Digital Signal Processing 37 (2015) 65–74

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Fast local image inpainting based on the Allen–Cahn model

Yibao Li[a], Darae Jeong[b], Jung-il Choi[c], Seunggyu Lee[b], Junseok Kim[b,*]

[a] School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China
[b] Department of Mathematics, Korea University, Seoul 136-713, Republic of Korea
[c] Department of Computational Science and Engineering, Yonsei University, Seoul, 120-749, Republic of Korea

ARTICLE INFO

Article history:
Available online 5 December 2014

Keywords:
Image inpainting
Energy minimization
Allen–Cahn equation
Operator splitting
Unconditionally stable scheme

ABSTRACT

In this paper, we propose a fast local image inpainting algorithm based on the Allen–Cahn model. The proposed algorithm is applied only on the inpainting domain and has two features. The first feature is that the pixel values in the inpainting domain are obtained by curvature-driven diffusions and utilizing the image information from the outside of the inpainting region. The second feature is that the pixel values outside of the inpainting region are the same as those in the original input image since we do not compute the outside of the inpainting region. Thus the proposed method is computationally efficient. We split the governing equation into one linear equation and one nonlinear equation by using an operator splitting technique. The linear equation is discretized by using a fully implicit scheme and the nonlinear equation is solved analytically. We prove the unconditional stability of the proposed scheme. To demonstrate the robustness and accuracy of the proposed method, various numerical results on real and synthetic images are presented.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

Image inpainting is the process of filling in missing or damaged parts of images based on information from surrounding areas [1]. Up to now, a large number of algorithms have been proposed to solve the image inpainting problem. For the recent survey of the theoretical foundations, the different categories of methods, and illustrations of the main applications about the image inpainting, see the review paper [2] and references therein. Qin et al. [3] proposed an efficient image inpainting approach, which progressively propagates neighboring information into damaged region and can restore sharp edge successfully. In [4], the authors proposed a compact and fast PDE-based inpainting method using anisotropic heat transfer model, which can propagate both the structure and texture information from surrounding region into damaged region simultaneously. Liu and Caselles [5] presented a novel formulation of exemplar-based inpainting [6–8] as a global energy optimization problem, written in terms of the offset map. They also proposed a multiscale graph cuts algorithm to efficiently solve the energy minimization problem. Recently, Ramamurthy et al. [9] regularized the sparse models with manifold projection for image inpainting and Turek et al. [10] used the signal generation model for image inpainting.

Many inpainting algorithms are based on partial differential equations, in which the missing region is filled by diffusing the image information from the known region into the missing region at the pixel level [11–13]. Among them, one of widely applied methods is proposed by Chan and Shen [12]. They proposed a variational framework based on total variation to recover the missing information, which minimizes the following energy functional:

$$\mathcal{E}_{TV}(c) = \int_{\Omega} |\nabla c| d\mathbf{x} + \int_{\Omega} \frac{\lambda(\mathbf{x})}{2} \big(f(\mathbf{x}) - c\big)^2 d\mathbf{x}, \tag{1}$$

where $\mathbf{x} = (x, y)$, $f(\mathbf{x})$ is a given image, and c is the gray scale image in a domain $\Omega \subset \mathbb{R}^2$. $\Omega_D \subset \Omega$ is the inpainting domain, $\partial\Omega_D$ is the boundary of inpainting domain, and $\Omega \backslash \Omega_D$ is the complement of Ω_D in Ω (see Fig. 1). The fidelity term $\lambda(\mathbf{x})(f(\mathbf{x}) - c)^2$ was used to keep the solutions close to the given image in $\Omega \backslash \Omega_D$. For this purpose, $\lambda(\mathbf{x}) = 0$ if $\mathbf{x} \in \Omega_D$; otherwise $\lambda(\mathbf{x}) = \lambda_0$.

The steepest descent equation for the energy functional (1) is given by

$$\frac{\partial c}{\partial t} = \nabla \cdot \left(\frac{\nabla c}{|\nabla c|}\right) + \lambda(\mathbf{x})\big(f(\mathbf{x}) - c\big). \tag{2}$$

[*] Corresponding author. Fax: +82 2 929 8562.
E-mail addresses: yibaoli@xjtu.edu.cn (Y. Li), cfdkim@korea.ac.kr (J. Kim).
URLs: http://gr.xjtu.edu.cn/web/yibaoli (Y. Li), http://math.korea.ac.kr/~cfdkim (J. Kim).

http://dx.doi.org/10.1016/j.dsp.2014.11.006
1051-2004/© 2014 Elsevier Inc. All rights reserved.

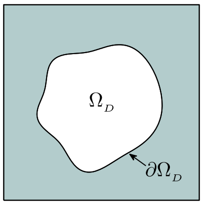


Fig. 1. Schematic illustration of the inpainting domain.

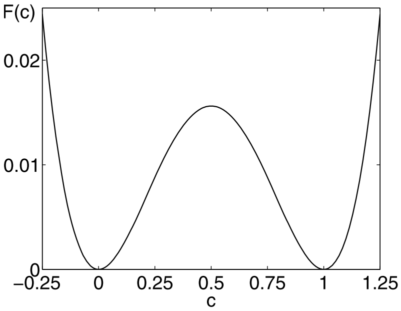


Fig. 2. A double well-potential, $F(c) = 0.25c^2(1-c)^2$.

