Institute for Computational Mathematics
Hong Kong Baptist University

ICM Research Report
08-02

NUMERICAL METHODS FOR INTERACTIVE MULTIPLE CLASS IMAGE SEGMENTATION PROBLEMS

MICHAEL K. NG*, GUOPING QIU†, AND ANDY M. YIP‡

Abstract. In this paper, we consider a bilaterally constrained optimization model arising from the semi-supervised multiple class image segmentation problem. We prove that the solution of the corresponding unconstrained problem satisfies a discrete maximum principle. This implies that the bilateral constraints are satisfied automatically and that the solution is unique. The structures of coefficient matrices arising from the optimality conditions of the segmentation problem are different for different input images, we still show that they are M-matrices in general. Therefore we study several numerical methods for solving such linear systems and demonstrate that domain decomposition with block relaxation methods are quite effective and outperform other tested methods. We also carry out a numerical study of condition numbers on the effect of boundary conditions on the optimization problems which provides some insights into the specification of boundary conditions as an input knowledge in the learning context.

Key words. Image segmentation, discrete maximum principle, domain decomposition, M-matrix, condition numbers, boundary conditions

1. Introduction. Foreground-background segmentation has wide applications in computer vision (e.g. scene analysis), computer graphics (e.g. image editing) and medical imaging (e.g. organ segmentation). Fully automatic image segmentation has many intrinsic difficulties and is still a very hard problem. In many applications, such as image editing in computer graphics and organ segmentation in medical imaging, semi-automatic and interactive approaches, where human operators provide strong priors for the computational algorithms to perform segmentation, cannot only overcome the inherent technical difficulties of fully automatic image segmentation, but may also be desirable because the segmentation process and results can be controlled in many of these applications. Due to their desirable properties, there have been increasing activities in the research community to develop interactive semi-automatic image segmentation techniques [1, 2, 3, 4, 5].

Recently, Guan and Qiu [5] have developed an optimization based segmentation technique, where a transparency image was computed by optimizing a quadratic cost function with user supplied linear constraints. For multiple class image segmentation problem [6], a cost function that is similar to that of the case of figure and ground segmentation can also be formulated. This approach formulates multiple class segmentation in a single cost function and a solution to this cost function achieves multiple class segmentation in a single step. However, optimizing the multiple class cost function is computationally much more challenging [6]. The main aim of this paper is to study numerical methods for such bilaterally constrained optimization model arising from the semi-supervised multiple class image segmentation problem. We prove that the solution of the corresponding unconstrained problem satisfies a discrete maximum principle. This implies that the bilateral constraints are satisfied automatically and that the solution is unique.

The main issue of this multiple class image segmentation model is to require to

*Centre for Mathematical Imaging and Vision and Institute for Computational Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong (mng@math.hkbu.edu.hk). Research supported in part by Hong Kong Research Grants Council Grant Number 201508 and HKBU FRGs.

†School of Computer Science, University of Nottingham, United Kingdom (qiu@cs.nott.ac.uk).

‡Department of Mathematics, National University of Singapore, 2, Science Drive 2, S117543, Singapore (andyyip@nus.edu.sg).

solve linear systems where the structures of these coefficient matrices do not have specific patterns. The construction of coefficient matrices depends on input images, we show that they are M-matrices in general. Therefore we study several numerical methods for solving such linear systems and demonstrate that domain decomposition with block relaxation methods are quite effective and outperform other tested methods.

On the other hand, an important aspect of this and other similar interactive approaches is that users have to supply constraints (manually labeled pixels). These user supplied constraints greatly affect the segmentation results. However, in the literature, there is no theory to guide users to label the most suitable pixels. We carry out a numerical study on the effect of boundary conditions on the optimization problems which provides some insights into the specification of boundary conditions as an input knowledge in the learning context; and such insights can provide guidance for the users to label suitable pixels.

The outline of this paper is given as follows. In Section 2, we study multiple class image segmentation problems. In Section 3, we analyze the linear system arising from the problem. In Section 4, we study numerical solvers for this linear system. In Section 5, numerical results are presented to illustrate the performance of numerical solvers. We also study the effect of boundary conditions in image segmentation problems. Finally, concluding remarks are given in Section 6.

2. Image Segmentation Problems.

2.1. An Optimization Model for Two-class Problems. The observed image $u = (u_i)$ is modeled as a convex combination of a foreground $F = (F_i)$ and a background $B = (B_i)$:

$$u_i = \alpha_i F_i + (1 - \alpha_i) B_i.$$

Here i denotes a linear order of the pixel location and $\alpha = (\alpha_i)$ with $0 \leq \alpha_i \leq 1$ for each i indicates the degree of membership of each pixel to the foreground. A foreground and a background in an image are correspondingly two classes. When $\alpha_i = 1$, the i-th pixel is a certain foreground pixel. When $\alpha_i = 0$, the pixel is a certain background pixel. In image segmentation, the problem is to estimate the membership function $\{\alpha_i\}$ from the given image $\{u_i\}$ and some sample pixels in which $u_i = F_i$ or $u_i = B_i$.

In the segmentation model, membership functions $\{\alpha_i\}$ are based on the similarity of both geometric and photometric neighbors. They define the geometric neighborhood of pixel i as:

$$N_i^g := \{j \in \mathbb{Z}_+^2 : \ 0 < \|i - j\| \leq r_g\}$$

where $r_g > 0$ is a constant controlling the size of the neighborhood and $\|\cdot\|$ is a vector norm. For example, we often use a 3×3 window around a pixel as its geometric neighborhood (excluding pixel i itself). This corresponds to $r_g = 1$ and $\|\cdot\| = \|\cdot\|_\infty$. Let G_i be a feature vector (e.g. color, texture, etc.) computed around pixel i. We call G_i a photometric feature. Then, a natural way to define photometric neighborhood N_i^p of the pixel i is pixels whose photometric features are close to G_i. But this may lead to a large neighborhood. Moreover, these neighbors are time-consuming to compute. Instead, we can define the photometric neighborhood N_i^p of i to be pixels that are the top k photometrically closest (in Euclidean norm) pixels to i within a window around i (excluding i itself). In our experiments, each feature vector has 27

entries which are the RGB values of the 3×3 window around i, the window size is set to 17×17 and the top 4 neighbors are used, see for instance [5].

The neighbor N_i is defined to be $N_i^p \cup N_i^g$ which will be used to define membership functions. The geometric neighbor is a symmetric relation, i.e. $j \in N_i^g$ if and only if $i \in N_j^g$. However, photometric neighbor, and hence neighbor, is non-symmetric since it is defined based on k nearest neighbors.

The basic idea of the optimization model is that the membership (to the foreground) of neighboring pixels (either geometric or photometric) should be similar. More precisely, the optimization model for the two-class segmentation problem is given by:

$$\min_{\alpha} \sum_{i \in \Omega} \left[\alpha_i - \sum_{j \in N_i} (w_{ij}^g + \lambda w_{ij}^p) \alpha_j \right]^2$$

subject to

$$0 \le \alpha_i \le 1, \qquad \text{for all } i$$

and the boundary conditions

$$\alpha_i = \begin{cases} 1, & \text{on } \mathcal{F} \text{ (Foreground)}, \\ 0, & \text{on } \mathcal{B} \text{ (Background)}. \end{cases} \tag{2.1}$$

Here, $w_{ij}^g \ge 0$ is the geometric similarity score between pixel i and pixel j, $w_{ij}^p \ge 0$ is the photometric similarity score between pixel i and pixel j, $\lambda \ge 0$ is a user specified parameter controlling the relative importance of the geometric and photometric similarity scores. The sets $\mathcal{F}$ and $\mathcal{B}$ are pixels that are certain foreground and certain background respectively, supplied by the user interactively. These boundary conditions ensure the uniqueness of solution to the problem. We will prove such property in the next section. The set Ω consists of pixels at which the membership is unknown, and its complement Γ is equal to $\mathcal{F} \cup \mathcal{B}$.

In the model, the similarity scores w_{ij}^g and w_{ij}^p are defined by:

$$w_{ij}^g := \begin{cases} c_i^g e^{-\|i-j\|_2^2/\sigma_g^2}, & \text{if } j \in N_i^g, \\ 0, & \text{otherwise}, \end{cases}$$

$$w_{ij}^p := \begin{cases} c_i^p e^{-\|G_i - G_j\|_2^2/\sigma_p^2}, & \text{if } j \in N_i^p, \\ 0, & \text{otherwise}, \end{cases}$$

where σ_g^2 is the variance of the geometric locations within N_i^g, σ_p^2 is the variance of the photometric features within N_i^p, c_i^g and c_i^p are normalizing constants so that $\sum_j w_{ij}^g = \sum_j w_{ij}^p = \frac{1}{1+\lambda}$. It can be easily seen that w_{ij}^g (respectively w_{ij}^p) measures the similarity between the geometric (respectively photometric) features of pixel i and pixel j.

