J Sci Comput
DOI 10.1007/s10915-016-0207-8

Convergent Non-overlapping Domain Decomposition Methods for Variational Image Segmentation

Yuping Duan[1] · **Huibin Chang**[2] · **Xue-Cheng Tai**[3]

Received: 25 December 2015 / Revised: 5 April 2016 / Accepted: 7 April 2016

Abstract This paper concerns with the non-overlapping domain decomposition methods (DDMs) for the Chan–Vese model in variational image segmentation. We work with a saddle point formulation for the non-overlapping decompositions, which leads to independent subproblems decoupled by the primal–dual algorithm. With the non-overlapping DDMs, only the interfaces of the adjacent subdomains are coupled, which means the information transmission takes place on such interfaces and therefore makes the proposed DDMs flexible and efficient. Moreover, we consider both the stripe-type and checkerboard-type decomposition methods and provide the rigorous proof of the convergence. Our numerical experiments demonstrate that the proposed DDMs are convergent, efficient, and quite robust with respect to the model parameters and image resolutions.

Keywords Chan–Vese model · Non-overlapping domain decomposition methods · Image segmentation · Primal–dual algorithm

Mathematics Subject Classification 68U10 · 65M55 · 74S20

1 Introduction

Image segmentation plays an important role in computer vision and machine learning, which aims to separate a given image into several disjoint parts, each of which shares certain visual

The research was partially supported by Natural Science Foundation of China, Grant No. 11426165, No. 11501413 and Tianjin 131 Talent Project.

✉ Huibin Chang
changhuibin@gmail.com

[1] Center for Applied Mathematics, Tianjin University, Tianjin, People's Republic of China

[2] School of Mathematical Sciences, Tianjin Normal University, Tianjin, People's Republic of China

[3] Department of Mathematics, University of Bergen, Mons, Norway

Published online: 21 April 2016

characteristics, such as image intensities, colors and textures etc. Researchers have developed a wide range of methods for image segmentation including the thresholding methods, histogram-based methods, edge detection based methods, region-growing methods, partial differential equations (PDEs) based methods and variational methods, and graph partitioning methods, etc. [1].

Variational methods, including both edge-based models and region-based models, have been established to extract objects of interest in images for the task of segmentation. The active contour/snake model was proposed by Kass et al. [2], which deforms an initial contour towards the boundaries of the objects. It is a very typical representative of the edge-based models, where a lot of efforts have been stimulated to improve the model, e.g., the geometric active contours [3,4] and the geodesic active contours [5]. As one of the most successful region-based models, the Mumford–Shah (MS) model [6] was built up to pursue a piecewise smooth approximation of the given image. Letting $\Omega \subset \mathbb{R}^2$ be open and bounded and Γ be a closed Jordan curve in Ω, Mumford and Shah proposed to minimize the following functional to find a piecewise smooth approximation u and edge Γ

$$\min_{u,\Gamma} \alpha \int_{\Omega} (f-u)^2 dx + \mu \int_{\Omega\setminus\Gamma} |\nabla u|^2 dx + |\Gamma|, \tag{1.1}$$

where μ and α are positive parameters and $|\Gamma|$ denotes the length of Γ if given an observed image $f : \Omega \to \mathbb{R}$. The MS model requires to deal with both the shape optimizations and topology changes. However, due to the non-convexity, it is very challengeable to seek the minimizer, which may trap into local minima.

One of the most successful convex relaxation of the MS functional was proposed by Chan and Vese [7]. By assuming the given images as piecewise constant, it seeks for an approximation with a binary piecewise constant representation through a level set formulation [8]. Letting Γ be its boundaries, the Chan–Vese (CV) model is

$$\min_{c_1,c_2,\Gamma} MS(c_1, c_2, \Gamma) = \alpha \left(\int_{\Sigma} (f-c_1)^2 + \int_{\Omega\setminus\Sigma} (f-c_2)^2 \right) + |\Gamma|, \tag{1.2}$$

where c_1, c_2 are two constants that approximate the image intensity in Σ and $\Omega\setminus\Sigma$, respectively. One can obtain the binary valued segmentation results based on the total variation regularization [9] and the thersholding technique [10]. The two-phase segmentation model (1.2) was extended by Vese and Chan [11] to the multiple phase cases using a multiple level set formulation. There also exist other related convex CV models in [12–14].

The main purpose of this paper is to explore more efficient algorithms for the convex CV model [10]. The challenge to design efficient algorithms for such kind of models is the non-differentiability of the total variation semi-norm. In general, the existing algorithms can be classified into three categories based on the nature of manipulating the primal and dual variables and one can refer to [15]. The first category is the primal approach, such as the gradient descent method etc. [9,16–21]. In order to accelerate these methods, the most popular splitting algorithms were proposed based on the Bregman iteration [22,23], augmented Lagrangian methods (ALM) [24,25] and the alternative direction method of multiplier (ADMM) [26,27], etc. The second category is the dual approach [28], where a gradient projection method was designed for the dual formulation of the Rudin–Osher–Fatemi (ROF) model [9]. The third category is the primal–dual approach [29,30], which was further studied for image processing applications in [31–33].

Domain decomposition methods (DDMs) and multigrid methods are well-known techniques for solving the large-scale PDEs. However, their applications in image processing are relatively limited. The DDMs can break down a large scale problem into a sequence

of smaller scale subproblems, which could be solved in parallel. The space decomposition and subspace correction ideas for the minimization problems were adopted in [34–37]. Recently some researches were devoted to the non-differential non-additive total variation based optimization problems for image processing problems. Firsov and Lui proposed the DDMs with Dirichlet boundary condition for image denoising related to Gaussian curvature [38]. The overlapping DDMs were used based on a primal–dual formulation for the anisotropic total variation problem [39]. The subspace correction methods were applied to variational image restoration and segmentation in [40–42], where the original total variation based problems were successfully decomposed into the smaller-size subproblems. Two level DDMs involving a coarse mesh space correction were further considered [42]. Xu et al. [43] applied the DDMs to the image deblurring problems, which coupled with a convolution operator. Chang et al. [44] extended the DDMs to the nonlocal total variation (NLTV) based image restoration, where a unified framework was proposed for the NLTV and ROF model. Chang et al. [45,46] introduced a stable "unity decomposition" and derived an overlapping DDMs for the dual formulation, where the convergence rates were first deduced for the DDMs of total variation minimization problems. Some variants of the classical DDMs have been studied in [47–50]. Fornasier and Schönlieb [47] and Fornasier et al. [48] introduced the surrogate functional and generated an iterative proximity-map of the subproblems, where the subproblems were solved by the fast oblique thresholding. Such algorithms performed efficiently for image restoration and compress sensing problems and the weak convergence of the algorithms was proved as well. Hintermüller and Langer [49] studied the general TV–L^1–L^2 model for mixed noise removal and the related DDMs, where the convergence and monotone decay of the associated objective functional values for the Successive Subspace Correction (SSC) were guaranteed. A non-overlapping DDM was proposed by Hintermüler and Langer [51] for the dual formulation of the anisotropical total variation based image denoising, where fast solvers for the subproblems and convergence analysis of the algorithm were given as well. Similarly to [50], Lee et al. [52] proposed a non-overlapping block decomposition methods, which introduced a primal–dual stitching method.

Comparing with the overlapping DDMs, we prefer to use the non-overlapping DDMs for two reasons: (1) the ease of handling the constraints, where we do not need to introduce any sophisticated technique as [53]; (2) less communication cost and the convergence guarantee. Therefore, we mainly focus on the non-overlapping DDMs for the variational image segmentation tasks. Stimulated by the work in [54,55], the DDMs are build upon the coupled relations on the interfaces of the adjacent subdomains. The efficient primal–dual algorithm [31] is used to solve the subproblems. Actually, we use the DDMs to handle the more complicated model since the minimizer is not unique, and the optimization procedure is coupled with a convex set constraint. Moreover, the proposed DDMs are proved to be convergent theoretically. Meanwhile, inferred from our numerical experiments, the proposed DDMs are very efficient and stable, even for the case with the updating schemes.

The paper is organized as follows. The convex Chan–Vese model is reviewed in Sect. 2. In Sect. 3, the proposed DDMs are given with the convergence analysis. The numerical experiments are provided to verify the convergence and efficiency of the DDMs in Sect. 4. We conclude the paper with some remarks in Sect. 5.

2 Convex Chan–Vese Model

On the first place, we review the convex Chan–Vese (CCV) model [10] for (1.2), which can be expressed as follows

$$\min_{0\le h\in \mathrm{BV}(\Omega)\le 1} \mathrm{TV}(h;\Omega)+\alpha\int_\Omega \left(h(f-c_1)^2+(1-h)(f-c_2)^2\right),$$

where

$$\mathrm{TV}(h;\Omega):=\sup\left\{-\int_\Omega \mathrm{div}(\boldsymbol{\phi})h : \boldsymbol{\phi}=(\phi_1,\phi_2)\in \left(C_c^1(\Omega)\right)^2, \|\boldsymbol{\phi}\|_{L^\infty}\le 1\right\},$$

$\mathrm{div}(\boldsymbol{\phi}):=\partial_{x_1}\phi_1+\partial_{x_2}\phi_2$, $\mathrm{BV}(\Omega):=\{h\in L^1(\Omega):\mathrm{TV}(h)<+\infty\}$. By fixing c_1, c_2, the above CCV model is reduced to the following form

$$\min_{h\in V} \quad F(h)=\mathrm{TV}(h)+\alpha\langle s,h\rangle, \tag{2.1}$$

where $V=K_h\bigcap \mathrm{BV}(\Omega)$, $K_h=\{h:0\le h\le 1\}$, $\langle\cdot,\cdot\rangle$ denotes the inner product over the space $L^2(\Omega)$. In (2.1), we define

$$s(x):=2(c_2-c_1)(f(x)-(c_1+c_2)/2),\quad \forall x\in\Omega,$$

and $\mathrm{TV}(h):=\mathrm{TV}(h;\Omega)$.

Theorem 2.1 *The CCV model* (2.1) *admits at least a minimizer* $h^*\in V$, *i.e.*

$$h^*=\arg\min_{0\le h\le 1} F(h). \tag{2.2}$$

Remark 2.2 One can readily know that CCV model is not strictly convex, that can not guarantee a unique minimizer.

Theorem 2.3 ([10]) *For any given* $c_1, c_2\in\mathbb{R}$, 1_Σ *by thresholding* h^* *as for a.e.* $\mu\in[0,1]$ *in* (2.2) *is the global minimizer to* $MS(c_1,c_2,\cdot)$,*where* $\Sigma:=\{x:h(x)^*\ge\mu\}$.

