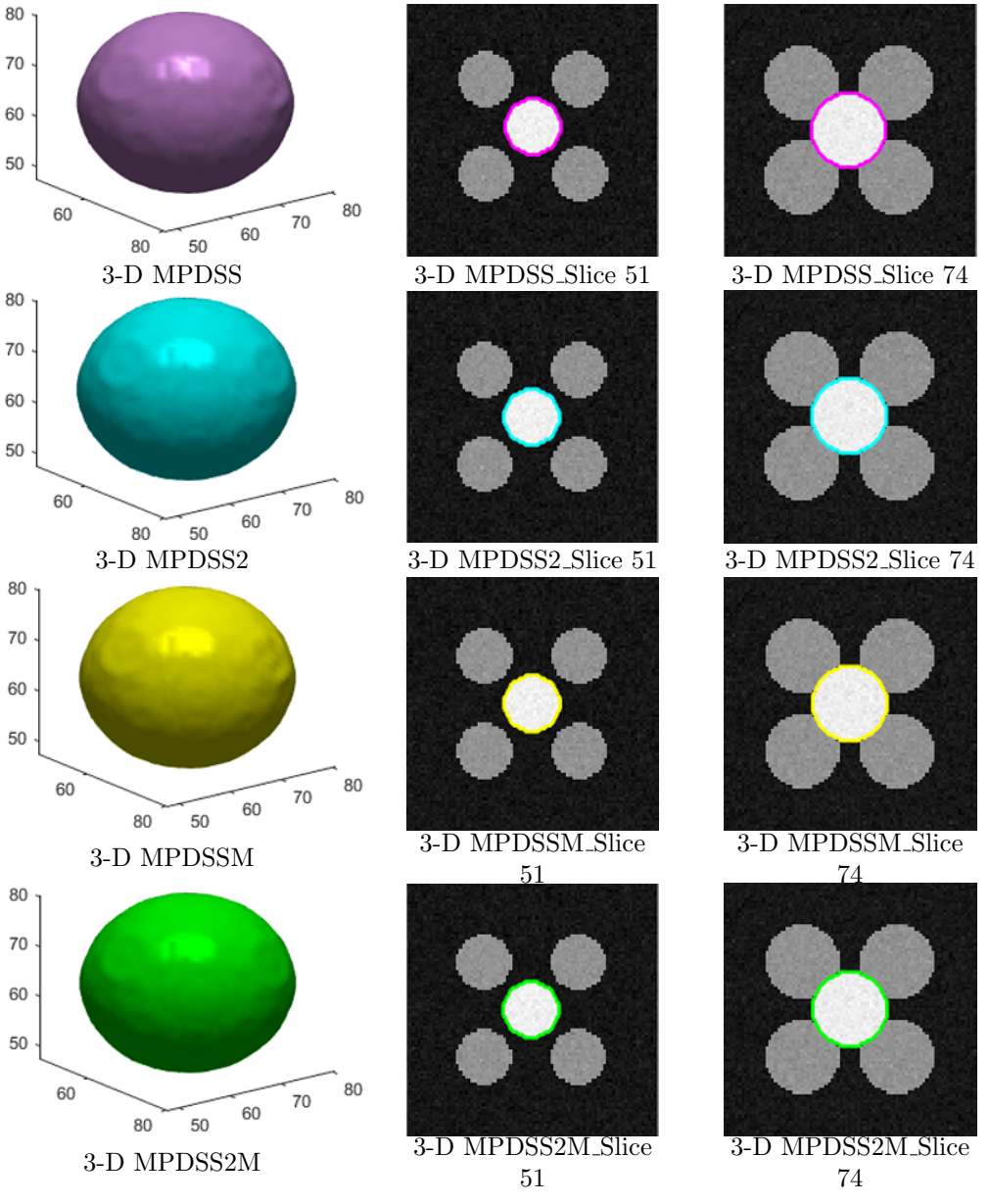


FIGURE 3. Experiment 1 –The first column demonstrates that the algorithms 3-D MPDSS, 3-D MPDSS2, 3-D MPDSSM, and 3-D MPDSS2M successfully segmenting the targeted object in an image of size $128 \times 128 \times 128$. Some samples of slice representation results are shown in the second and third columns.