Let $w_{ij} := w_{ij}^g + \lambda w_{ij}^p$. Then we have

$$\sum_j w_{ij} \equiv 1 \qquad \forall i. \tag{2.2}$$

We note that w_{ij} is the relative combined geometric and photometric similarity measures of pixel i to other neighboring pixels.

Using the lexicographical ordering of the pixels, we can express the above optimization in matrix-vector notations:

$$\min_{\mathbf{0}\leq\alpha\leq\mathbf{1}} \|D_\Omega(I-W)\alpha\|_2^2 \tag{2.3}$$

subject to the boundary conditions (2.1). Here, $\mathbf{0}$ and $\mathbf{1}$ are vectors with all entries 0 and all entries 1 respectively, I is the identity matrix, $W=(w_{ij})$ is an $N\times N$ matrix recording the weights, $\alpha=(\alpha_i)$ is an N vector representing the membership function, N is the number of pixels and D_Ω is an $|\Omega|\times N$ downsampling matrix from the image domain to Ω. The inequality constraints are understood entry-wise. The matrix W is non-symmetric because each pair of pixels have different sets of geometric and photometric neighbors, therefore, the normalization constants c_i^g and c_i^p are different for different pixels.

In [5], Guan and Qiu have presented experimental results on this model and showed that it is very effective in estimating the membership function from the image. Moreover, they showed that the use of both geometric and photometric neighbors is very important for its success. Furthermore, the image segmentation problem is often ambiguous. Therefore, they resolved this problem by requesting the user to supply pixels that are certain background and certain foreground.

2.2. An Optimization Model for Multiple-class Problems. In this subsection, we generalize the two-class model to the case of multiple-class image segmentation. This multi-class model allows us to handle images with multiple segments. Now the image is modeled as a convex combination of M images:

$$u_i=\alpha_i^1F_i^1+\alpha_i^2F_i^2+\ldots+\alpha_i^MF_i^M.$$

Here, the M membership functions satisfy

$$0\leq\alpha_i^m\leq 1$$

for each m and each i and

$$\sum_{m=1}^{M}\alpha_i^m=1$$

for each i. In the multiple-class setting, the user supplies a set of M regions

$$\Gamma:=\Gamma^1\cup\Gamma^2\cup\cdots\cup\Gamma^M$$

where Γ^m is the set of pixels with certain membership to the m-th class. We assume that Γ is non-empty but each individual region Γ^m can be empty. We further assume that each pixel in Γ is a neighbor of some pixel in Ω. If the user specifies a large region which contains pixels that are not neighbor of Ω, then such pixels have no effect on the optimization problem. Therefore, we can safely ignore them.

The optimization model for estimating the membership functions is given as follows:

$$\min_{\alpha^1,\ldots,\alpha^M}\sum_{m=1}^{M}\|D_\Omega(I-W)\alpha^m\|_2^2 \tag{2.4}$$

subject to $\mathbf{0} \le \alpha^m \le \mathbf{1}$, $\sum_m \alpha^m = \mathbf{1}$ and the boundary conditions

$$\alpha_i^m = \begin{cases} 1, & \text{on } \Gamma^m, \\ 0, & \text{on } \Gamma \setminus \Gamma^m, \end{cases} \tag{2.5}$$

for $m = 1, 2, \ldots, M$. Similar to the two-class case, Ω is the set of pixels at which the membership functions are unknown (i.e. the complement of Γ) and D_Ω is the downsampling matrix from the image domain to Ω. When $M = 2$, this model can be shown to reduce to the two-class model in (2.3).

We observe that the objective and the constraints are convex. Moreover, the feasible set is compact. Therefore, a global minimizer exists. We will show in the next subsection that the solution is unique, see (2.9) and Corollary 3.3.

2.3. Optimality Conditions. For the sake of completeness, we derive in this subsection the optimality conditions for the multiple-class model (2.4). Indeed, we show that the optimization can be simplified so that the optimal membership functions $\{\alpha^m\}$ are computed independently. In contrast, Guan and Qiu [5] solved all the membership functions simultaneously which requires a huge computational cost, see the remarks at the end of this subsection. We here ignore the bilateral constraints $\mathbf{0} \le \alpha^m \le \mathbf{1}$. We will show in the next section that an explicit imposition of such constraints is indeed unnecessary, see Corollary 3.2.

Let us define some notations. Let $A = I - W$. Without loss of generality, we assume that the pixels are ordered so that A and α^m can be partitioned as follows:

$$A = \begin{bmatrix} A_{\Omega\Omega} & A_{\Omega\Gamma} \\ A_{\Gamma\Omega} & A_{\Gamma\Gamma} \end{bmatrix} \qquad \text{and} \qquad \alpha^m = \begin{bmatrix} \alpha_\Omega^m \\ \alpha_\Gamma^m \end{bmatrix}.$$

In fact, we have $A_{\Omega\Omega} = D_\Omega A D_\Omega^T$, $A_{\Omega\Gamma} = D_\Omega A D_\Gamma^T$ and $\alpha_\Omega^m = D_\Omega \alpha^m$ for $m = 1, 2, \cdots, M$. Other variables with subscripts Ω or Γ are defined similarly.

The Lagrangian for (2.4) is given by

$$L(\alpha, \mu, \eta) = \frac{1}{2} \sum_{m=1}^{M} \|D_\Omega A \alpha^m\|_2^2 - \langle \sum_{m=1}^{M} \alpha_\Omega^m - \mathbf{1}, \mu_\Omega \rangle - \sum_{m=1}^{M} \langle \alpha_\Gamma^m - \bar{\alpha}_\Gamma^m, \eta_\Gamma^m \rangle$$

where $\bar{\alpha}_\Gamma^m$ are the given boundary data on Γ, $\mu_\Omega = (\mu_i)$ are the Lagrange multipliers for the constraints $\sum_m \alpha_\Omega^m = \mathbf{1}$ and $\eta_\Gamma^m = (\eta_i^m)$ are the Lagrange multipliers for the constraints $\alpha_\Gamma^m = \bar{\alpha}_\Gamma^m$. We have included the constant $1/2$ in the first term for convenience.

The optimality conditions are given by

$$\frac{\partial L}{\partial \alpha^m} = \mathbf{0} \quad (m = 1, 2, \cdots, M) \quad \text{and} \quad \frac{\partial L}{\partial \mu} = \frac{\partial L}{\partial \eta} = \mathbf{0}.$$

We will examine only the equation $\frac{\partial L}{\partial \alpha^m} = \mathbf{0}$ for the others just correspond to the equality constraints. We have

$$\frac{\partial L}{\partial \alpha^m} = A^T D_\Omega^T D_\Omega A \alpha^m - \begin{bmatrix} \mu_\Omega \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \eta_\Gamma^m \end{bmatrix} = \mathbf{0}, \quad m = 1, 2, \cdots, M. \tag{2.6}$$

By summing the above equation for all m and using the constraints $\sum_m \alpha^m = \mathbf{1}$, we have

$$A^T D_\Omega^T D_\Omega A \mathbf{1} = \begin{bmatrix} M\mu_\Omega \\ \sum_{m=1}^{M} \eta_\Gamma^m \end{bmatrix}, \quad m = 1, 2, \cdots, M.$$

Notice that $A\mathbf{1} = (I - W)\mathbf{1} = \mathbf{1} - W\mathbf{1} = \mathbf{0}$ because of (2.2). Hence, we have $\mu_\Omega = \mathbf{0}$ and $\sum_{m=1}^{M} \eta_\Gamma^m = \mathbf{0}$. Equation (2.6) can now be written as:

$$A^T D_\Omega^T D_\Omega A \alpha^m = \begin{bmatrix} \mathbf{0} \\ \eta_\Gamma^m \end{bmatrix}, \quad m = 1, 2, \cdots, M.$$

Using the partition of A and α^m, we have

$$\begin{bmatrix} A_{\Omega\Omega}^T A_{\Omega\Omega} & A_{\Omega\Omega}^T A_{\Omega\Gamma} \\ A_{\Omega\Gamma}^T A_{\Omega\Omega} & A_{\Omega\Gamma}^T A_{\Omega\Gamma} \end{bmatrix} \begin{bmatrix} \alpha_\Omega^m \\ \alpha_\Gamma^m \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \eta_\Gamma^m \end{bmatrix}, \quad m = 1, 2, \cdots, M.$$

We will show in Corollary 3.4 that $A_{\Omega\Omega}$ is non-singular. We also notice that the boundary conditions are $\alpha_\Gamma^m = \bar{\alpha}_\Gamma^m$. Thus, the optimality conditions can be reduced to

$$A_{\Omega\Omega}\alpha_\Omega^m + A_{\Omega\Gamma}\alpha_\Gamma^m = \mathbf{0}, \tag{2.7}$$

$$\alpha_\Gamma^m = \bar{\alpha}_\Gamma^m. \tag{2.8}$$

Finally, we write the conditions as

$$\tilde{A}\alpha^m = \mathbf{b}^m, \quad m = 1, 2, \cdots, M. \tag{2.9}$$

where

$$\tilde{A} = \begin{bmatrix} A_{\Omega\Omega} & A_{\Omega\Gamma} \\ 0 & I \end{bmatrix} \tag{2.10}$$

and

$$\mathbf{b}^m = \begin{bmatrix} \mathbf{0} \\ \bar{\alpha}_\Gamma^m \end{bmatrix}.$$

Here there are several remarks for the above equations.