In practise, one needs to provide suitable values of c_1, c_2. Generally speaking, there exist two approaches to obtain the mean values for unsupervised segmentation tasks. One is to compute the values from the image histogram. The other is to iteratively update the values from

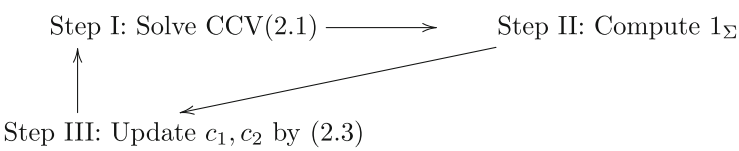
$$c_1=\frac{\langle f,h\rangle}{\int_\Omega h},\quad \text{and}\quad c_2=\frac{\langle f,1-h\rangle}{\int_\Omega(1-h)}. \tag{2.3}$$

Actually the following Table 1 illuminates the procedure of the two phase segmentation problem. We consider to solve (2.1) instead of the original CV model (1.2) by the DDMs, since the left parts including the updating c_i, $i=1,2$ can be accelerated readily by parallel computing.

3 Non-overlapping DDMs for Chan–Vese Models

We follow the non-overlapping DDM framework in [54,55], where the original problem defined on the whole domain is localized with a consistency constraint. Indeed only one layer of the subdomains are coupled with the adjacent subdomains. Here we do not adopt any first order or higher order transmission conditions as [56], since the optimization variables in (2.1) belong to $\mathrm{BV}(\Omega)$, which is not sufficiently smooth.

Table 1 Flow-process diagram for two phase segmentation



3.1 Non-overlapping DDMs with Two Subdomains

In the continuous setting, assume that there exists the decomposition of the space $V = \bigoplus_{i=1}^{2} V_i$, where $V_i = \{h : 0 \le h(x) \le 1,\ \forall x \in \Omega_i\} \cap BV(\Omega_i)$, associated with the non-overlapping domain decomposition as follows

$$\Omega = \Omega_1 \cup \Omega_2, \qquad \mathring{\Omega}_1 \cap \mathring{\Omega}_2 = \emptyset,$$

and the interface $\Gamma = \Omega_1 \cap \Omega_2$. We consider the minimization problem followed [55] as

$$\begin{aligned} &\min_{h_i \in V_i} \quad \sum_{i=1}^{2} F_i(h_i)\ , \\ &\text{s.t.,} \quad h_1 = h_2 \text{ on } \Gamma, \end{aligned} \tag{3.1}$$

where $F_i(h) = \mathrm{TV}(h; \Omega_i) + \alpha \langle s, h \rangle_{\Omega_i}$, and $\langle \cdot, \cdot \rangle_{\Omega_i}$ denotes the L^2 inner product over Ω_i. One can readily observe that the subproblems of the above minimization problem are linked by the constraints on the interfaces for adjacent subdomains. By introducing a Lagrangian multiplier $\lambda : \Gamma \to \mathbb{R}$, the saddle point problem for (3.1) can be built up as follows

$$\max_{g} \min_{h_i \in V_i} \sum_{i=1}^{2} F_i(h_i) + \langle g, h_1 - h_2 \rangle_\Gamma. \tag{3.2}$$

Therefore, if a better decoupled method is employed to solve the above saddle point problem, we can design the efficient DDMs to realize fast image segmentation.

For simplicity and the purpose of real applications, we rebuild the framework of the non-overlapping DDMs in the discrete setting hereafter, where the same notations such as Ω, Ω_i and Γ are adopted as in continuous setting. Figure 1 illustrates the non-overlapping domain decomposition in the discrete setting. Assume that the image is of the resolution $\mathscr{M} \times \mathscr{N}$, i.e., $\Omega := \{0, 1, 2, \ldots, \mathscr{M} - 1\} \times \{0, 1, 2, \ldots, \mathscr{N} - 1\}$. The two subdomains of the non-overlapping DDM with the stripe-style as shown in Fig. 1 are $\Omega_1 := \{0, 1, 2, \ldots, [(\mathscr{M} - 1)/2]\} \times \{0, 1, 2, \ldots, \mathscr{N} - 1\}$ and $\Omega_2 := \{[(\mathscr{M} - 1)/2], [(\mathscr{M} - 1)/2] + 1, \ldots, \mathscr{M} - 1\} \times \{0, 1, 2, \ldots, \mathscr{N} - 1\}$, respectively. The interface of the two subdomains $\Gamma = \{[(\mathscr{M} - 1)/2]\} \times \{0, 1, 2, \ldots, \mathscr{N} - 1\}$. Therefore, one can define the function

$$h_i \in V_i := \left\{ v : \Omega_i \to \mathbb{R},\ \ 0 \le v^{j_1, j_2} \le 1, \quad \forall (j_1, j_2) \in \Omega_i \right\}.$$

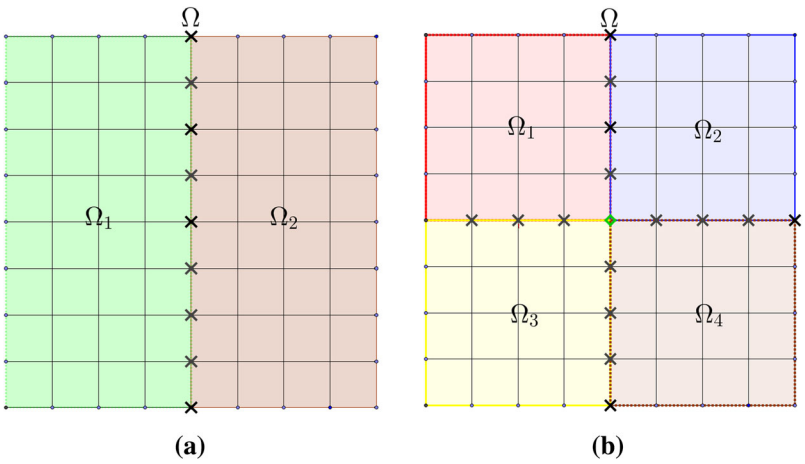


Fig. 1 Discrete domain decomposition with stripe-style in **(a)** and checkerboard-style in **(b)**

In the following, we consider how to decompose the objective functional (2.1) over the subdomains. First, we rewrite the discrete form of $F(h)$ as follows

$$\begin{aligned} F(h) &= \sum_{(j_1,j_2)\in\Omega} \left(\left| (Dh)^{j_1,j_2} \right| + \alpha s^{j_1,j_2} h^{j_1,j_2} \right) \\ &= \sum_{(j_1,j_2)\in\Omega_1\setminus\Gamma} \left(\left| (Dh)^{j_1,j_2} \right| + \alpha s^{j_1,j_2} h^{j_1,j_2} \right) + \sum_{(j_1,j_2)\in\Omega_2} \left(\left| (Dh)^{j_1,j_2} \right| + \alpha s^{j_1,j_2} h^{j_1,j_2} \right) \\ &:= F_1(h) + F_2(h), \end{aligned} \tag{3.3}$$

where Dh denotes the forward difference discretization of the gradient operator and $|\cdot|$ denotes the isotropic or the anisotropic absolute value of a vector, i.e., $|[v_1, v_2]| = \sqrt{|v_1|^2 + |v_2|^2}$ or $|[v_1, v_2]| = |v_1| + |v_2|$, respectively.

Next, we introduce two variables $h_i : \Omega_i \to \mathbb{R}$, $i = 1, 2$, for arbitrary $h : \Omega \to \mathbb{R}$, which satisfies that

$$h_i := h|_{\Omega_i}.$$

Conversely, in order to reconstruct h from h_i, one needs additional condition for h_i, that is to say,

$$h_1|_\Gamma = h_2|_\Gamma.$$

Therefore, we obtain an equivalent form as (3.1) for the original problem (2.1), where the same notations are used for the discrete setting as in the continuous setting. The Lagrangian functional of the constrained optimization (3.1) is built up as follows

$$\mathcal{L}(h_1, h_2; g) := \sum_{i=1}^{2} F_i(h_i) + \langle g, h_1 - h_2 \rangle_\Gamma, \tag{3.4}$$

where $g \in V_\Gamma := \{v : \Gamma \to \mathbb{R}\}$ and $\langle g, h\rangle_\Gamma := \sum_{(j_1,j_2)\in\Gamma} g^{j_1,j_2} h^{j_1,j_2}$.

We can solve (3.4) by considering the following saddle point problem

$$\max_{g\in V_\Gamma} \min_{h_i \in V_i} \mathcal{L}(h_1, h_2; g). \tag{3.5}$$

One can readily have the equivalence between the saddle point problem (3.5) and the original optimization problem (3.1).

3.2 Numerical Algorithm

Define the extensive operator $\mathbf{E}_{\Gamma\to\Omega} : V_\Gamma \to V$ as

$$\mathbf{E}_{\Gamma\to\Omega}(g)^{j_1,j_2} = \begin{cases} g^{j_1,j_2}, & \text{if } (j_1, j_2) \in \Gamma, \\ 0, & \text{otherwise,} \end{cases}$$

and the restrictive operator $\mathbf{R}_{S\to T} : V_S \to V_T$ as

$$\mathbf{R}_{S\to T}(h)^{j_1,j_2} = h^{j_1,j_2}, \quad \text{if } (j_1, j_2) \in T,$$

where $T \subseteq S$. We employ the primal–dual algorithm [57] (see the "Appendix") to solve (3.5) shown as Algorithm I.

Algorithm I: Primal–Dual algorithm for (3.5)

1. Initialization: choose $g^0 = 0$, $\bar{h}_i^0 = h_i^0 = 0$, and select parameters τ, σ and $\theta \in [0, 1]$.
2. Iterations for $n \geq 0$: Update h_i^n, g^n, $\bar{h}_i^n$ as follows:

$$\begin{cases} \text{Step I}: \ g^{n+1} = g^n + \sigma(\mathbf{R}_{\Omega_1\to\Gamma}\bar{h}_1^n - \mathbf{R}_{\Omega_2\to\Gamma}\bar{h}_2^n), \\ \text{Step II}: \ (h_1^{n+1}, h_2^{n+1}) = \arg\min_{h_i\in V_i} \sum_{i=1}^{2} \left(F_i(h_i) + \dfrac{1}{2\tau} \sum_{(j_1,j_2)\in\Omega_i} |h_i^{j_1,j_2} - (\hat{h}_i^n)^{j_1,j_2}|^2 \right), \\ \qquad\qquad \text{with } (\hat{h}_1^n, \hat{h}_2^n) = (h_1^n, h_2^n) - \tau \left(\mathbf{E}_{\Gamma\to\Omega_1} g^{n+1}, -\mathbf{E}_{\Gamma\to\Omega_2} g^{n+1} \right), \\ \text{Step III}: \ \bar{h}_i^{n+1} = (1+\theta)h_i^{n+1} - \theta h_i^n. \end{cases} \tag{3.6}$$

End till some stopping criterion meets.