TABLE 2. Experiment 2–The processing time of 3-D MPDSS, 3-D MPDSS2, 3-D MPDSSM, and 3-D MPDSS2M in segmenting Figure 2(b) and Figure 2(c).

Algorithm	Figure 2(b)	Figure 2(c)
3-D MPDSS	560.1	682.2
3-D MPDSS2	265.3	373.6
3-D MPDSSM	702.0	855.3
3-D MPDSS2M	396.2	520.7

values in third column of Table 1.

The residual plots of 3-D MPDSS, 3-D MPDSS2, 3-D MPDSSM, and 3-D MPDSS2M are illustrated in Figure 4 for segmenting a $64 \times 64 \times 64$ 3-D synthetic data in Figure 2(a). The residual approaches 0 as the iteration is extended up to 10 iterations which indicates the convergence of the algorithms.

7.2 Experiment 2: Segmentation on 3-D medical image

Besides synthetic images, we also tested 3-D MPDSS, 3-D MPDSS2, 3-D MPDSSM, and 3-D MPDSS2M algorithms in segmenting 3-D medical images. They are blood vessel in Figure 2(b)and kidney in 2(c) of size of $128 \times 128 \times 128$ with $\theta = 11200$ and $\theta = 7000$ respectively. The computation time are recorded in Table 2.

Based on the Table 2, the algoritm 3-D MPDSS2 is faster than 3-D MPDSS, 3-D MPDSSM and 3-D MPDSS2M. We are unable to compute the JSC value because the ground truth solutions are not available. To visualize the result, we demonstrate the segmentation output from 3-D MPDSS2 in Figure 5. The first column shows the successful results in 3-D representation while the results in slice representation are demonstrated in the second and third column respectively.

Based on Experiment 1 and 2, the algorithm 3-D MPDSS2 is better compare to the 3-D MPDSS, 3-D MPDSSM, and 3-D MPDSS2M. Hence, we have chosen the algorithm 3-D MPDSS2 to be compared with the existing models.

7.3 Experiment 3: Comparison of 3-D MPDSS2 with 3-D PDSS

In Experiment 3, we make a comparison between the algorithm 3-D MPDSS2 with the existing model, 3-D PDSS by [16]. The 3-D PDSS was solved using projection algorithm in [16]. Here, the object in Figure 2(d) is segmented with different stopping accuracy: $tol = 10^{-1}, 10^{-3}, 10^{-6}$ and 10^{-9}. The size of the image is $256 \times 256 \times 256$ and we used $\theta = 70/(256 \times 10^{-4})$. The computation time is tabulated in Table 3. For $tol = 10^{-1}$, it can be seen that 3-D PDSS is almost as efficient as 3-D MPDSS2. However, as the stopping precision decreases, 3-D MPDSS2 outperforms 3-D PDSS in terms of efficiency. Hence, we deduce that when the degree of stopping accuracy employed is smaller, the convergence of 3-D PDSS is slower than 3-D MPDSS2. Figure 6 shows the segmentation result from both techniques using $tol = 10^{-6}$.

7.4 Experiment 4: Comparison of 3-D MPDSS2 with 3-D ZC

In the final Experiment 4, by utilizing two different initialization, we evaluate the performance of our 3-D MPDSS2 and 3-D ZC [33] in segmenting MRI medical image containing breast illness. $tol = 10^{-2}$ and $\theta = 10^4$ are used in this example. Figure 7(a) depicts the 3-D image with an initial