1. $\tilde{A}$ is defined by replacing a_{ij} with the Kronecker Delta δ_{ij} if the pixel i is in Γ.
2. The Lagrange multipliers η_Γ^m need not to be computed in order to obtain the optimal membership functions. But once the optimal membership functions are obtained, we can obtain η_Γ^m by $A_{\Omega\Gamma}^T A_{\Omega\Omega}\alpha_\Omega^m + A_{\Omega\Gamma}^T A_{\Omega\Gamma}\alpha_\Gamma^m$.
3. The optimal membership functions can be computed independently for each m. Therefore, reducing the computational complexity compared to the approach of Guan and Qiu which solves the least squares problem:

$$\min_{\alpha^1,\ldots,\alpha^M} \left\| \begin{bmatrix} \tilde{A} & & & \\ & \tilde{A} & & \\ & & \ddots & \\ & & & \tilde{A} \\ I & I & \cdots & I \end{bmatrix} \begin{bmatrix} \alpha^1 \\ \alpha^2 \\ \vdots \\ \alpha^M \end{bmatrix} - \begin{bmatrix} \mathbf{b}^1 \\ \mathbf{b}^2 \\ \vdots \\ \mathbf{b}^M \\ \mathbf{1} \end{bmatrix} \right\|_2 .$$

   It can easily verified that this least squares problem is equivalent to the original problem. But solving such a least square problem directly requires solving a system of equations which is M times larger in size.
4. The coefficient matrix $\tilde{A}$ is independent of m, therefore, solution methods for multiple right hand side can be explored.
5. Some savings in computational cost can be achieved by solving (2.9) for $m = 1, 2, \ldots, M-1$ followed by setting $\alpha^M = \mathbf{1} - \sum_{m=1}^{M-1} \alpha^m$.

3. Properties of the Coefficient Matrix $\tilde{A}$. In this section, we study the properties of the coefficient matrix $\tilde{A}$ in (2.9) and (2.10).

3.1. Discrete Maximum Principle. We first show that the solution of $\tilde{A}\alpha^m = \mathbf{b}^m$ satisfies a discrete maximum principle which implies $\tilde{A}$ is non-singular and the bilateral constraints $\mathbf{0} \le \alpha^m \le \mathbf{1}$ are automatically satisfied.

We first recall the (strong) discrete maximum principle. Let Ω be a discrete connected domain and let Γ be its boundary ($\Omega \cap \Gamma = \emptyset$). Let $\alpha = (\alpha_i)$ be a discrete function defined on $\Omega \cup \Gamma$. Then the principle says that α attains its maximum in Γ only, unless α is a constant everywhere in $\Omega \cup \Gamma$. The minimum principle can be defined similarly. The connectedness of Ω is defined base on the neighbor N_i. Pixel i and pixel j are connected if and only if they are a neighbor of each other. That is, w_{ij} and w_{ji} are both positive. If Ω has more than one connected component, then the principle can be applied to each component independently. Therefore, we can safely assume that Ω is connected. As we mentioned in Section 2.3, we also assume that each point in Γ is a neighbor of some pixel in Ω.

Next, we show that the optimal membership functions satisfy the maximum principle. The minimum principle can be proved similarly.

THEOREM 3.1. *The solution to $\tilde{A}\alpha = \mathbf{b}$ satisfies the discrete maximum principle. Here, $\mathbf{b} = (b_i)$ is a vector of the form*

$$\mathbf{b} = \begin{bmatrix} \mathbf{0} \\ \mathbf{b}_\Gamma \end{bmatrix}.$$

Proof. We assume that the maximum is attained at an interior point $i_0 \in \Omega$. Then, $(\tilde{A}\alpha)_{i_0} = 0$ because $b_{i_0} = 0$. Moreover, the i_0-th row of $\tilde{A}$ is the same as the i_0-th row of $A = I - W$. Thus,

$$(\tilde{A}\alpha)_{i_0} = \alpha_{i_0} - \sum_{j \in N_{i_0}} w_{i_0 j}\alpha_j = 0$$

or

$$\alpha_{i_0} = \sum_{j \in N_{i_0}} w_{i_0 j}\alpha_j.$$

Since $w_{i_0 j} > 0$ for $j \in N_{i_0}$ and $\sum_{j \in N_{i_0}} w_{i_0 j} = 1$, the above equation says that the maximum value α_{i_0} equals to a weighted average of $\{\alpha_j : \ j \in N_{i_0}\}$. This implies that $\alpha_{i_0} = \alpha_j$ for all $j \in N_{i_0}$. For each $j \in N_{i_0}$, α_j is also the maximum so that the same argument can be applied to conclude that the value of α on N_j is again the maximum. Since the domain Ω is connected, we can conclude that α attains its maximum at every point in Ω and in the neighbor of Ω which covers Γ. This shows that if α has an interior maximum, then α is constant everywhere in $\Omega \cup \Gamma$. □

In particular, the maximum (and minimum) principle implies that the strict bilateral constraints $\mathbf{0} < \alpha < \mathbf{1}$ and uniqueness of solution. We present these results in the next two corollaries.

COROLLARY 3.2. *The solution to $\tilde{A}\alpha = \mathbf{b}$ automatically satisfies the strict bilateral constraints*

$$\mathbf{0} < \alpha < \mathbf{1}$$

provided that $\Gamma = \{i \in \Gamma : \ b_i = 0\} \cup \{i \in \Gamma : \ b_i = 1\}$ and both sets are non-empty.

Proof. By the maximum principle, we have $\alpha \leq \max_{i\in\Gamma} \alpha_i = 1$ and $\alpha \geq \min_{i\in\Gamma} \alpha_i = 0$. Since α is not constant on Γ, it cannot have any interior maximum nor interior minimum. Therefore, the bilateral constraints are strict in Ω. □

COROLLARY 3.3. *The solution to $\tilde{A}\alpha = \mathbf{b}$ is unique.*

Proof. Let $\mathbf{b} = \mathbf{0}$, i.e. all boundary pixels have zero membership. Then, by the maximum principle, we have $\alpha_\Omega \leq \max_{i\in\Gamma} \alpha_i = 0$ and $\alpha_\Omega \geq \min_{i\in\Gamma} \alpha_i = 0$. Thus, $\alpha = \mathbf{0}$. This shows that the only solution to $\tilde{A}\alpha = \mathbf{0}$ is $\alpha = \mathbf{0}$. Therefore, $\tilde{A}$ is non-singular. □

A direct consequence is that the submatrix $A_{\Omega\Omega}$ of $\tilde{A}$ is also non-singular, a fact we used when deriving the optimality conditions (2.7)–(2.8).

COROLLARY 3.4. *The matrix $A_{\Omega\Omega}$ is non-singular.*

Proof. We note that $\tilde{A}$ is block upper triangular with diagonal blocks $A_{\Omega\Omega}$ and I. Thus, $0 \neq \det(\tilde{A}) = \det(A_{\Omega\Omega})\det(I) = \det(A_{\Omega\Omega})$ where $\det(\cdot)$ denotes the determinant. □

3.2. M-matrix. Next we show that $\tilde{A}$ is an M-matrix. This property will be useful when we design some effective preconditioners for $\tilde{A}$. We recall that a matrix A is called an M-matrix if

1. $a_{ii} > 0$ for all i;
2. $a_{ij} \leq 0$ for all $i \neq j$;
3. A is non-singular;
4. $A^{-1} \geq 0$ (entry-wise).