Remark 3.1 Here we do not solve the saddle point problem by the ADMM [26,27] because the corresponding subproblems defined on the adjacent subdomains by the ADMM will be coupled together.

We give some observations of Algorithm I. For the subproblem in Step II of (3.6), one can solve the minimization problem with respect to h_i independently, which means the subproblems can be solved in parallel.

Now we will separately discuss how to obtain h_1^{n+1} and h_2^{n+1} in the following. On one hand, the subproblem h_1^{n+1} can be explicitly defined as follows

$$
\begin{aligned}
h_1^{n+1} &= \arg\min_{h_1\in V_1} F_1(h_1) + \frac{1}{2\tau}\sum_{(j_1,j_2)\in\Omega_1}\left|h_1^{j_1,j_2} - \left(\hat{h}_1^n\right)^{j_1,j_2}\right|^2 \\
&= \arg\min_{h_1\in V_1} \sum_{(j_1,j_2)\in\Omega_1\setminus\Gamma}\left|(Dh_1)^{j_1,j_2}\right| + \left(\sum_{(j_1,j_2)\in\Omega_1\setminus\Gamma}\alpha s_1^{j_1,j_2}h_1^{j_1,j_2}\right. \\
&\qquad \left. + \frac{1}{2\tau}\sum_{(j_1,j_2)\in\Omega_1}\left|h_1^{j_1,j_2} - \left(\hat{h}_1^n\right)^{j_1,j_2}\right|^2\right) \\
&:= \arg\min_{h_1\in V_1} \sum_{(j_1,j_2)\in\Omega_1\setminus\Gamma}\left|(Dh_1)^{j_1,j_2}\right| + \mathscr{H}_1(h_1),
\end{aligned}
\tag{3.7}
$$

where $\mathscr{H}_1(h_1) := \sum_{(j_1,j_2)\in\Omega_1\setminus\Gamma}\alpha s_1^{j_1,j_2}h_1^{j_1,j_2} + \frac{1}{2\tau}\sum_{(j_1,j_2)\in\Omega_1}|h_1^{j_1,j_2} - (\hat{h}_1^n)^{j_1,j_2}|^2$.

By applying the algorithm in the appendix to the subproblem (3.7) and defining $K_p = \{\boldsymbol{p} = (p_1, p_2) : \sqrt{p_1^2 + p_2^2} \le 1\}$, we can obtain Algorithm II-I for the subproblem h_1^{n+1}.

Algorithm II-I: Primal–Dual algorithm for (3.7)

1. Initialization: choose $\boldsymbol{p}_1^0 = 0$, $\check{h}_1^0 = h_1^0 = 0$, and select parameters κ, γ and $\theta \in [0, 1]$.
2. For $k = 0, 1 \ldots,$

$$
\begin{cases}
\boldsymbol{p}_1^{k+1} = \arg\min\limits_{\boldsymbol{p}_1} I_{K_p}(\boldsymbol{p}_1) + \dfrac{1}{2\kappa}\displaystyle\sum_{(j_1,j_2)\in\Omega_1\setminus\Gamma}|\boldsymbol{p}_1 - \hat{\boldsymbol{p}}_1^k|^2, \\
\qquad \text{where } \hat{\boldsymbol{p}}_1^k = \boldsymbol{p}_1^k + \kappa\mathbf{R}_{\Omega_1\to\Omega_1\setminus\Gamma}(\nabla\check{h}_1^k), \\
h_1^{k+1} = \arg\min\limits_{h_1} \alpha\displaystyle\sum_{(j_1,j_2)\in\Omega_1}\mathbf{E}_{\Omega_1\setminus\Gamma\to\Omega_1}(s_1)h_1 + I_{K_h}(h_1) \\
\qquad\qquad + \dfrac{1}{2\gamma}\displaystyle\sum_{(j_1,j_2)\in\Omega_1}|h_1 - \tilde{h}_1^k|^2 + \dfrac{1}{2\tau}\sum_{(j_1,j_2)\in\Omega_1}|h_1 - \hat{h}_1^n|^2, \\
\qquad \text{where } \tilde{h}_1^k = h_1^k + \gamma\mathbf{E}_{\Omega_1\setminus\Gamma\to\Omega_1}(\operatorname{div}\boldsymbol{p}_1^{k+1}), \\
\check{h}_1^{k+1} = (1+\theta)h_1^{k+1} - \theta h_1^k.
\end{cases}
$$

End till some stopping criterion meets.

The variable $\boldsymbol{p}_1^{k+1}$ in Algorithm II-I has the closed-form form as follows

$$
\boldsymbol{p}_1^{k+1}(x) = \frac{\hat{\boldsymbol{p}}_1^k(x)}{\max\{1, |\hat{\boldsymbol{p}}_1^k(x)|\}}, \quad \forall x \in \Omega_1\setminus\Gamma,
$$

and the variable h_1^{k+1} is computed from

$$
h_1^{n+1}(x) = \min\left\{1, \max\left\{0, \tau\tilde{h}_1^k + \gamma\hat{h}_1^n - \tau\gamma\alpha\mathbf{E}(s_1)(x)\right\}/(\tau+\gamma)\right\}, \quad \forall x \in \Omega_1.
$$

On the other hand, the subproblem of h_2^{n+1} is formulated as follows

$$\begin{aligned} h_2^{n+1} &= \arg\min_{h_2\in V_2} F_2(h_2) + \frac{1}{2\tau}\sum_{(j_1,j_2)\in\Omega_2}\left|h_2^{j_1,j_2} - \left(\hat{h}_2^n\right)^{j_1,j_2}\right|^2 \\ &= \arg\min_{h_2\in V_2}\sum_{(j_1,j_2)\in\Omega_2}\left|(Dh_1)^{j_1,j_2}\right| + \left(\sum_{(j_1,j_2)\in\Omega_2}\alpha s_2^{j_1,j_2}h_1^{j_1,j_2}\right. \\ &\quad \left. + \frac{1}{2\tau}\sum_{(j_1,j_2)\in\Omega_2}\left|h_1^{j_1,j_2} - \left(\hat{h}_1^n\right)^{j_1,j_2}\right|^2\right) \\ &:= \arg\min_{h_2\in V_2}\sum_{(j_1,j_2)\in\Omega_2}\left|(Dh_2)^{j_1,j_2}\right| + \mathscr{H}_2(h_2), \end{aligned} \tag{3.8}$$

where $\mathscr{H}_2(h_2) := \sum_{(j_1,j_2)\in\Omega_2}\alpha s_2^{j_1,j_2}h_2^{j_1,j_2} + \frac{1}{2\tau}\sum_{(j_1,j_2)\in\Omega_2}|h_2^{j_1,j_2} - (\hat{h}_2^n)^{j_1,j_2}|^2$.

Similarly to Algorithm II-I, we can obtain Algorithm II-II for the subproblem h_2^{n+1}.

Algorithm II-II: Primal–Dual algorithm for (3.8)

1. Initialization: choose $\boldsymbol{p}_2^0 = 0$, $\check{h}_2^0 = h_2^0 = 0$, and select parameters κ, γ and $\theta \in [0, 1]$.
2. For $k = 0, 1\ldots,$

$$\begin{cases} \boldsymbol{p}_2^{k+1} = \arg\min_{\boldsymbol{p}_2} I_{K_p}(\boldsymbol{p}_2) + \frac{1}{2\kappa}\sum_{(j_1,j_2)\in\Omega_2}|\boldsymbol{p}_2 - \hat{\boldsymbol{p}}_2^k|^2, \\ \qquad \text{with } \hat{\boldsymbol{p}}_2^k = \boldsymbol{p}_2^k + \kappa\nabla\check{h}_2^k, \\ h_2^{k+1} = \arg\min_{h_2} \alpha\sum_{(j_1,j_2)\in\Omega_2} s_2h_2 + I_{K_h}(h_2) + \frac{1}{2\gamma}\sum_{(j_1,j_2)\in\Omega_2}|h_2 - \tilde{h}_2^k|^2 \\ \qquad + \frac{1}{2\tau}\sum_{(j_1,j_2)\in\Omega_2}|h_2 - \hat{h}_2^n|^2, \\ \qquad \text{with } \tilde{h}_2^k = h_2^k + \gamma\,\mathrm{div}\,\boldsymbol{p}_2^{k+1}, \\ \check{h}_2^{k+1} = (1+\theta)h_2^{k+1} - \theta h_2^k. \end{cases}$$

End till some stopping criterion meets.

It is clear that $\boldsymbol{p}_2^{k+1}$ has the closed-form form as

$$\boldsymbol{p}_2^{k+1}(x) = \frac{\hat{\boldsymbol{p}}_2^k(x)}{\max\left\{1, |\hat{\boldsymbol{p}}_2^k(x)|\right\}}, \quad \forall x \in \Omega_2,$$

and h_2^{k+1} can be computed from

$$h_2^{n+1}(x) = \min\left\{1, \max\left\{0, \tau\tilde{h}_2^k + \gamma\hat{h}_2^n - \tau\gamma\alpha s_2(x)\right\}/(\tau+\gamma)\right\}, \quad \forall x \in \Omega_2.$$

Remark 3.2 The Algorithm II-I and Algorithm II-II are almost the same except for the dimension of $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$. Therefore additional restrictive and extensive operators are introduced in Algorithm II-I.