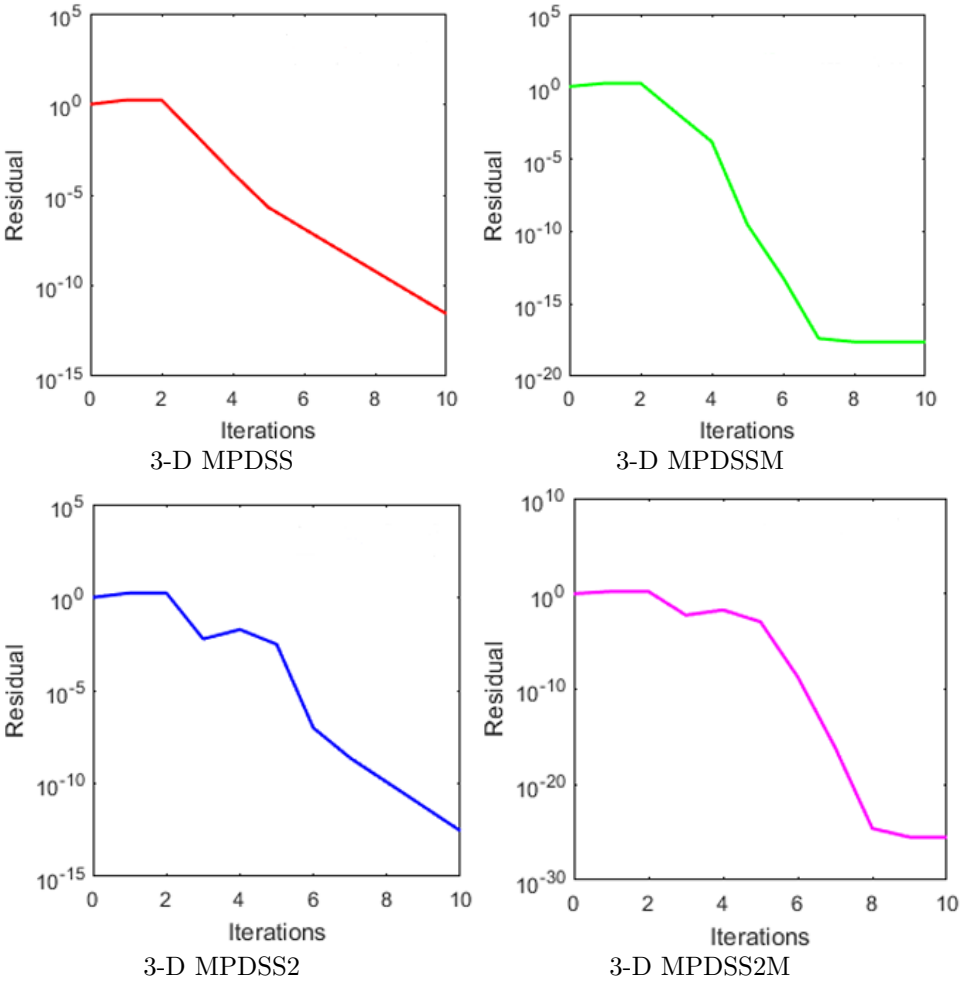


FIGURE 4. Experiment 1–Convergence curve of 3-D MPDSS, 3-D MPDSS2, 3-D MPDSSM, and 3-D MPDSS2M based on the residual value.