It can be shown that the third and the fourth conditions can be replaced by $\rho(I - D^{-1}A) < 1$ where $\rho(\cdot)$ denotes the spectral radius (the largest eigenvalue in absolute value) and D is the diagonal part of A, see [7, p.29].

THEOREM 3.5. *The coefficient matrix $\tilde{A}$ is an M-matrix.*

Proof. It is obvious that $\tilde{A} = (\tilde{a}_{ij})$ satisfies $\tilde{a}_{ii} = 1 > 0$ for all i and $\tilde{a}_{ij} = -w_{ij} \leq 0$ for all $i \neq j$. The diagonal part of $\tilde{A}$ is I, and therefore, it remains to show that $\rho(I - \tilde{A}) < 1$.

We partition W into two submatrices:

$$W = \begin{bmatrix} W_\Omega \\ W_\Gamma \end{bmatrix}.$$

Then, we have

$$\tilde{W} := I - \tilde{A} = \begin{bmatrix} W_\Omega \\ \mathbf{0} \end{bmatrix}.$$

We note that the row sum $\tilde{W}\mathbf{1}$ is a binary vector. Moreover, $\tilde{W} \geq 0$ so that $\|\tilde{W}\|_\infty = \|\tilde{W}\mathbf{1}\|_\infty = 1$. Since $\rho(\cdot) \leq \|\cdot\|$ for any matrix norm, $\rho(\tilde{W}) \leq \|\tilde{W}\|_\infty = 1$. To show the strictly inequality, we assume to the contrary that $\rho(\tilde{W}) = 1$. Then $\tilde{W}$ has an eigenvector $\mathbf{x} \neq \mathbf{0}$ and a corresponding eigenvalue ν (complex-valued) with $|\nu| = 1$. So, $\tilde{W}\mathbf{x} = \nu\mathbf{x}$. In particular, $|(\tilde{W}\mathbf{x})_i| = |\nu||x_i| = |x_i|$. If $i \in \Omega$, then $|\sum_j w_{ij}x_j| = |x_i|$ so that the arguments used to prove the maximum principle can be applied to argue that $|x_i| \equiv$ constant for all $i \in \Omega\cup\Gamma$. If $i \in \Gamma$, then we have $(\tilde{W}\mathbf{x})_i = 0$ so that $x_i = 0$. Together with the fact that $|x_i|$ is constant, we have $x_i = 0$ for all $i \in \Omega \cup \Gamma$. This contradicts to $\mathbf{x} \neq \mathbf{0}$. Thus, $\rho(\tilde{W}) < 1$. □

4. Numerical Solvers. We have shown that the multiple-class image segmentation problem amounts to solving systems of linear equations of the form $\tilde{A}\alpha = \mathbf{b}$.

Typically, we are dealing with a very large system as the number of pixels can be very large. We note that $\tilde{A}$ is a sparse matrix because the number of nonzeros in each row depends on the size of geometric and photometric neighbors. Since the nonzero positions are image dependent (i.e., $\tilde{A}$ is not fixed), the computer may run out of memory due to the problem of fill-in for some nonzero patterns of $\tilde{A}$. Direct methods may not be effective. As $\tilde{A}$ is sparse, its matrix-vector multiplication can be computed very efficiently.

Thus we apply iterative methods [7] to solving linear systems arising from the optimality conditions of optimization problems. In particular, we study domain decomposition (DD) methods and the preconditioned GMRES method with domain decomposition preconditioners for solving such systems. We demonstrate that they are quite effective and outperform the other tested iterative methods. We found in our numerical tests that alternative methods such as BiCGSTAB may fail to converge whereas CGS and CGNR are very slow — the relative residual remains larger than 10^{-6} after exceeding the preset limit of 200 iterations.

In the domain decomposition method, we first decompose the image domain $\Omega \cup \Gamma$ into possibly overlapping regions $\Omega_1, \Omega_2, \ldots, \Omega_L$. Let $\alpha^{(k)}$ be the k-th iterate. For each $l = 1, 2, \ldots, L$, we solve the smaller system

$$\tilde{A}_{\Omega_l \Omega_l} \alpha_{\Omega_l}^{(k+1/2)} = \mathbf{b}_{\Omega_l} - \tilde{A}_{\Omega_l \Omega_l'} \alpha_{\Omega_l'}^{(k)}$$

for $\alpha_{\Omega_l}^{(k+1/2)}$. Here Ω_l' is the complement of Ω_l and $\tilde{A}$ is partitioned into

$$\tilde{A} = \begin{bmatrix} \tilde{A}_{\Omega_l \Omega_l} & \tilde{A}_{\Omega_l \Omega_l'} \\ \tilde{A}_{\Omega_l' \Omega_l} & \tilde{A}_{\Omega_l' \Omega_l'} \end{bmatrix}.$$

Then, we update the new iterate by $\alpha_{\Omega_l}^{(k+1)} = \alpha_{\Omega_l}^{(k+1/2)}$ for each l. If pixel i belongs to more than one region, then the value of $\alpha_i^{(k+1)}$ is defined to be the average of the contributions from the several regions containing i. This method is also known as the block Jacobi method [7]. Similarly, the block Gauss Seidel method can be considered.

Next, we state that the iterations are well-defined because the matrix $\tilde{A}_{\Omega_l \Omega_l}$ is non-singular.

THEOREM 4.1. *The matrix $\tilde{A}_{\Omega_l \Omega_l}$ is non-singular for each l.*

Proof. Let B be the matrix

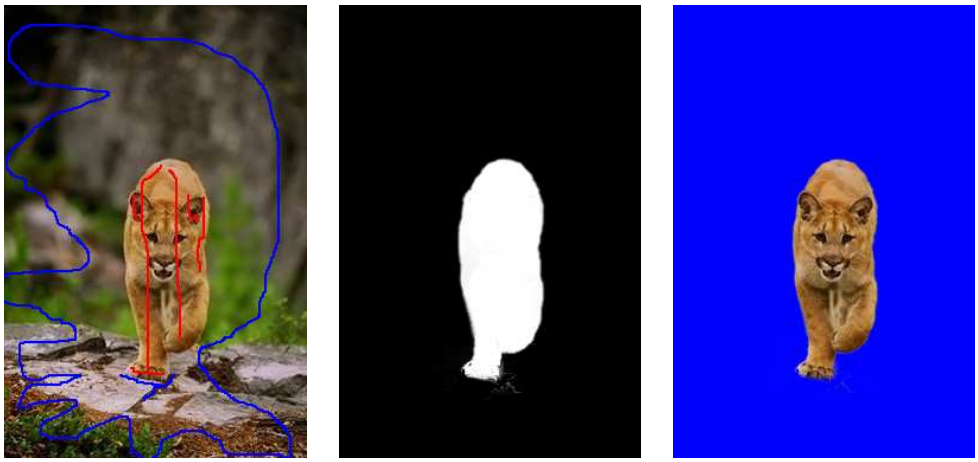
$$B = \begin{bmatrix} \tilde{A}_{\Omega_l \Omega_l} & 0 \\ 0 & I \end{bmatrix}.$$

Then we have $\tilde{A} \leq B$ and $b_{ij} \leq 0$ for all $i \neq j$. Recall that $\tilde{A}$ is an M-matrix. Then, by Theorem 1.18 in [7], B is also an M-matrix. In particular, B, and hence $\tilde{A}_{\Omega_l \Omega_l}$, is non-singular. □

In the implementation, these smaller systems can be solved by direct methods. Moreover, since the matrices $\tilde{A}_{\Omega_l \Omega_l}$ are fixed over the iterations, we can compute and store their LU factorization once and for all. In general, the regions Ω_l's should be chosen to be as large as possible to encourage a fast convergence. But they should also be small enough so that direct methods can be used efficiently. On the other hand, the convergence rate generally increases with the amount of overlapping between regions. A good feature of this method is that it can be applied to very huge problems. We just have to decompose the domain so that Ω_l's are small enough. The downside of having too many small regions is a slower convergence. Nevertheless the memory problem is solved. Moreover, this method can be easily parallelized.

5. Experimental Results.

5.1. Efficiency of Numerical Solvers. In this subsection, we test the performance of the numerical solvers discussed in §4. Three test images are shown in Figures 5.1, 5.2 and 5.3 which are used to solve a two-class, a two-class and a three-class segmentation problems respectively. In the figures, we show an original image, juxtaposed with the boundary conditions (scribbles) put on by the user. We also show the resulting membership functions and synthesized images using the membership functions. We observe that the tiger and the background are nicely separated in Figure 5.1, the Merlion and the rest are nicely separated in Figure 5.2 whereas the trees, the sky and the sea are also gracefully separated in Figure 5.3. Moreover, the membership functions allow a smoother transition from one segment to another than hard segmentation. Therefore, the synthesized images look natural.