3.3 Extension to the Cases with More Than Two Subdomains

One can also deduce the DDMs for the cases with T non-overlapping subdomains ($T > 2$), i.e.,

$$\Omega := \bigcup_{i=1}^{T} \Omega_i,$$

with the inner boundaries $\Gamma_{i,k} = \Omega_i \bigcap \Omega_k$, where $\Gamma_{i,k} = \Gamma_{k,i}$. To describe the set of boundary information transmission, we denote $\hat{\Gamma} = \{\Gamma_{i,k} : \ \Gamma_{i,k}$ has more than one point$\}$, where we only consider the transfer between two adjacent regions sharing a boundary (not a point). We split the functional F as $F := \sum_{1\le i\le T} F_i$ such that F_i is only defined upon the subdomain Ω_i, and define the minimization problem as follows

$$\begin{aligned} &\min_{h_i\in V_i} \ \sum_{i=1}^{T} F_i(h_i)\ , \\ &\text{s.t.,} \ \ h_i = h_k \text{ on } \Gamma_{i,k} \in \hat{\Gamma}, \quad \forall\ 1 \le i < k \le T. \end{aligned} \tag{3.9}$$

Remark 3.3 One can readily give the details for splitting of the functional F. For the stripe-type domain decomposition, it is the same as (3.3). And similar splitting technique can be employed for the checkerboard-type domain decomposition.

Similarly to the case of two subdomains, we can build up the following saddle point problem

$$\max_{g_{i,k}} \min_{h_i\in V_i} \sum_{i=1}^{T} F_i(h_i) + \sum_{\Gamma_{i,k}\in\hat{\Gamma}} \langle g_{i,k}, h_i - h_k\rangle_{\Gamma_{i,k}}.$$

By considering the checkerboard-type decomposition, Algorithm III gives the details of solving the associated saddle point problem.

Algorithm III: Primal–Dual algorithm for (3.5) with checkerboard-type domain decomposition

1. Initialization: choose $g_{i,k}^0 = 0$, $\bar{h}_i^0 = h_i^0 = 0$, and select parameters τ, σ and $\theta \in [0, 1]$.
2. Iterations for $n \ge 0$: Update h_i^n, $g_{i,k}^n$, $\bar{h}_i^n$ as follows:

$$\begin{cases} g_{i,k}^{n+1} = g_{i,k}^n + \sigma \mathbf{E}_{\Gamma_{i,k}\to\Omega}(\mathbf{R}_{\Omega_i\to\Gamma_{i,k}}\bar{h}_i - \mathbf{R}_{\Omega_k\to\Gamma_{i,k}}\bar{h}_k), \forall\ 1 \le i < k \le T \\ (h_1^{n+1}, h_2^{n+1}, \ldots) = \arg\min\limits_{h_i\in V_i} \sum\limits_{i=1}^{T} \Big(F_i(h_i) + \dfrac{1}{2\tau} \sum\limits_{(j_1,j_2)\in\Omega_i} |h_i^{j_1,j_2} - (\hat{h}_i^n)^{j_1,j_2}|^2 \Big), \\ \quad \text{with } \ \hat{h}_i^n = h_i^n - \tau \left(\sum\limits_{\left\{k:\ k>i, \Gamma_{i,k}\in\hat{\Gamma}\right\}} \mathbf{E}_{\Gamma_{i,k}\to\Omega_i} g_{i,k}^{n+1} - \sum\limits_{\left\{k:\ k<i, \Gamma_{k,i}\in\hat{\Gamma}\right\}} \mathbf{E}_{\Gamma_{k,i}\to\Omega_i} g_{k,i}^{n+1} \right), \\ \bar{h}_i^{n+1} = (1+\theta) h_i^{n+1} - \theta h_i^n. \end{cases} \tag{3.10}$$

End till some stopping criterion meets.

As mentioned, one uses an iteratively scheme to refine the mean values, the update schemes of which are given in the discrete setting as follows

$$c_1 = \frac{\sum_{(j_1,j_2)\in\Omega} f^{j_1,j_2} h^{j_1,j_2}}{\sum_{(j_1,j_2)\in\Omega} h^{j_1,j_2}} \quad \text{and} \quad c_2 = \frac{\sum_{(j_1,j_2)\in\Omega} f^{j_1,j_2}(1-h^{j_1,j_2})}{\sum_{(j_1,j_2)\in\Omega}(1-h^{j_1,j_2})},$$

where we assume that $\sum_{(j_1,j_2)\in\Omega} h^{j_1,j_2} \neq 0$ and 1. The above update schemes could be incorporated into Algorithm I, where c_1, c_2 can be initialized as 0 and 1, respectively.

3.4 Convergence Analysis

In order to analyze the convergence of the proposed Algorithm I, we shall first define the following operator

$$\mathbf{K}(h_1, h_2) = \mathbf{E}_{\Gamma\to\Omega}\big(\mathbf{R}_{\Omega\to\Gamma}(h_1 - h_2)\big), \quad \forall\, h_1, h_2$$

for domain decomposition with two subdomains. For T subdomains, one can define

$$\mathbf{K}_T(h_1, h_2, \ldots, h_T)$$

as

$$[\mathbf{K}_T(h_1, h_2, \ldots, h_T)]_{i,k} = \mathbf{E}_{\Gamma_{i,k}\to\Omega}(\mathbf{R}_{\Omega_i\to\Gamma_{i,k}} h_i - \mathbf{R}_{\Omega_k\to\Gamma_{i,k}} h_k).$$

Denote the discrete L^2 inner product by $\langle\cdot,\cdot\rangle$.

Proposition 3.1 *The norm of the operator $\mathbf{K}_T$ is bounded.*

Proof We can readily prove the boundedness of this operator by the property of the linear operator in the finite dimension space. □

In order to set the parameters, we need to further study the upper bound of the operator $\mathbf{K}_T$. For the simplicity, we first analyze it by using the stripe-type domain decomposition

$$\begin{aligned}
&\langle \mathbf{K}_T(h_1, h_2, \ldots, h_T), \mathbf{K}_T(h_1, h_2, \ldots, h_T)\rangle \\
&= \sum_{1\le i\le T-1} \sum_{(j_1,j_2)\in\Gamma_{i,i+1}} \left(h_i^{j_1,j_2} - h_{i+1}^{j_1,j_2}\right)^2 \\
&\le 2 \sum_{1\le i\le T-1} \sum_{(j_1,j_2)\in\Gamma_{i,i+1}} \left(\left(h_i^{j_1,j_2}\right)^2 + \left(h_{i+1}^{j_1,j_2}\right)^2\right) \\
&\le 2 \sum_{1\le i\le T} \sum_{(j_1,j_2)\in\Omega_i} \left(h_i^{j_1,j_2}\right)^2.
\end{aligned}$$

There exists $\|\mathbf{K}_T\|^2 \le 2$ for the stripe-type domain decompositions. For the checkerboard-type domain decomposition, we have

$$\begin{aligned}
&\langle \mathbf{K}_T(h_1, h_2, \ldots, h_T), \mathbf{K}_T(h_1, h_2, \ldots, h_T)\rangle \\
&= \sum_{\Gamma_{i,k}\in\hat{\Gamma}} \sum_{(j_1,j_2)\in\Gamma_{i,k}} \left(h_i^{j_1,j_2} - h_k^{j_1,j_2}\right)^2
\end{aligned}$$

$$\leq 2 \sum_{\Gamma_{i,k} \in \hat{\Gamma}} \sum_{(j_1,j_2) \in \Gamma_{i,k}} \left(\left(h_i^{j_1,j_2} \right)^2 + \left(h_k^{j_1,j_2} \right)^2 \right)$$

$$\leq 8 \sum_{1 \leq i \leq T} \sum_{(j_1,j_2) \in \Omega_i} \left(h_i^{j_1,j_2} \right)^2 .$$

There exists $\|\mathbf{K}_T\|^2 \leq 8$ for the checkerboard-type domain decomposition. By defining the norm of $\mathbf{K}$ as $L := \|\mathbf{K}_T\| < +\infty$, we can have the following theorem.

Theorem 3.4 *Algorithm I is convergent if* $\tau\sigma < 1/L^2$.

Proof Following [31] and Proposition 3.1, one can readily prove this theorem.

Theorem 3.5 *Algorithm II-I and Algorithm II-II for solving the subproblems of Algorithm I are both convergent if* $\kappa\gamma < 1/8$.

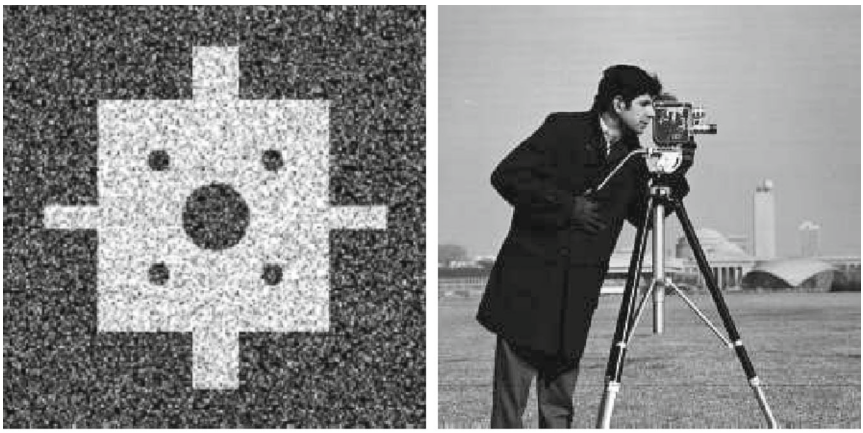
4 Numerical Experiments

In this section, we conduct several numerical experiments to verify the performance of the proposed DDMs. All the experiments are implemented in a **MATLAB** environment on a desktop with Intel (R) Core(TM) i3-2120 CPU@3.30GHz, and 8G RAM. Hereafter, we set the parameter $\theta = 1$.

4.1 Performance and Convergence

We first test our proposed DDMs on two images as shown in Fig. 2, where the left one is a synthetic image and the right one is a natural image.

In order to verify the convergence, we shall have the exact solution. Alternatively, we numerically get h^* as a "ground truth" solution by the primal–dual algorithm [31] without



(a) **(b)**

Fig. 2 Both images in **(a)** and **(b)** are of the resolution 256×256

the DDMs (see the "Appendix"), where the stopping condition is set as

$$\left\| h^k - h^{k-1} \right\| / \left\| h^k \right\| \leq \text{TOL}, \tag{4.1}$$

with $\text{TOL} = 1.0 \times 10^{-15}$ or the maximum iteration number reaches 1.0×10^6. Two measurements are used to qualify the convergence of the proposed DDMs, i.e., the normalized error $e_k := \|h^k - h^*\|/\|h^*\|$ and the numerical energy $E_k := F(h_k)$.