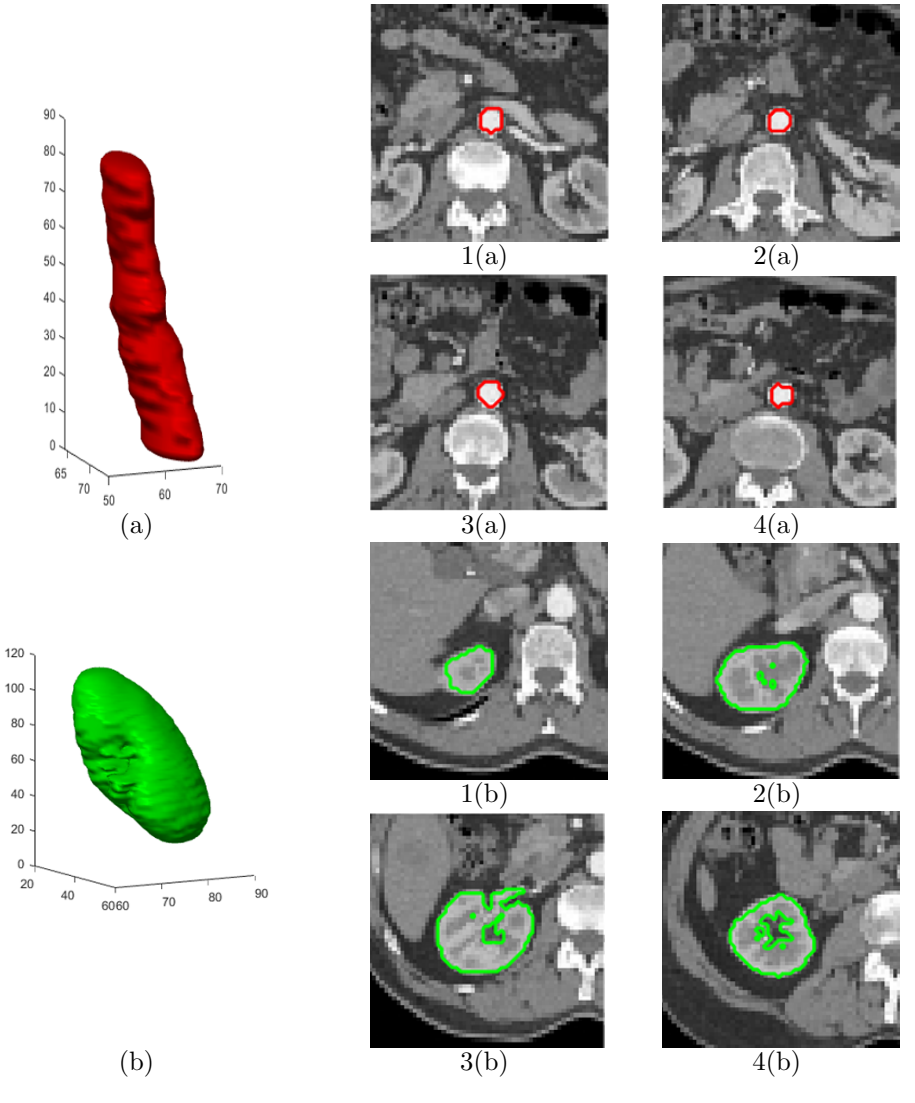
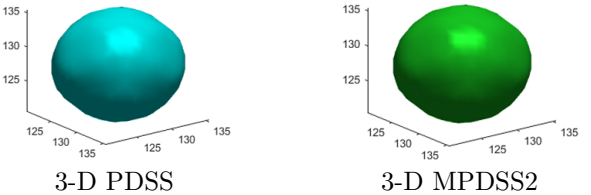


FIGURE 5. Experiment 2–The segmentation output from 3-D MPDSS2.

TABLE 3. Experiment 3–Computation time of 3-D PDSS and 3-D MPDSS2 with different accuracy, *tol*.

Algorithm	Stopping Accuracy	CPU Time	JSC
3-D MPDSS2	10^{-1}	8.9	1.0
	10^{-3}	10.3	1.0
	10^{-6}	11.8	1.0
	10^{-9}	13.5	1.0
3-D PDSS	10^{-1}	12.1	1.0
	10^{-3}	99.2	1.0
	10^{-6}	103.7	1.0
	10^{-9}	757.8	1.0



3-D PDSS 3-D MPDSS2

FIGURE 6. Experiment 3–Results of selective segmentation by 3-D PDSS and 3-D MPDSS2 using $tol = 10^{-6}$.