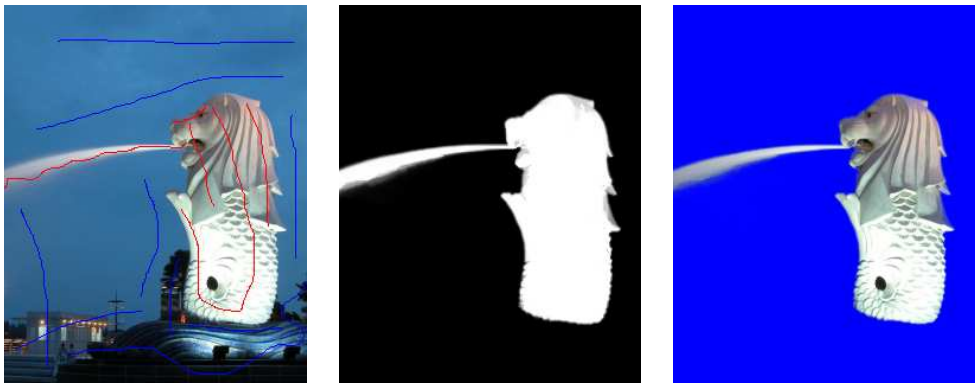


(a) (b) (c)

FIG. 5.1. *Two-class image segmentation. (a) The original* 384×256 *Tiger image juxtaposed with initial scribbles put on by the user indicating foreground (red) and background (blue). (b) The estimated membership function. (c) The foreground viewed against a uniform blue background.*

In Figures 5.4, we compare the convergence of the relative residuals of point Jacobi method, point Gauss-Seidel method and several domain decomposition methods using the Tiger image in Figure 5.1(a), the Merlion image in Figure 5.2(a) and the Sunset image in Figure 5.3(a). Domain decomposition methods include block Jacobi iterations with or without overlap and block Gauss-Seidel iterations with or without overlap. We find that the point Jacobi method, point Gauss-Seidel method and domain decomposition methods without overlap cannot reach a high level of accuracy within 100 iterations. Indeed, it essentially stagnates after 100 iterations. On the other hand, the convergence of domain decomposition methods with overlap is significantly faster.

Next, we study the effect of the amount of overlapping in the domain decomposition method. In Figures 5.5(a)–(c), we show the results using the Tiger, Merlion and Sunset images respectively. Here, the domain is decomposed into 3×2, 4×2, 3×4 blocks respectively where the size of each block is varied so that the number of overlapping pixels with its adjacent block is 0, 8, 16 and 32 pixels. The iterations are stopped if the relative residual drops below 10^{-6}. We observe that the larger the



(a) (b) (c)

FIG. 5.2. *Two-class image segmentation. (a) The original* 320×256 *Merlion image juxtaposed with initial scribbles put on by the user indicating foreground (red) and background (blue). (b) The estimated membership function. (c) The foreground viewed against a uniform blue background.*

number of overlapping pixels, the faster the convergence rate. Therefore, in practice, one may want to use as many overlapping pixels as possible while keeping the size of the linear systems to solve in each DD iteration manageable.

We can employ domain decomposition methods as preconditioners for the preconditioned GMRES method. In Figures 5.6, we compare the convergence of the relative residuals of different preconditioners: point Jacobi method, point Gauss-Seidel method and domain decomposition preconditioners of block Jacobi iterations with or without overlap and block Gauss-Seidel iterations with or without overlap. We find that the convergence behavior of the preconditioned GMRES method with all the tested preconditioners are improved with respect to the convergence results in Figure 5.4. In particular, the convergence of the preconditioned GMRES method with domain decomposition preconditioners of block Jacobi and block Gauss-Seidel iterations with overlap is very quick.

5.2. Effect of Boundary Conditions. In the image segmentation model, the user needs to specify some boundary conditions for each class of pixels. Therefore, it is useful to study how the boundary conditions should be specified. In this subsection, we carry a numerical study of the effect of boundary condition specification. We use a synthetic image and a real image, shown in Figures 5.7(a) and 5.8(a) respectively.

5.2.1. Location of boundary conditions. We first segment each image into foreground and background. In Figure 5.7(a), the foreground is the white square in the middle; the background is the black rim around the foreground. In Figure 5.8(a), the segmentation into foreground and background is done manually and is shown in Figure 5.8(b). Next, we use the pixel at the center of the image as the boundary condition for the foreground. One pixel in the background segment is used as the boundary condition for the background. This experiment is repeated using all possible locations of the background boundary condition.

To assess the effect of the location of the boundary condition, we compute the condition number of the corresponding coefficient matrix A. A smaller condition number is more desirable because the linear system can in general be solved faster. Intuitively, when boundary conditions are well specified, it should be easy to tell

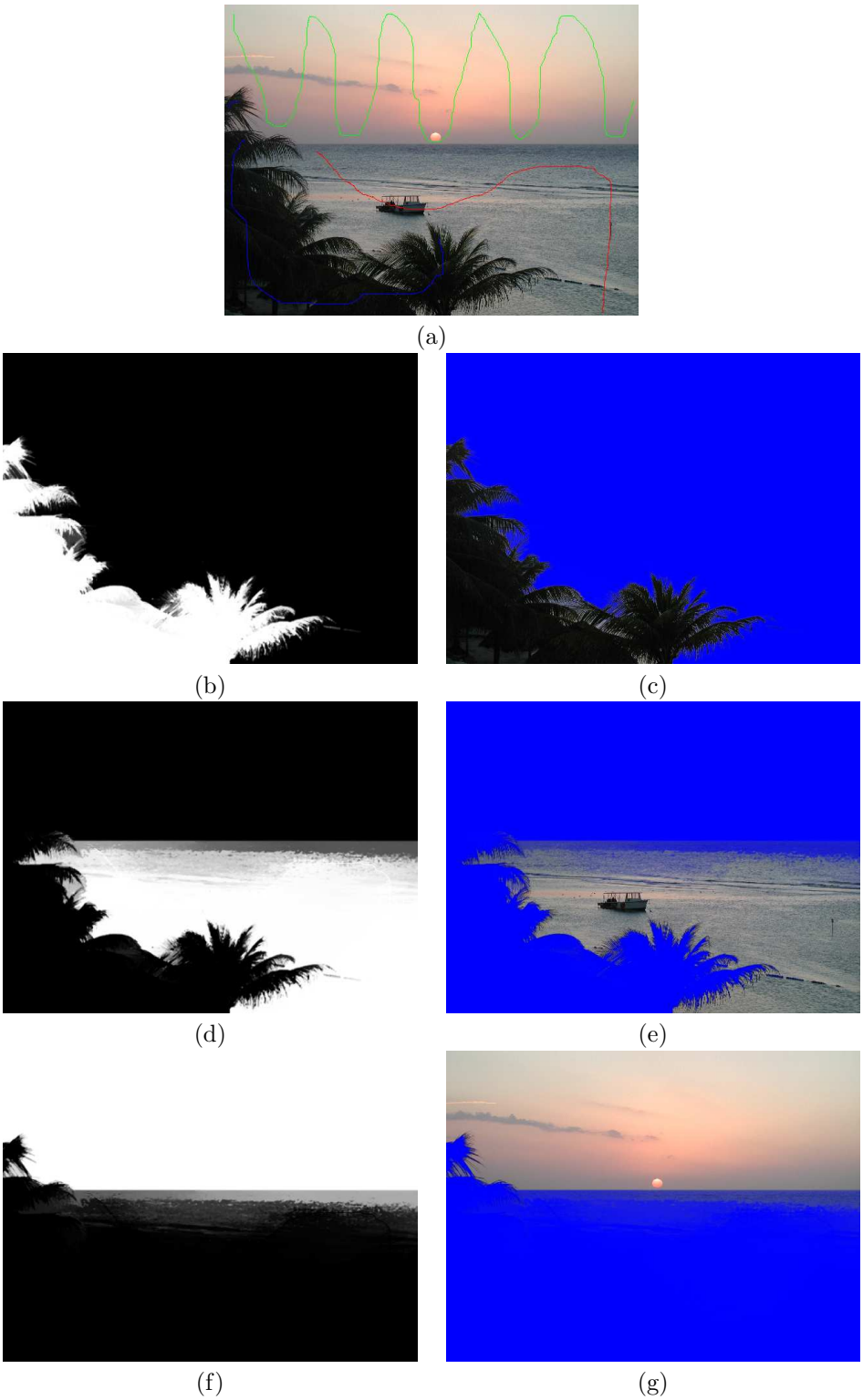
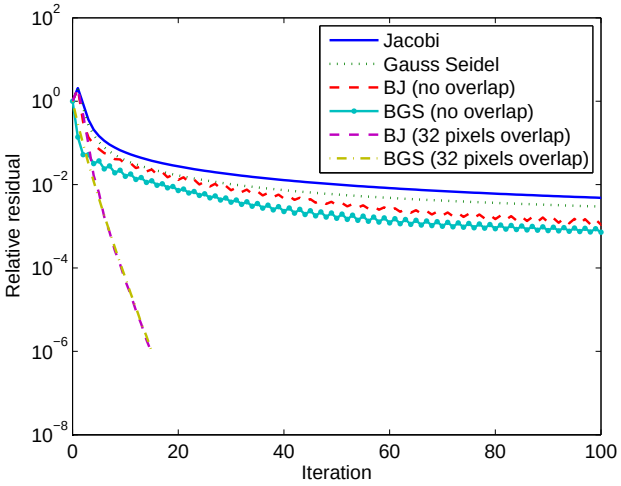
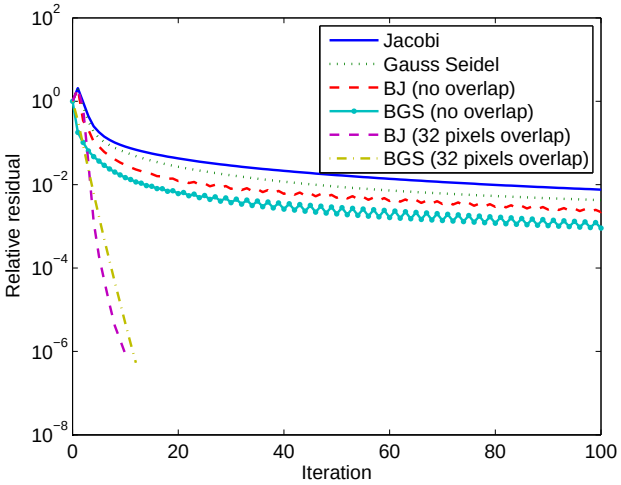