Firstly, we show the performance of the DDMs on Fig. 2a with the stripe-type domain decomposition. The parameters are set as $\tau = 0.5, \sigma = 1.0, \kappa = 1.0, \gamma = 1/8$. We implement Algorithm I with the stopping condition (4.1) with $\text{TOL} = 1.0 \times 10^{-5}$ or the maximum iteration number $N_{out} = 1.0 \times 10^3$. Here, the inner iteration number of Algorithm II-I and Algorithm II-II are fixed to be $N_{in} = 1.0 \times 10^3$ in order to guarantee the convergence. Figure 3 shows the segmentation results with different α and Fig. 4 displays the differences images between the ground truth and the solutions of the DDMs. Both two figures demonstrate that our proposed DDMs performs well and are quite robust with respect to different number of subdomains and different parameter α.

Secondly, we implement Algorithm III to show the performance of the DDMs with the checkerboard-type domain decomposition. The segmentation results and differences between the ground truth and the solutions by the DDMs are shown in Fig. 5 and in Fig. 6, respectively. Inferred from these two figures, we observe that the DDMs based algorithm works well for the checkerboard-type domain decomposition. In order to further verify our DDMs, we test both algorithms on the natural image in Fig. 2b and the results are shown in Figs. 7 and 8. Once again, both decomposition methods produce satisfactory and accurate segmentation results on the nature image.

The convergence curves of the stripe-type decomposition (i.e., Fig. 3h) are plotted in Fig. 9, where both the errors and the numerical energy of the objective functional decrease

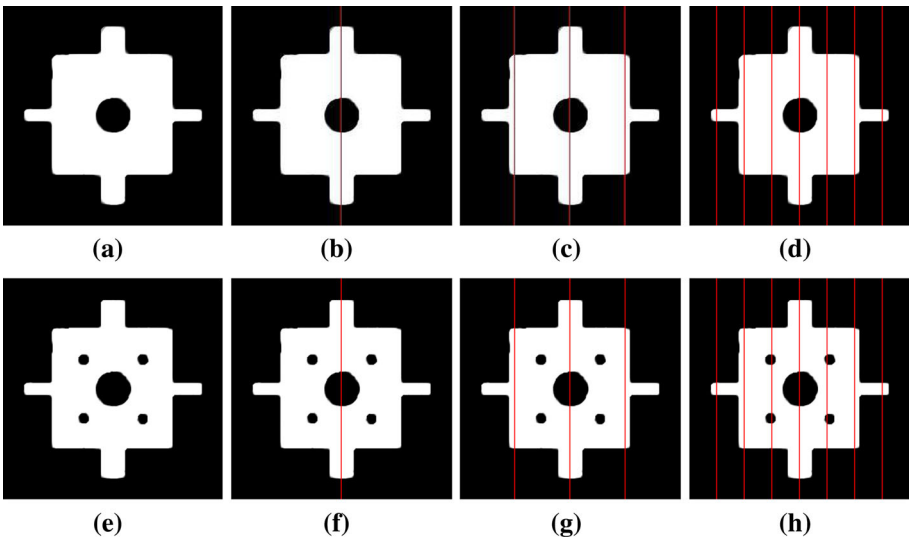


Fig. 3 Performance of DDMs with the stripe-type domain decomposition (*red lines* represent the interfaces of domain decomposition). The parameter α for the *first row* is $\alpha = 0.4$ and the *second row* is $\alpha = 0.6$. From *left* to *right* the ground truth with different α, the segmentation results by the DDMs with 2, 2^2 and 2^3 subdomains, respectively (Color figure online)

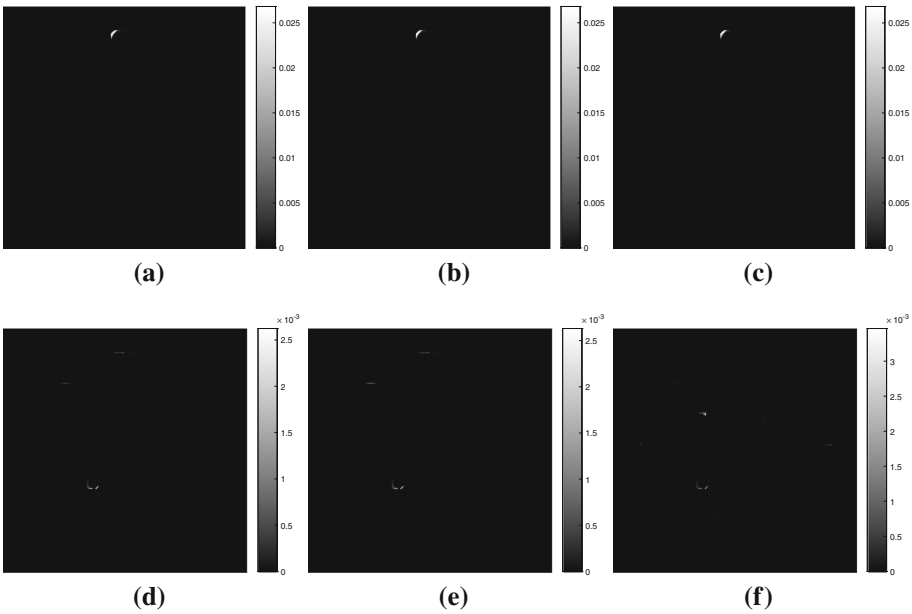


Fig. 4 Differences between the ground truth and the solution by the DDMs with the stripe-type domain decomposition of Fig. 3. From *left* to *right*: differences by DDMs with 2, 2^2 and 2^3 subdomains, respectively

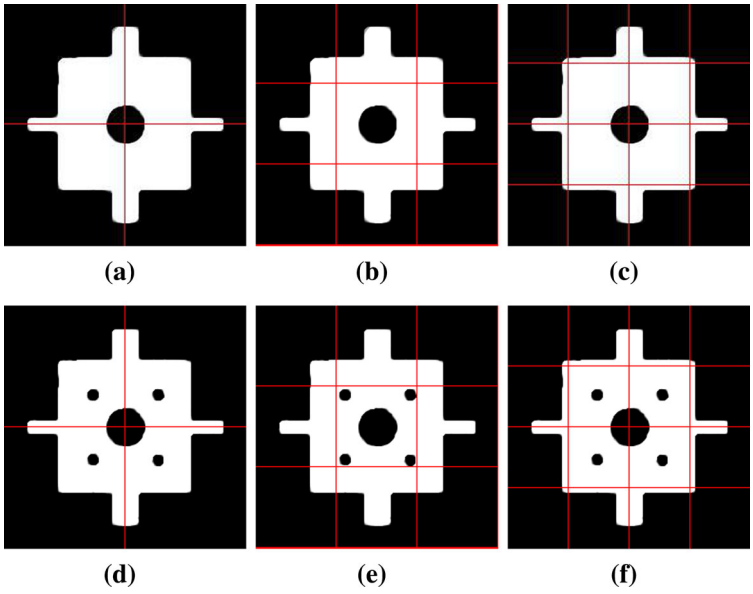


Fig. 5 Performance of DDMs with checkerboard-type domain decomposition. The parameter α for the *first row* is $\alpha = 0.4$ and the *second row* is $\alpha = 0.6$. From *left* to *right* the segmentation results by DDMs with 2^2, 3^2, and 4^2 subdomains, respectively

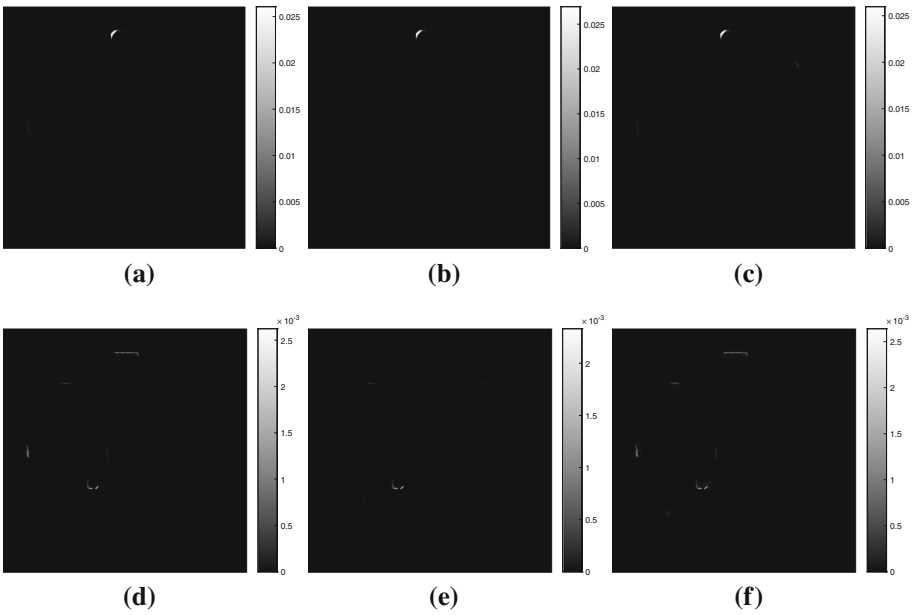


Fig. 6 Differences for Fig. 5 between the ground truth and the solution by the checkerboard-type domain decomposition. From *left* to *right* the differences by DDMs with 2^2, 3^2, and 4^2 subdomains, respectively

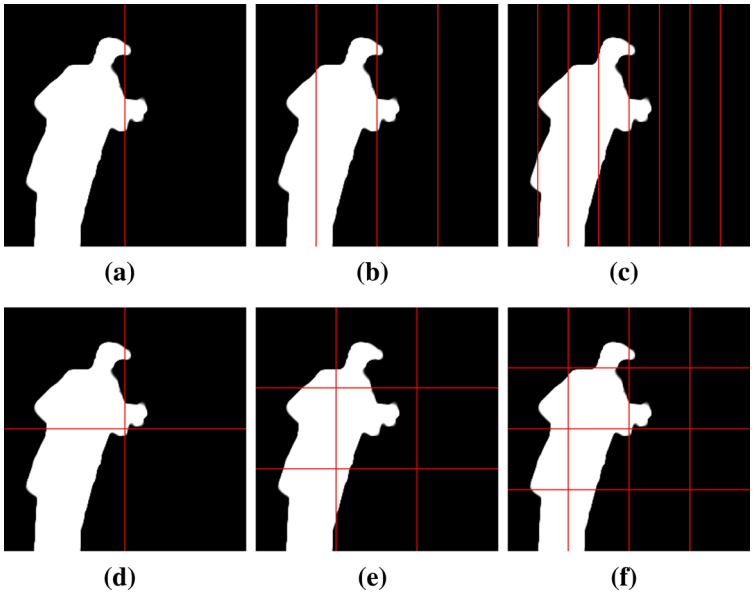


Fig. 7 Performance of DDMs with stripe-type (the *first row*) and checkerboard-type (the *second row*) domain decomposition. From *left* to *right* on the *first row* are the segmentation result by the DDMs with 2, 2^2, and 2^3 subdomains while on the *second row* are the segmentation results by the DDMs with 2^2, 3^2, and 4^2 subdomains, respectively