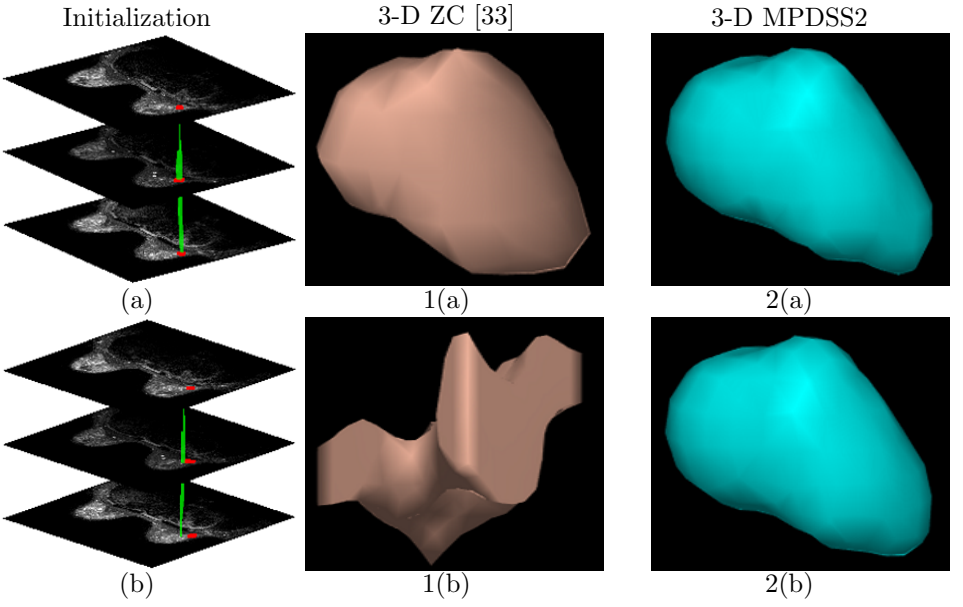


FIGURE 7. Experiment 4–Results of selective segmentation by 3-D ZC and 3-D MPDSS2 using 2 different initial solutions.

solution constructed using the position of markers, whereas Figure 7(b) depicts a 3-D image with an initial solution that is somewhat distant from the marker.

Using the starting solution in Figure 7(a), Figures 7 1(a) and 2(a) show the successful outputs by 3-D ZC and 3-D MPDSS2 algorithms, respectively. The algorithm 3-D MPDSS2 produces a consistent segmentation curve for the second initialization in 7(b), as seen in Figure 7 2(b), demonstrating the benefits of our approach. However, because the minimization issue involves extremely non-convex terms, the segmentation results of 3-D ZC are inconsistent, implying that 3-D ZC is largely reliant on the initialization.

8 Conclusion

To solve the current model, 3-D PDSS, we successfully designed a new 3-D optimization-based multilevel method named 3-D MPDSS in this research. We renamed the 3-D MPDSS algorithm 3-D MPDSS2 after modifying it to take use of a local feature of a targeted item in an image. To improve the decaying property of 3-D MPDSS and 3-D MPDSS2, new variants named 3-D MPDSSM and 3-D MPDS2M are suggested. Experiments on 3-D synthetic and 3-D medical data demonstrate that 3-D MPDSS2 is significantly quicker than 3-D MPDSS, 3-D MPDSSM, and 3-D MPDS2M.

When comparing 3-D MPDSS2 to the existing 3-D PDSS, it is clear that 3-D MPDSS2 outperforms 3-D PDSS at lower stopping accuracy values. There is also a comparison of 3-D MPDSS2

with 3-D ZC. In comparison to 3-D ZC, which is sensitive to initial surface, the results demonstrate that 3-D MPDSS2 is consistent with diverse starting solutions (less dependent on initialization). In the future, we want to provide a 3-D formulation in segmenting 3-D vector valued image

Acknowledgements

This research was supported by the Geran Penyelidikan Khas (GPK) Universiti Teknologi MARA Shah Alam, grant number 600-RMC/GPK 5/3 (185/2020), 2020, Malaysian Ministry of Higher Education. The second author is grateful to the support from the grant EP/N014499/1 from the UK EPSRC.

References

[1] S. A. Abdullah and A. K. Jumaat. Selective image segmentation models using three distance functions. *Journal of Information and Communication Technolog*, 21(1):95–116, 2022.

[2] N. Badshah and K. Chen. Image selective segmentation under geometrical constraints using an active contour approach. *Communications in Computational Physics*, 7(4):759–778, 2010.

[3] 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*, 28:151–167, 2007.

[4] 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*, 6:368–390, 2013.

[5] J. L. Carter. *Dual Method for Total Variation-Based Image Restoration.* PhD thesis, Department of Mathematics, University of California, Los Angeles, CA, USA, 2002.

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

[7] T. F. Chan and K. Chen. An optimization-based multilevel algorithm for total variation image denoising. *Multiscale Model. Simul.*, 5(2):615–645, 2006.

[8] 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(5):1632–1648, 2006.