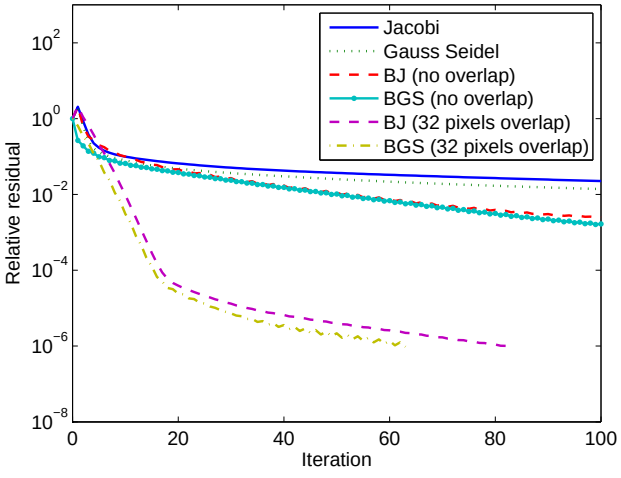
FIG. 5.3. *Three-class image segmentation. (a) The original* 480×640 *Sunset image juxtaposed with initial scribbles put on by the user indicating three regions (blue, red and green). (b),(d),(f) The estimated membership function for the blue, red and green region respectively. (c),(e),(g) The blue, red and green region viewed against a uniform blue background respectively.*



(a)

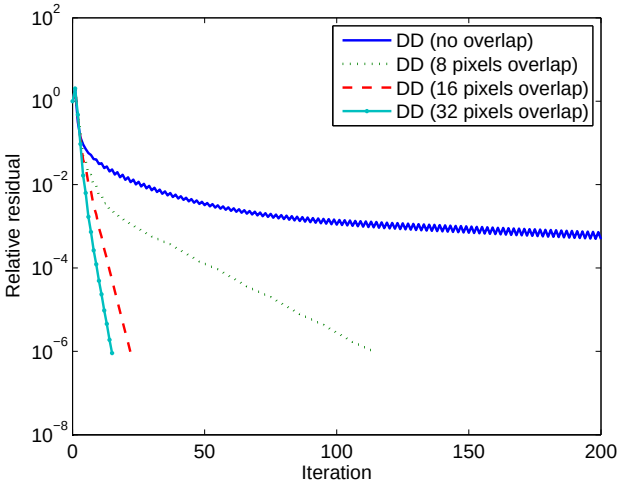


(b)

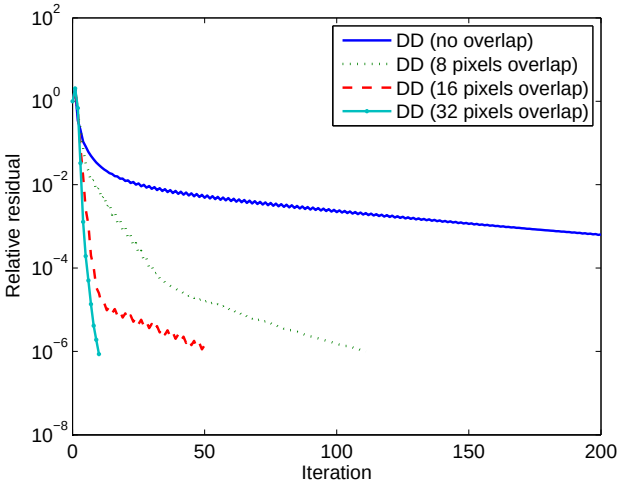


(c)

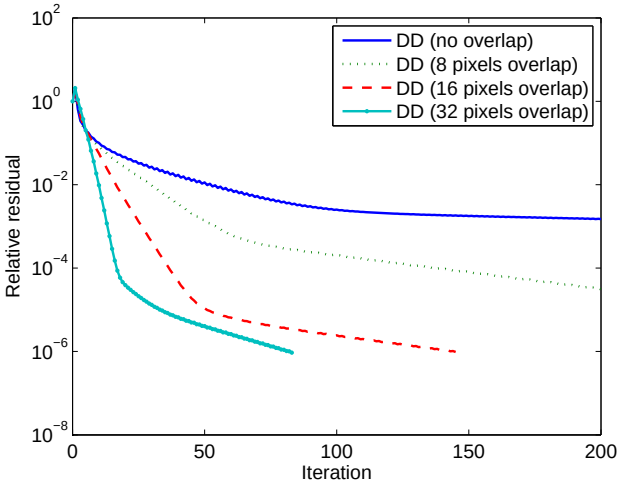
FIG. 5.4. *Convergence profile of various iterative methods. (a) The Tiger image. (b) The Merlion image. (c) The Sunset image.*



(a)

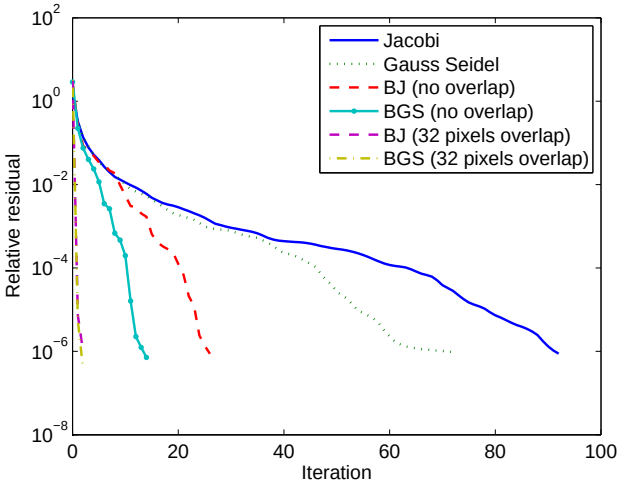


(b)

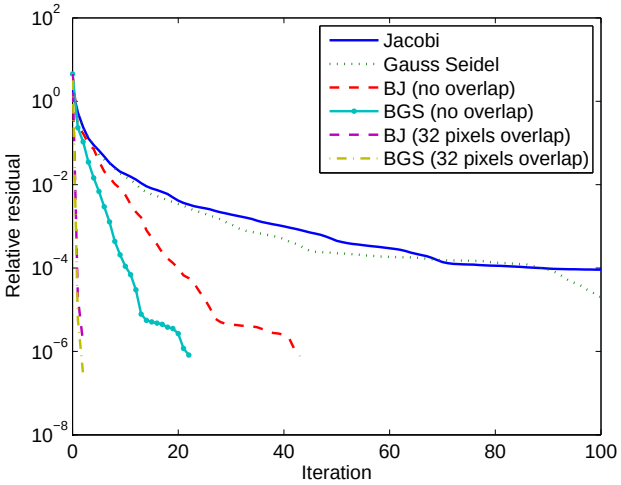


(c)