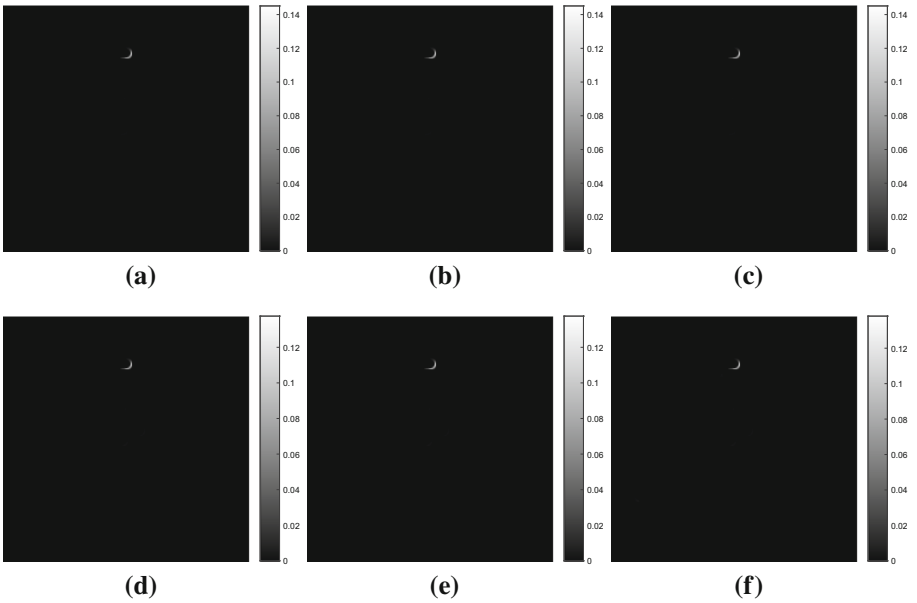


Fig. 8 Differences between ground truth and the solution by the DDMs with stripe-type (the *first row*) and checkerboard-type (the *second row*) decomposition w.r.t. Fig. 7

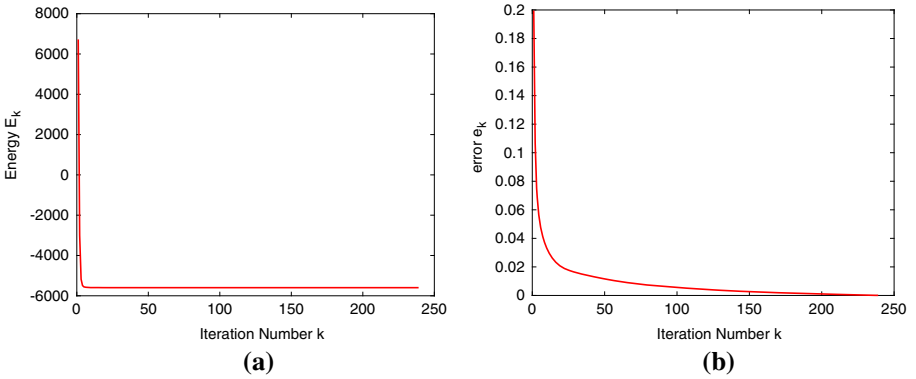


Fig. 9 The convergence curves of the DDMs w.r.t. Fig. 3h. **(a)** Decay of numerical energy E_k. **(b)** Errors e_k

as the iteration number increases. We use $c_1 = 1, c_2 = 0$ for Fig. 2a and $c_1 = 0.5, c_2 = 0$ for Fig. 2b as the default values. We see that the errors and the objective functional values monotonely decrease fast, which numerically verifies our theoretical analysis.

Additional tests are done on the medical images of different resolutions, i.e., Fig. 10a with the resolution 512×512 and Fig. 10d with the resolution 1024×1024. In the experiment, we set $\alpha = 1, c_1 = 0.5, c_2 = 0$ for Fig. 10a and $\alpha = 50, c_1 = 0, c_2 = 1$ for Fig. 10d. All other parameters and stopping conditions for the DDMs are the same as the previous experiments. We display the segmentation results in Fig. 10, which demonstrates that the DDMs work quite well for images with different resolutions.

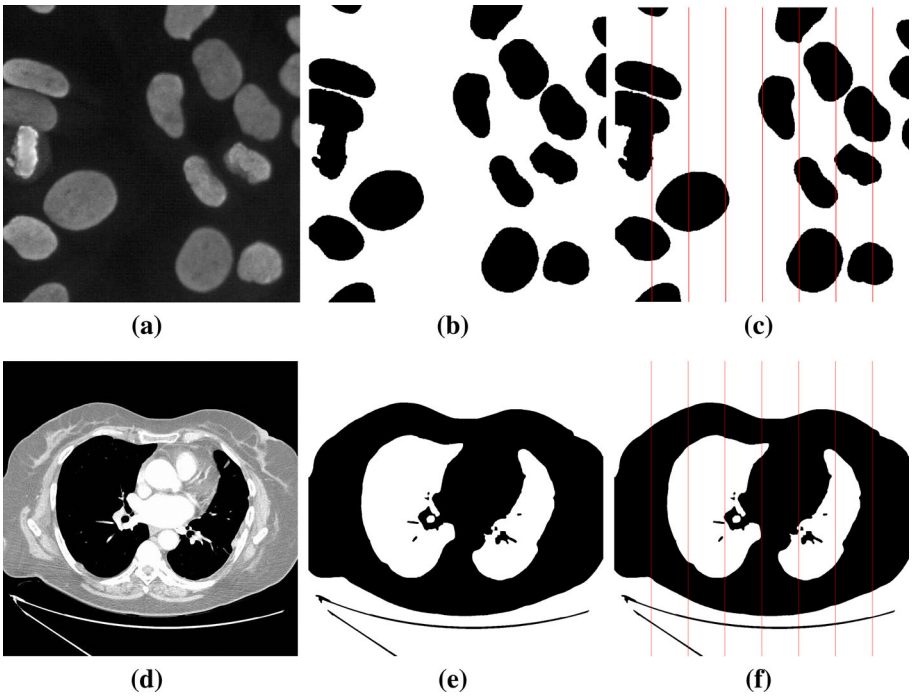


Fig. 10 Performance of the DDMs with stripe-type domain decomposition for **a** with the resolution 512×512 and **d** with the resolution 1024×1024. The values of the parameters for the *first row* are $\alpha = 1, c_1 = 0.5, c_2 = 0$ and for the *second row* are $\alpha = 50, c_1 = 0, c_2 = 1$. From *left* to *right* the original images, the ground truth, the segmentation results by the DDMs with 2^3 subdomains

4.2 Performance w.r.t. the Inner Iteration Number N_{in}

In order to discover the performance of the DDMs affected by the iteration number of the inner solver of Algorithm II-I and Algorithm II-II, we test the DDMs with different inner iterations under the stopping condition as

$$e_k \leq 5.0 \times 10^{-5} \tag{4.2}$$

for the stripe-type decomposition of 2 subdomains. All other parameters are set to be the same as the above subsection. The results of the testing image Fig. 2a are shown in Table 2. By inferring from the table, one can see that our DDMs with less inner iterations require more outer iterations N_{out} to obtain the same error. Meanwhile, the DDMs become slower gradually as the cost of per outer iteration increases although less outer iterations are used. Therefore, a suitable inner iteration number N_{in} needs to be chosen to achieve the best performance. Hereafter, we set $N_{in} = 10$ as the default value.

4.3 Performance w.r.t. the Number of Subdomains and Other Parameters

We use Fig. 2a and set $N_{in} = 10$ and other parameters are set to be the same as Subsection 4.1. We first test the performance of the DDMs w.r.t. different numbers of the subdomains

Table 2 Performance of the stripe-type decomposition with different inner iterations N_{in}

N_{in}	Time	e_k (1.0×10^{-5})	E_k	N_{out}
1	20	4.9987	−5597.0078	1960
2	15	4.9965	−5597.0078	991
5	12	4.9918	−5597.0079	389
10	11	4.9860	−5597.0080	201
20	20	4.5986	−5597.0090	174
50	43	4.2034	−5597.0097	172
100	85	4.2638	−5597.0099	172
200	167	4.3518	−5597.0100	172
500	418	4.3966	−5597.0100	172
1000	833	4.3978	−5597.0100	172

Time denotes the elapsed time in seconds, e_k denotes the error, E_k denotes the objective functional values and N_{out} denotes the outer iterations

Table 3 Performance of the stripe-type decomposition with different numbers of the subdomains N_{sub}

N_{sub}	Time	e_k (1.0×10^{-5})	E_k	N_{out}
2	11	4.9860	−5597.0080	201
4	12	4.9906	−5597.0080	202
8	18	4.9922	−5597.0082	285
16	22	4.9979	−5597.0078	301
24	32	4.9918	−5597.0080	379

Time denotes the elapsed time in seconds, e_k denotes the error, E_k denotes the objective functional values and N_{out} denotes the outer iterations

N_{sub}. We choose the stopping condition as (4.2) and display the results in Table 3. As the number of subdomain increases, more outer iterations are required to reach the given error tolerance. On the other hand, the segmentation results are not so sensitive to the number of subdomains. Thus, we may consider to incorporate some global information into the original minimization problem in our future work.

We also evaluate the performance of our DDMs w.r.t. other parameters including $\tau, \sigma, \kappa, \gamma$. The effect of parameters κ, γ has been discussed in [31]. In our proposed DDMs, these parameters play the role of controlling the inner solver for the subproblems, which are quite similar to [31]. Therefore, we only show the impact of the parameters τ, σ, which actually control the outer solver for the DDMs. In the tests ahead, we set $\kappa = 1, \gamma = 1/8$ as the default values. The iterations for the DDMs stop if (4.2) is satisfied or the maximum iteration number reaches 1.0×10^3. Our estimation demonstrates the upper bound of the multiplication of the two parameters are $\tau\sigma = 1/2$ to guarantee the convergence. Hence, we conduct the experiments with $\tau\sigma = 1/2$. By selecting σ in $\{2^{-m}, 2^{-m+1}, \ldots, 0, \ldots, 2^{m-1}, 2^m\}$ with $m = 5$, we show the results in Fig. 11, which illustrates that a suitable value of σ is needed for the best performance. In this paper, we do not conclude any rule for selecting the optimal parameters for the DDMs and leave it as a future work.