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

[10] T.F. Chan and L. A. Vese. *Active Contour and Segmentation Models using Geometric PDE's for Medical Imaging. In: Malladi R. (eds) Geometric Methods in Bio-Medical Image Processing. Mathematics and Visualization.* Springer, Berlin, Heidelberg, 2002.

[11] 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*, 39:155–173, 2005.

[12] C.L. Guyader and C. Gout. Geodesic active contour under geometrical conditions theory and 3d applications. *Numerical Algorithms*, 48:105–133, 2008.

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

[14] A. K. Jumaat and K. Chen. Fast algorithm for selective image segmentation model. *International Journal of Engineering and Technology (UAE)*, 7(4):41–44, 2018.

[15] A. K. Jumaat and K. Chen. A reformulated convex and selective variational image segmentation model and its fast multilevel algorithm. *Numerical Mathematics: Theory, Methods and Applications*, 12(2):403–437, 2019.

[16] A. K. Jumaat and K. Chen. Three-dimensional convex and selective variational image segmentation model. *Malaysian Journal of Mathematical Sciences*, 14(3):437–450, 2020.

[17] A. K. Jumaat, W. E. Z. W. A. Rahman, A. Ibrahim, and R. Mahmud. Segmentation and characterization of masses in breast ultrasound images using active contour. In *IEEE International Conference on Signal and Image Processing Applications, ICSIPA 2011*, pages 404–409, 2011.

[18] A. K. Jumaat, W. E. Z. W. A. Rahman, S. S. Yasiran, R. Mahmud, and A. A. Malek. Masses characterization based on angular margin measurement. In *Proceedings of International Conference on Computational Intelligence, Modelling and Simulation*, pages 265–269, 2012.

[19] A. K. Jumaat, S. S. Yasiran, A. A. Malek, W. E. Z. W. A. Rahman, N. Badrin, S. R. Rafiee S. H. Osman, and R. Mahmud. Performance comparison of canny and sobel edge detectors on balloon snake in segmenting masses. In *International Conference on Computer and Information Sciences, ICCOINS 2014*, pages 1–5, 2014.

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

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

[22] 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*, 113:946–965, 2009.

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

[24] 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):3734–3743, 2012.

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

[26] L. Rada and K. Chen. Improved selective segmentation model using one level set. *Journal of Algorithm and Computational Technology*, 7:509–541, 2013.

[27] L. Rada and K. Chen. A variational model and its numerical solution for local, selective and automatic segmentation. *Numerical Algorithm*, 66:399–430, 2014.

[28] J. Spencer and K. Chen. A convex and selective variational model for image segmentation. *Communication in Mathematical Sciences*, 13:1453–1452, 2015.

[29] X. Xie and M. Mirmehdi. Radial basis function based level set interpolation and evolution for deformable modelling. *Image Vision Comput.*, 29:167–177, 2011.

[30] S. S. Yasiran, A. K. Jumaat, A. A. Malek, F. H. Hashim, N. D. Nasrir, S. N. A. S. Hassan, N. Ahmad, and R. Mahmud. Microcalcifications segmentation using three edge detection techniques. In *International Conference on Electronic Devices, Systems, and Applications*, pages 207–211, 2012.

[31] S. S. Yasiran, A. K. Jumaat, M. Manaf, A. Ibrahim, W. E. Z. W.. A. Rahman, A. A. Malek, M. F. Laham, and R. Mahmud. Comparison between gvf snake and ed snake in segmenting microcalcifications. In *IEEE Conference on Computer Applications and Industrial Electronics*, pages 597–601, 2011.

[32] N. A. K. Zaman, W. E. Z. W. A. Rahman, A. K. Jumaat, and S. S. Yasiran. Classification of breast abnormalities using artificial neural network. In *AIP Conference Proceedings*, pages 1–7, 2015.

[33] J. Zhang and K.Chen. A fast algorithm for automatic segmentation and extraction of a single object by active surfaces. *Int. J. of Computer Mathematics*, 92:1251–1274, 2015.

[34] J. Zhang, K.Chen, and B.Yu. A multigrid algorithm for the 3d chan-vese model of variational image segmentation. *Int. J. of Computer Mathematics*, 89:160–189, 2011.

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