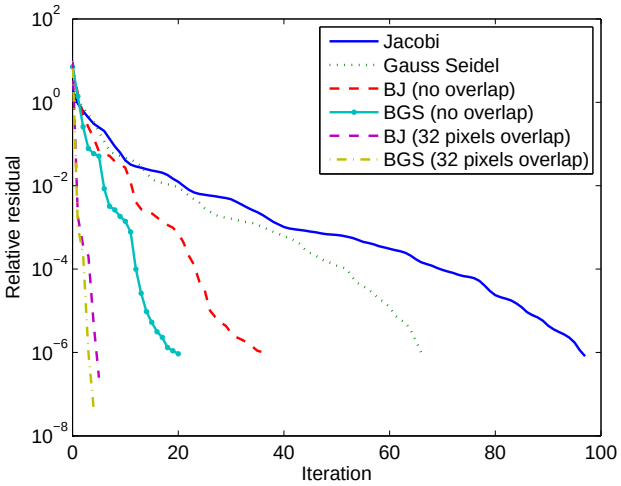
FIG. 5.5. *Convergence profile of domain decomposition method with various degree of overlapping. (a) The Tiger image. (b) The Merlion image. (c) The Sunset image.*



(a)



(b)



(c)

FIG. 5.6. *Convergence profile of the preconditioned GMRES method with domain decomposition preconditioners. (a) The Tiger image. (b) The Merlion image. (c) The Sunset image.*

whether each pixel belongs the foreground or the background. This is in turn reflected in the condition number of A.

In Figures 5.7(d) and 5.8(e), the plots of the logarithmic condition number versus the background boundary condition location are shown. We see that the condition number increases significantly as the location is getting close to the foreground. This is expected because the feature vectors are computed over a small window. Therefore, when the background boundary condition is close the the foreground, the feature vector is computed using pixels from both the background and foreground. As a result, such a feature vector can be very different from those deep inside the background and those deep inside the foreground. This makes the classification of the true background pixels difficult because their feature vectors do not match any of the feature vectors from the boundary conditions. Another interesting observation from Figure 5.7(d) is that the condition number is smaller at the middle of the rim and is getting larger as we move towards either the inner or the outer boundary of the background segment.

In Figures 5.7(b) and 5.8(c), the membership functions resulted from the location leading to the best conditioned A are shown. The result in Figure 5.7(b) is very reasonable. But the result in Figure 5.8(c) is not ideal. It is because the picture in Figure 5.8(a) is quite complex, and therefore, boundary conditions with more number of pixels are needed to obtain good results.

In Figures 5.7(c) and 5.8(d), the membership functions resulted from the location leading to the worst conditioned A are shown. We can observe that these membership functions are of much poorer quality.

5.2.2. Size of boundary conditions. In this test, we set out to investigate the effect of the number of pixels used in the boundary conditions. We randomly pick $k = 5, 10, 15, 20, \ldots, 800$ pixels from the image to be used as boundary conditions and record the condition number of the corresponding coefficient matrix A. Note that the matrix A is independent of whether each chosen pixel is used as foreground or background boundary condition. For each fixed number of pixels k, we repeat the test 10 times with different sets of k pixels chosen. Then, the average condition number and the average $\pm$ one standard deviation are reported. The resulting plots for the Square image in Figure 5.7(a) and the Black Swallowtail image in Figure 5.8(a) are shown in Figures 5.9(a) and (b) respectively. As expected, the average condition number decreases as more pixels as specified as boundary conditions. Moreover, the decrease is more significant initially.

6. Concluding Remarks. In this paper, we have studied theoretically and numerically the problem of multiple class image segmentation. We show that the existence and uniqueness of the optimal membership function. We also show through the discrete maximum principle that the bilateral constraints $0 \leq \alpha \leq 1$ are automatically satisfied so that the simpler unconstrained counterpart can be used instead. We propose to use domain decomposition methods and the preconditioned GMRES method with domain decomposition preconditioners to solve the optimality condition. Our experimental results show that domain decomposition preconditioners with overlap are very effective for such linear systems. Furthermore, we have studied numerically the effect of boundary conditions and shown that the condition number of A can in general be decreased by specifying the boundary conditions at the central location of the respective region and by using more pixels as boundary conditions.

REFERENCES

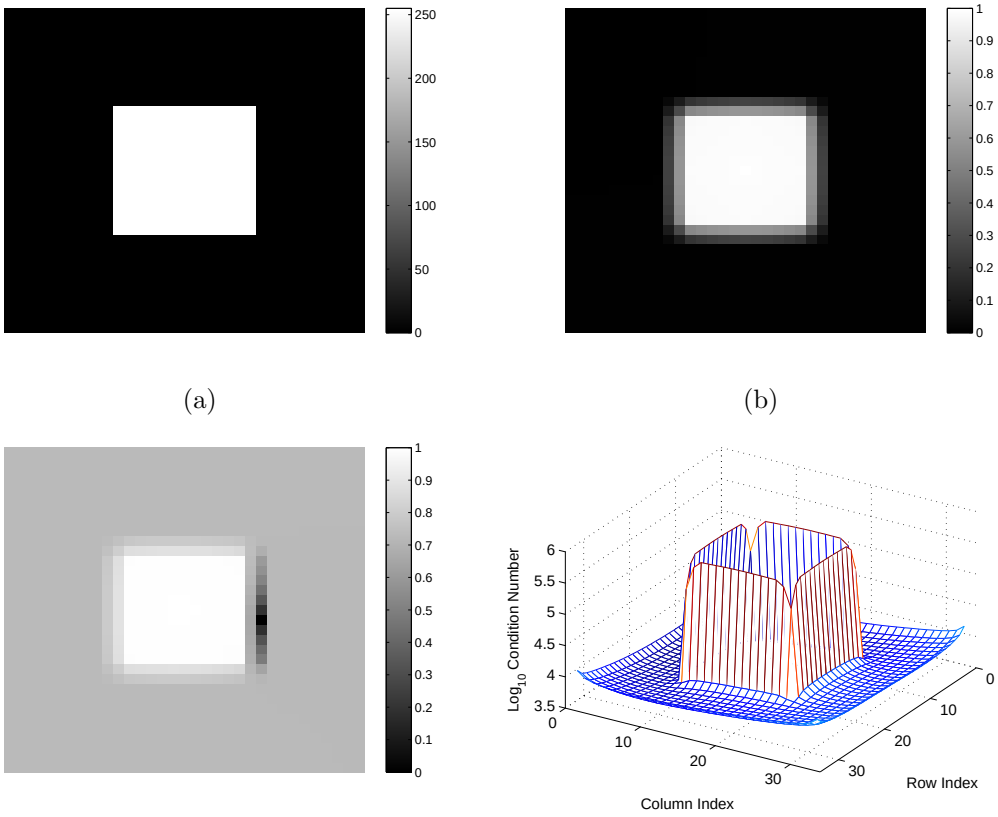


FIG. 5.7. *Effect of location of boundary conditions (synthetic image). (a) The input* 33×33 *image with a black background and a white foreground. The foreground boundary condition is given at the pixel at the center of the image. (b) The membership function obtained in the best conditioned case, where the background boundary condition is given at* $(10, 6)$*. (c) The membership function obtained in the worst conditioned case, where the background boundary condition is given at* $(18, 24)$*. (d) Logarithmic condition number of A with respect to location of the background boundary condition.*

[1] Y. Boykov and V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, IEEE Trans. PAMI, V26 (2004), pp. 1124-1137.
[2] C. Rother, V. Kolmogorov and A. Blake, "GrabCut": Interactive foreground extraction using iterated graph cuts, ACM Trans. Graph., V23(3) (2004), pp. 309-314.
[3] J. Wang and M. Cohen, An iterative optimization approach for unified image segmentation and matting, Proc. IEEE Int. Conf. on Computer Vision, 2005.
[4] S. Yu and J. Shi, Segmentation given partial grouping constraints, IEEE Trans. PAMI, V26 (2004), pp. 173-183.
[5] J. Guan and G. Qiu, Interactive image matting using optimization, technical report 01-2006, School of Computer Science and Information Technology, University of Nottingham, U.K., 2006.
[6] J. Guan and G. Qiu, Interactive image segmentation using optimization with statistical priors, Proc. ECCV 2006, Graz, Austria, 2006.
[7] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing Company, 1996.
[8] J. Shi and J. Malik, Normalized cuts and image segmentation, IEEE Trans. PAMI, V22 (2000), pp. 888-905.