4.4 Performance with the Updating Means

In this subsection, instead of selecting the means c_1, c_2 manually, we use the updating schemes (2.3) to compute the means automatically. We set $\alpha = 1$ and stopping condition (4.1) of

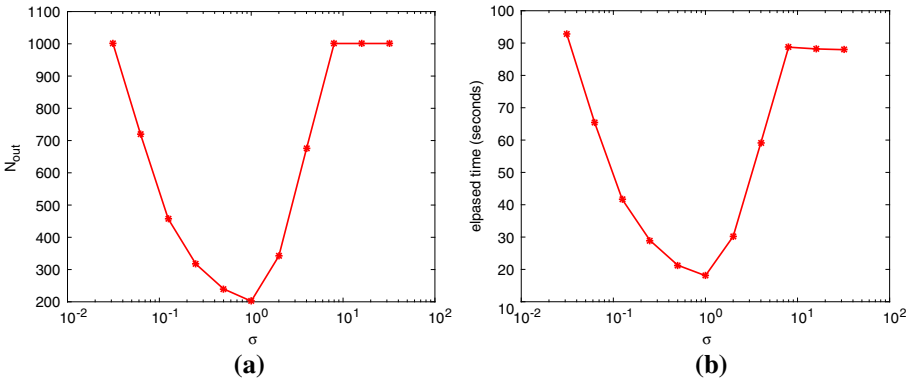


Fig. 11 Performance influenced by parameter σ w.r.t. iteration number. **(a)** N_{out} and **(b)** elapsed time with respect to the iteration number

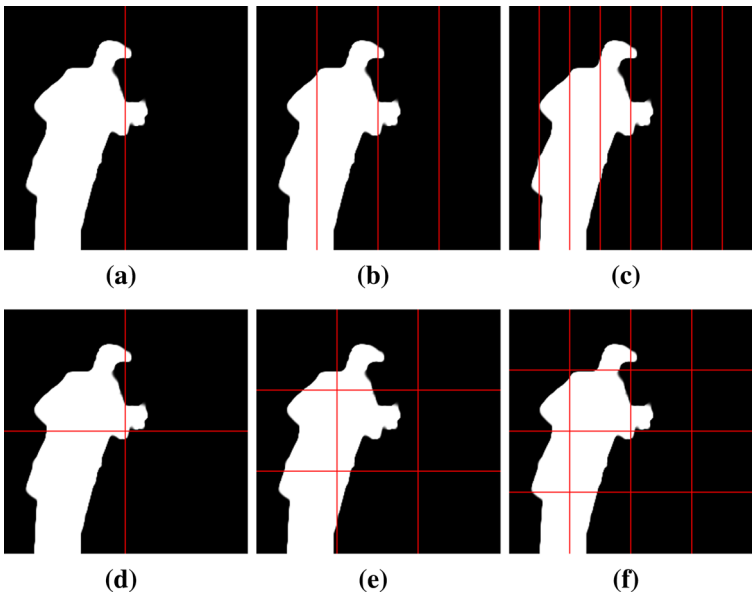


Fig. 12 Performance of the stripe-type and checkerboard-type domain decomposition with the updating means. From *left* to *right* on the *first row* are the segmentation results of the stripe-type decomposition with 2, 2^2 and 2^3 subdomains while on the *second row* are the segmentation results by the checkerboard-type decomposition with 2^2, 3^2 and 4^2 subdomains

TOL $= 1.0 \times 10^{-5}$. The segmentation results of the stripe-type and checkerboard-type decomposition are displayed in Fig. 12. The updating schemes perform very well since the means of the DDMs always keep the same values for the decomposition with different numbers of subdomains.

4.5 The Large-Scale Image Segmentation

In order to test practical performance of our proposed DDMs, three large-scale images with the resolution 512×512 in Fig. 10a, 1024×1024 in Fig. 10d and 4096×4096 (interpolation

Table 4 Performances of the DDMs with updating means for the image of resolution 512×512

N_{sub}	2^2	4^2	6^2	8^2	10^2
N_{out}	201	201	202	199	201
E_k (1.0×10^{-5})	-4.2929000	-4.292900	-4.292900	-4.292900	-4.292900
Time	25	18	19	21	24

The parameters are $\alpha = 50$ and $E_k \leq -4.292900 \times 10^5$. *Time* denotes the elapsed time in seconds, E_k denotes the objective functional values and N_{out} is the iteration number of Algorithm III

Table 5 Performances of the DDMs with updating means for the image of resolution 1024×1024

N_{sub}	2^2	4^2	6^2	8^2	10^2
N_{out}	345	346	346	346	346
E_k (1.0×10^{-5})	-2.204990	-2.204990	-2.204990	-2.204990	-2.204990
Time	219	174	119	125	130

The parameters are $\alpha = 1$ and $E_k \leq -2.204990 \times 10^5$. *Time* denotes the elapsed time in seconds, E_k denotes the objective functional values and N_{out} is the iteration number of Algorithm III

Table 6 Performances of the DDMs with updating means for the image of resolution 4096×4096

N_{sub}	2^2	4^2	6^2	8^2	10^2
N_{out}	52	52	52	52	53
E_k (1.0×10^{-6})	-3.375788	-3.375603	-3.375066	-3.375066	-3.381588
Time	370	372	373	378	388

The parameters are $\alpha = 1$ and $E_k \leq -3.374890 \times 10^6$ is adopted. *Time* denotes the elapsed time in seconds, E_k denotes the objective functional values and N_{out} is the iteration number of Algorithm III

version of "Cameraman" in Fig. 2b) are employed. A different stopping condition $E_k \leq \mathrm{TOL_e}$ is used as [51,53], which is defined to guarantee the quality of segmentation visually. We set $\mathrm{TOL_e} = -4.292900 \times 10^5, -2.204990 \times 10^5, -3.374890 \times 10^6$, which are the numerical energy of the iterative solutions by the primal–dual algorithm without the DDMs for the three images, respectively. It is very expensive for the computation on the large-scale images and we set $N_{in} = 2$ in this experiment. We separately display the results in Tables 4, 5, and 6. As the number of subdomains increases, the iteration numbers N_{out} vary very slightly. It means the DDMs are rather robust w.r.t. the number of subdomains when segmenting the large-scale images. Moreover, inferred from Tables 4 and 5, the elapsed time first decreases and increases as the number of subdomains increases. It implies that one should adopt the DDMs with a suitable number of subdomains to gain the best performances in practise.

4.6 Comparison with the Original Algorithm by the Level Set Method

In this subsection, our proposed DDMs with the updating means are compared with the original algorithm solved by the level set method [10]. We set $\alpha = 0.6$. In order to give a fair comparison, these two algorithms stop at the same time if $t_{DDM}, t_{LS} \geq 40$ in seconds, where t_{DDM}, t_{LS} denote the elapsed time by the checkerboard-type domain decomposition

Table 7 The RAM and numerical energy values E_k when $t_{DDM}, t_{LS} \geq 40$ s

Methods	Memories (MB)	E_k
DDM	13.63	-2.732757×10^3
LS	11.25	-2.705405×10^3

with 4 subdomains and the level set based method, respectively. The package "TVREG" [58] is used for the level set method. Both the RAM used and the corresponding energy values are reported in Table 7. It is observed that the objective functional value obtained by the DDMs is smaller than the one by the level set method. On the other hand, the RAM used by the DDMs is a little bit bigger than the one by the level set method. It is because the current DDMs is implemented with a sequential manner. Both the memory cost and CPU time will be greatly reduced when a parallel pattern is realized (It is shown that the speed-up efficiency reported could reach 0.8 even for nonlocal total variation based restoration problems by the DDMs in [59]).

5 Conclusion

It is very important to investigate the convergence for the CCV model with the updating schemes. We will further investigate the multi-phase segmentation problems in the future. In order to accelerate the computation speed, the DDMs for the continuous max-flow [12,60] and the min-cut model [40,41] will be considered as well.

Appendix

There are many solvers to minimize (2.1) including the primal–dual algorithm [57]. The saddle point problem of (2.1) is built up as follows:

$$\max_{\boldsymbol{p}} \min_{h} \int_{\Omega} \nabla h \cdot \boldsymbol{p} + \alpha \int_{\Omega} sh - I_{K_p}(\boldsymbol{p}) + I_{K_h}(h), \tag{5.1}$$

where

$$I_S(h) = \begin{cases} 0, & h \in S, \quad (5.2) \\ +\infty, & h \notin S. \quad (5.2') \end{cases}$$

By defining $G_1(h) = \alpha \int_{\Omega} sh + I_{K_h}(h)$ and $G_2(\boldsymbol{p}) = I_{K_p}(\boldsymbol{p})$, Eq. (5.1) reduces to the following form:

$$\max_{\boldsymbol{p}} \min_{h} \int_{\Omega} \nabla h \cdot \boldsymbol{p} + G_1(h) - G_2(\boldsymbol{p}).$$

The primal–dual algorithm by Chambolle and Pock [31] can be used to solve the above problem.

Primal–Dual algorithm for (1.2)

1. Initialization: choose $\boldsymbol{p}^0 = 0, \bar{h}^0 = h^0 = 0$, and select parameters $\tau, \sigma (\tau\sigma < 1/8)$ and $\theta \in [0, 1]$.

2. Iterations for $n \geq 0$: Update h^n, $\boldsymbol{p}^n$, $\bar{h}^n$ as follows:

$$\begin{cases} \boldsymbol{p}^{n+1} = \arg\min_{\boldsymbol{p}} G_2(\boldsymbol{p}) + \dfrac{1}{2\sigma}\int_\Omega |\boldsymbol{p} - \hat{\boldsymbol{p}}^{\boldsymbol{n}}|^2, \quad \text{with } \hat{\boldsymbol{p}}^{\boldsymbol{n}} = \boldsymbol{p}^n + \sigma\nabla\bar{h}^n, \\ h^{n+1} = \arg\min_h G_1(h) + \dfrac{1}{2\tau}\int_\Omega |h - \hat{h}^n|^2, \quad \text{with } \hat{h}^n = h^n + \tau\,\mathrm{div}\,\boldsymbol{p}^{n+1}, \\ \bar{h}^{n+1} = (1+\theta)h^{n+1} - \theta h^n. \end{cases} \tag{5.3}$$

End till some stopping criterion meets.