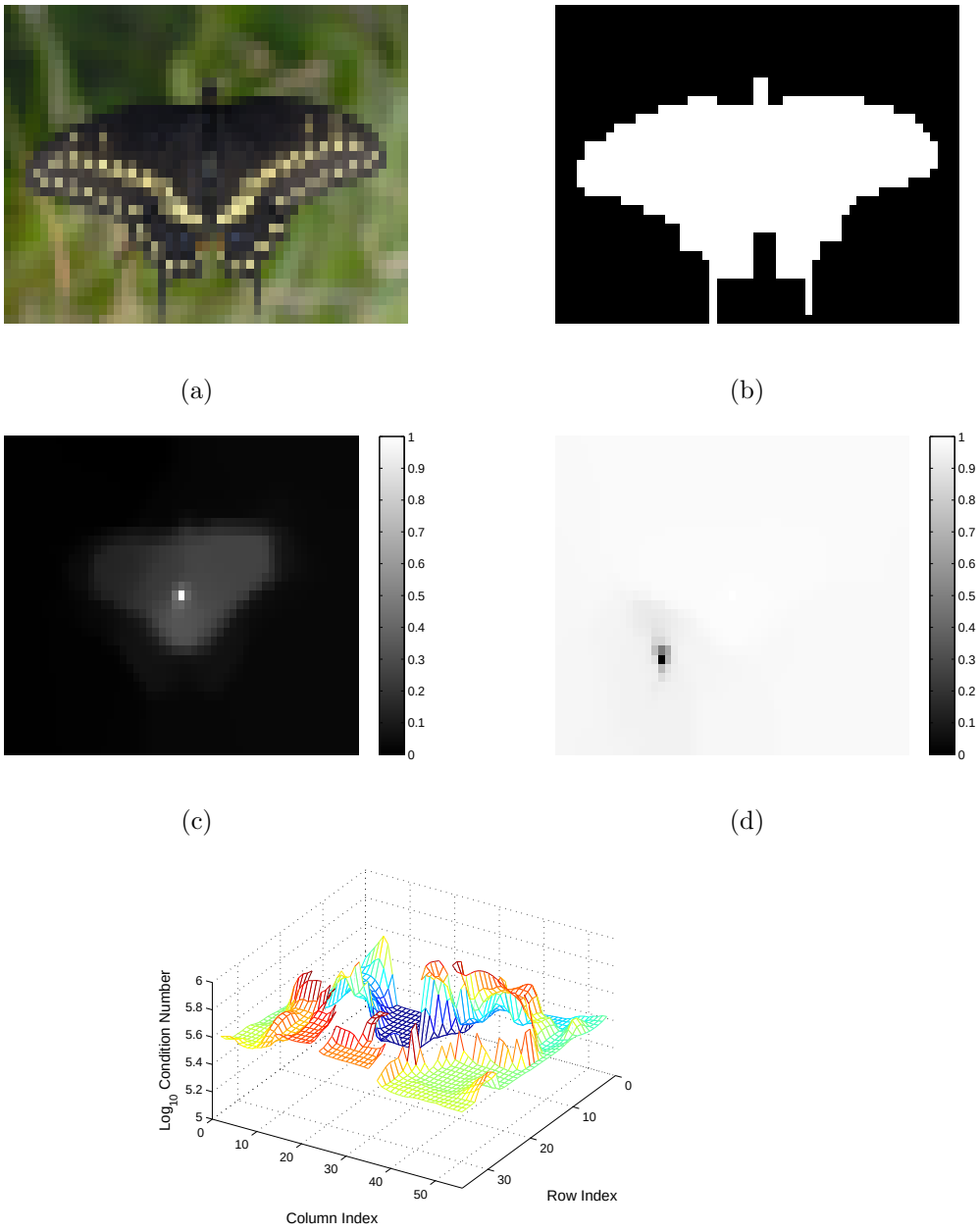
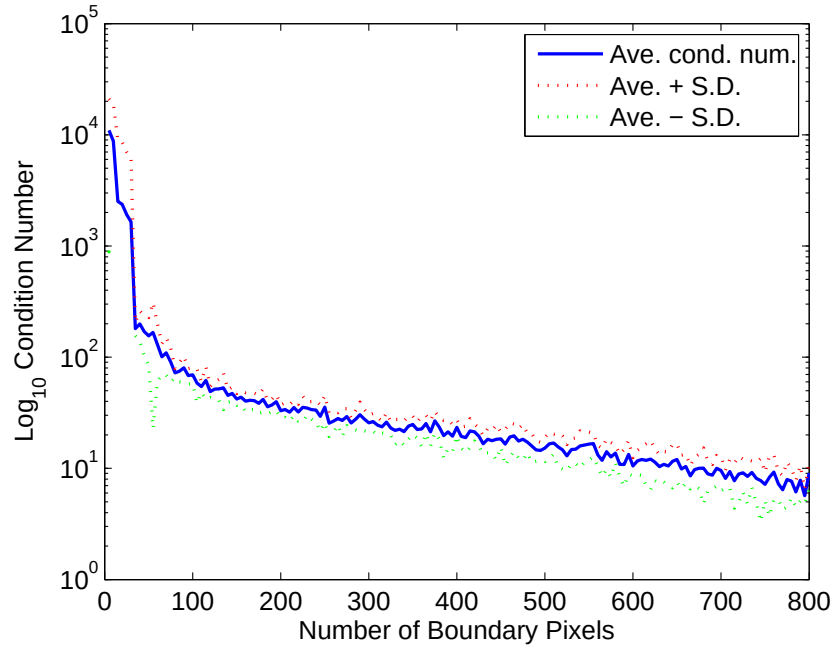
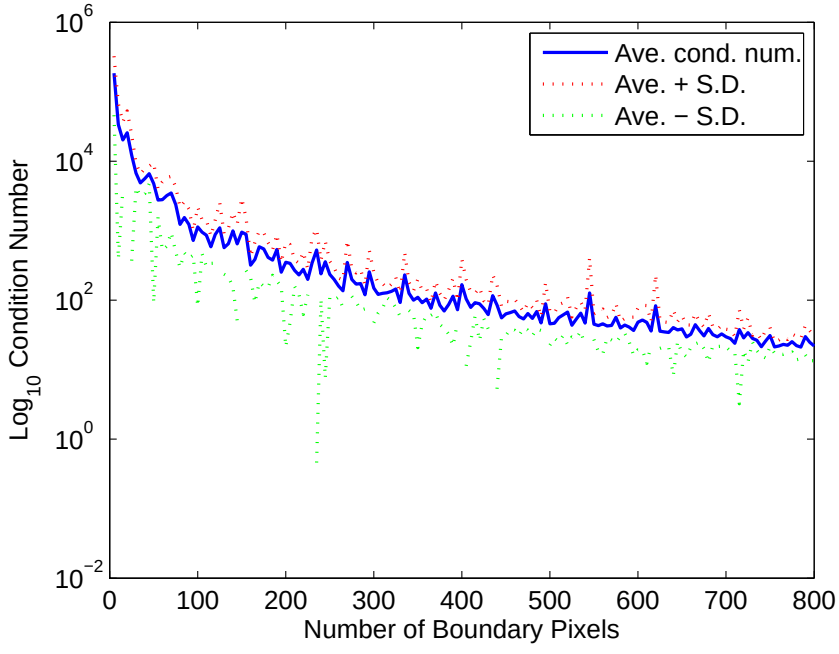


FIG. 5.8. *Effect of location of boundary conditions (real image). (a) The input* 35×55 *Black Swallowtail image. (b) Foreground and background segmentation obtained manually. The foreground boundary condition is given at the pixel at the center of the image. (c) The membership function obtained in the best conditioned case, where the background boundary condition is given at pixel location* $(5, 21)$*. (d) The membership function obtained in the worst conditioned case, where the background boundary condition is given at pixel location* $(25, 17)$*. (e) Logarithmic condition number of* A *with respect to location of the background boundary condition.*



(a)



(b)

FIG. 5.9. *Effect of number of pixels used in the boundary conditions. The blue curve shows the average logarithmic condition number of A versus the number of randomly chosen boundary pixels. The red (green) curve shows the average plus (minus) one standard deviation of the logarithmic condition number of A. (a) Results using the Square image in Figure 5.7(a). (b) Results using the Black Swallowtail image in Figure 5.8(a).*