Both subproblems of $\boldsymbol{p}$ and h in Eq. (5.3) have the closed-form forms, which are

$$\boldsymbol{p}^{n+1}(x) = \frac{\hat{\boldsymbol{p}}^{\boldsymbol{n}}(x)}{\max\{1, |\hat{\boldsymbol{p}}^{\boldsymbol{n}}(x)|\}},$$

and

$$h^{n+1}(x) = \min\left\{1, \max\{0, \hat{h}^n(x) - \tau\alpha s(x)\}\right\}, \quad \forall x \in \Omega.$$

Theorem 5.1 ([31]) *The primal–dual algorithm is convergent if* $\tau\sigma < 1/8$.

References

1. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision. Cengage Learning, Boston (2014)
2. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. Int. J. Comput. Vis. **1**(4), 321–331 (1988)
3. Caselles, V., Catté, F., Coll, T., Dibos, F.: A geometric model for active contours in image processing. Numer. Math. **66**(1), 1–31 (1993)
4. Malladi, R., Sethian, J.A., Vemuri, B.C.: Shape modeling with front propagation: a level set approach. IEEE Trans. Pattern Anal. Mach. Intell. **17**(2), 158–175 (1995)
5. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int. J. Comput. Vis. **22**(1), 61–79 (1997)
6. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Commun. Pure Appl. Math. **42**(5), 577–685 (1989)
7. Chan, T.F., Vese, L.A.: Active contours without edges. IEEE Trans. Image Process. **10**(2), 266–277 (2001)
8. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. J. Comput. Phys. **79**(1), 12–49 (1988)
9. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation noise removal algorithm. Phys. D **60**, 259–268 (1992)
10. Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. SIAM J. Appl. Math. **66**(5), 1632–1648 (2006)
11. Vese, L.A., Chan, T.F.: A multiphase level set framework for image segmentation using the Mumford and Shah model. Int. J. Comput. Vis. **50**(3), 271–293 (2002)
12. Bae, E., Yuan, J., Tai, X.-C.: Global minimization for continuous multiphase partitioning problems using a dual approach. Int. J. Comput. Vis. **92**, 112–129 (2011)
13. Brown, E., Chan, T., Bresson, X.: Completely convex formulation of the Chan–Vese image segmentation model. Int. J. Comput. Vis. **98**, 103–121 (2012)
14. Chambolle, A., Cremers, D., Pock, T.: A convex approach to minimal partitions. SIAM J. Imaging Sci. **5**, 1113–1158 (2012)
15. Scherzer, O. (ed.): Handbook of Mathematical Methods in Imaging. Springer, New York (2011)
16. Acar, R., Vogel, C.: Analysis of bounded variation penalty methods for ill-posed problems. Inverse Probl. **10**, 1217–1230 (1994)
17. Marquina, A., Osher, S.: Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal. SIAM J. Sci. Comput. **22**, 387–405 (2000)

18. Vogel, C.: A multigrid method for total variation-based image denoising. In: Bowers, K.L., Lund, J. (eds.) Computation and Control IV, Progress in Systems and Control Theory, vol. 20, pp. 323–331. Birkhäuser, Boston (1995)
19. Vogel, C.: Computational Methods for Inverse Problems. SIAM, Philadelphia (2002)
20. Vogel, C., Oman, M.: Iterative methods for total variation denoising. SIAM J. Sci. Comput. **17**, 227–238 (1996)
21. Vogel, C.: Fast, robust total variation-based reconstruction of noisy, blurred images. IEEE Trans. Image Process. **7**, 813–824 (1998)
22. Goldstein, T., Osher, S.: The split Bregman method for l^1-regularized problems. SIAM J. Imaging Sci. **2**, 323–343 (2009)
23. Wang, Y., Yang, J., Yin, W., Zhang, Y.: A new alternating minimization algorithm for total variation image reconstruction. SIAM J. Imaging Sci. **1**, 248–272 (2008)
24. Glowinski, R., Tallec, P.: Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics. SIAM, Philadelphia (1989)
25. Wu, C., Tai, X.-C.: Augmented Lagrangian method, dual methods and split-Bregman iterations for ROF, vectorial TV and higher order models. SIAM J. Imaging Sci. **3**, 300–339 (2010)
26. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends® Mach. Learn. **3**(1), 1–122 (2011)
27. Chan, R.H., Tao, M., Yuan, X.: Constrained total variational deblurring models and fast algorithms based on alternating direction method of multipliers. SIAM J. Imaging Sci. **6**(1), 680–697 (2013)
28. Chambolle, A.: An algorithm for total variation minimization and applications. J. Math. Imaging Vis. **20**(1–2), 89–97 (2004)
29. Appleton, B., Talbot, H.: Globally optimal geodesic active contours. J. Math. Imaging Vis. **23**, 67–86 (2005)
30. Arrow, K., Hurwicz, L., Uzawa, H.: Studies in linear and non-linear programming, with contributions by H. B. Chenery, S. M. Johnson, S. Karlin, T. Marschak, R. M. Solow. Stanford Mathematical Studies in the Social Sciences, vol. II. Stanford University Press, Stanford (1958)
31. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vis. **40**(1), 120–145 (2011)
32. Zhu, M., Chan, T.: An efficient primal-dual hybrid gradient algorithm for total variation image restoration. UCLA, pp. 08–34. Center for Applied Math, CAM Reports No (2008)
33. He, B., Yuan, X.: Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. SIAM J. Imaging Sci. **5**, 119–149 (2012)
34. Tai, X.-C.: Rate of convergence for some constraint decomposition methods for nonlinear variational inequalities. Numer. Math. **93**, 755–786 (2003)
35. Tai, X.-C., Espedal, M.: Applications of a space decomposition method to linear and nonlinear elliptic problems. Numer. Methods Partial Differ. Equ. **14**, 717–737 (1998)
36. Tai, X.-C., Espedal, M.: Rate of convergence of some space decomposition methods for linear and nonlinear problems. SIAM J. Numer. Anal. **35**, 1558–1570 (1998)
37. Tai, X.-C., Xu, J.: Global and uniform convergence of subspace correction methods for some convex optimization problems. Math. Comput. **71**, 105–124 (2002)
38. Firsov, D., Lui, S.H.: Domain decomposition methods in image denoising using Gaussian curvature. J. Comput. Appl. Math. **193**, 460–473 (2006)
39. Müller, J.: Parallel total variation minimization. University of Muenster, Diploma Thesis (2008)
40. Duan, Y., Tai, X.C.: Domain decomposition methods with graph cuts algorithms for total variation minimization. Adv. Comput. Math. **36**, 175–199 (2012)
41. Tai, X.-C., Duan, Y.: Domain decomposition methods with graph cuts algorithms for image segmentation. Int. J. Numer. Anal. Model. **8**, 137–155 (2011)
42. Xu, J., Tai, X.-C., Wang, L.-L.: A two-level domain decomposition method for image restoration. Inverse Probl. Imaging **4**, 523–545 (2010)
43. Xu, J., Chang, H., Qin, J.: Domain decomposition method for image deblurring. J. Comput. Appl. Math. **271**, 401–414 (2014)
44. Chang, H., Zhang, X., Tai, X.-C., Yang, D.: Domain decomposition methods for nonlocal total variation image restoration. J. Sci. Comput. **60**, 79–100 (2014)
45. Chang, H., Tai, X.-C., Yang, D.: Domain decomposition methods for total variation minimization. In: Tai, X.-C., et al. (eds.) Energy Minimization Methods in Computer Vision and Pattern Recognition, vol. 8932. LNCS, Springer, Berlin (2015)
46. Chang, H., Tai, X.-C., Wang, L.-L., Yang, D.: Convergence rate of overlapping domain decomposition methods for the Rudin–Osher–Fatami model based on a dual formulation. SIAM J. Imaging Sci. **8**, 564–591 (2015)

47. Fornasier, M., Schönlieb, C.: Subspace correction methods for total variation and l_1 minimization. SIAM J. Numer. Anal. **47**, 3397–3428 (2009)
48. Fornasier, M., Langer, A., Schönlieb, C.: A convergent overlapping domain decomposition method for total variation minimization. Numer. Math. **116**, 645–685 (2010)
49. Hintermüller, M., Langer, A.: Subspace correction methods for a class of nonsmooth and nonadditive convex variational problems with mixed l^1/l^2 data-fidelity in image processing. SIAM J. Imaging Sci. **6**, 2134–2173 (2013)
50. Langer, A., Osher, S., Schonlieb, C.: Bregmanized domain decomposition for image restoration. J. Sci. Comput. **54**, 549–576 (2013)
51. Hintermüller, M., Langer, A.: Non-overlapping domain decomposition methods for dual total variation based image denoising. J. Sci. Comput. **62**, 456–481 (2015)
52. Lee, C., Lee, J., Woo, H., Yun, S.: Block decomposition methods for total variation by primal-dual stitching. J. Sci. Comput. (2015). doi:10.1007/s10915-015-0138-9
53. Chang, H., Tai, X.-C., Wang, L.-L., Yang, D.: Convergence rate of overlapping domain decomposition methods for the Rudin–Osher–Fatami model based on a dual formulation. SIAM J. Imaging Sci. **8**(1), 564–591 (2015)
54. Kunisch, K., Tai, X.-C.: Nonoverlapping domain decomposition methods for inverse problems. In: Bjorstard, P., Espedal, M., Keyes, D. (eds.) Proceedings of 9th International Conference on Domain Decompostion Methods. Wiley (1997)
55. Boyd, S., Lin, X., Almir, M., Jacob, M.: Notes on Decomposition Methods. Notes for EE364B. Stanford University, Stanford (2007)
56. Lions, P.: On Schwarz Alternating Method III: A Variant for Nonoverlapping Subdomains. SIAM, Philadelphia (1990)
57. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vis. **40**(1), 120–145 (2011)
58. Pascal, G.: tvreg v2: variational imaging methods for denoising, deconvolution, inpainting, and segmentation (2010)
59. Chang, H., Zhang, X., Tai, X.-C., Yang, D.: Domain decomposition methods for nonlocal total variation image restoration. J. Sci. Comput. **60**(1), 79–100 (2014)
60. Wei, K., Tai, X.-C., Chan, T.F., Leung, S.: Primal-dual method for continuous max-flow approaches, vol. pp. 15–67. UCLA CAM Report (2015)