Hence,

$$\frac{\partial c}{\partial t} = \begin{cases} \nabla \cdot (\frac{\nabla c}{|\nabla c|}) & \text{if } \mathbf{x} \in \Omega_D, \\ \nabla \cdot (\frac{\nabla c}{|\nabla c|}) + \lambda_0 (f(\mathbf{x}) - c) & \text{if } \mathbf{x} \in \Omega \backslash \Omega_D. \end{cases} \tag{3}$$

As can be observed, the pixel values in the solution are the same as the values in $\Omega \backslash \Omega_D$, i.e., $c \approx f(\mathbf{x})$. And in the inpainting domain, the pixel values are obtained by curvature-driven diffusions. In [14], Chan and Shen proposed a new inpainting model based on the curvature-driven diffusions to realize the connectivity principle.

$$\frac{\partial c}{\partial t} = \nabla \cdot \left(G\big(\mathbf{x}, |\kappa|\big) \frac{\nabla c}{|\nabla c|} \right) + \lambda \big(f(\mathbf{x}) - c\big). \tag{4}$$

Here,

$$G\big(\mathbf{x}, |\kappa|\big) = \begin{cases} |\kappa|^p & \text{if } \mathbf{x} \in \Omega_D, \\ 1 & \text{if } \mathbf{x} \in \Omega \backslash \Omega_D, \end{cases} \tag{5}$$

where $p \geq 1$ and κ is the curvature, which is given by $\nabla \cdot (\nabla c / |\nabla c|)$. In [13], Esedoḡlu and Shen introduced the Mumford–Shah–Euler model, which is based on the Mumford–Shah image segmentation model [15], to solve the image inpainting problems. Ballester et al. [16] adapted a joint interpolation of vector fields and gray-levels to incorporate the principle of continuity in a variational framework. Li et al. [17] proposed a fast scheme to solve the Chan and Shen's inpainting model [12] with an operator splitting method. Another well-known method was introduced by Bertozzi et al. [18,19], where they proposed an inpainting model which is the modified Cahn–Hilliard (CH) equation

$$\begin{aligned} \mathcal{E}_B(c) &= \mathcal{E}_{B1}(c) + \mathcal{E}_{B2}(c) \\ &= \int_\Omega \left(\frac{F(c)}{\epsilon^2} + \frac{|\nabla c|^2}{2} \right) d\mathbf{x} + \int_\Omega \frac{\lambda}{2} \big(f(\mathbf{x}) - c\big)^2 d\mathbf{x}, \end{aligned} \tag{6}$$

where $F(c) = 0.25c^2(1-c)^2$ is the Helmholtz free energy density (see Fig. 2). The term $F(c)$ is a force that makes c to be approximately 0 or 1. $|\nabla c|^2$ is a gradient energy, ϵ is the gradient energy coefficient related to the interfacial energy.

By a superposition of gradient descent with respect to H^{-1} inner product for the energy $\mathcal{E}_{B1}$ and gradient descent with respect to L^2 inner product for the energy $\mathcal{E}_{B2}$, the authors in [18,19] proposed the following model:

$$\frac{\partial c}{\partial t} = \Delta \left(\frac{F'(c)}{\epsilon^2} - \Delta c \right) + \lambda \big(f(\mathbf{x}) - c\big). \tag{7}$$

We note that if $\lambda = 0$ in Eq. (7), then the equation becomes the classical CH equation [20], which has been used as a mathematical model to investigate the phase separation of binary mixture under quenching below a critical temperature. For physical, mathematical, and numerical derivations of the CH equation, see the recent review paper [21].

In this paper, we propose a new effective and accurate image inpainting method which is based on the local Allen–Cahn equation [22]. The equation has been used in solving problems in image processing [23–27]. By using the Allen–Cahn equation, we can perform fast image inpaintings, because its fast and accurate hybrid numerical solver is available [28]. It should be pointed that our model is the extension of Chan and Shen's model [12], since Allen–Cahn equation describes the motion of mean curvature flow. Compared to Cahn–Hilliard equation [18,19], we choose to use the Allen–Cahn equation, since its numerical treatment is simpler than that of the Cahn–Hilliard type, which involves fourth-order differential operators. The outline of this paper is the following. In Section 2, the governing equations for the image inpainting are presented. In Section 3, we describe the proposed operator splitting algorithm and give a detailed proof for its unconditional stability. In Section 4, we present computational examples to demonstrate the efficiency and robustness of our proposed method. Finally, conclusions are drawn in Section 5.

2. Description of the proposed model

The inpainting algorithms can be summarized as: First, the missing region is filled by diffusing the image information from the known region into the missing region at the pixel level. Second, the image information in the known region should be close to the given image. To reduce the computational cost and keep the accuracy of the inpainting algorithm, we propose the following equations:

$$\frac{\partial c}{\partial t}(\mathbf{x}, t) = \begin{cases} -F'(c(\mathbf{x}, t))/\epsilon^2 + \Delta c(\mathbf{x}, t) & \text{if } \mathbf{x} \in \Omega_D, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

$$c(\mathbf{x}, 0) = \begin{cases} 0.5 & \text{if } \mathbf{x} \in \Omega_D, \\ \frac{f(\mathbf{x}) - f_{\min}}{f_{\max} - f_{\min}} & \text{otherwise.} \end{cases} \tag{9}$$

$$\mathbf{n} \cdot \nabla c(\mathbf{x}, t) = 0, \quad \mathbf{x} \in \partial\Omega, \tag{10}$$

where $f_{\max}$ and $f_{\min}$ are the maximum and the minimum values of the given image, respectively, and $\mathbf{n}$ is the unit normal vector to $\partial\Omega$. Thus, we have the normalized image data $c(\mathbf{x}, 0) \in [0, 1]$. Eq. (8) is also called as the Allen–Cahn equation [22], which is widely applied in image processing due to the motion by mean curvature. We choose the AC equation because it has intrinsic smoothing effect on interfacial transition layers and its fast and accurate hybrid numerical solver is available [28].

For the initial condition, we can choose different initial guesses for the inpainting domain Ω_D because the solution of missing regions will be defined by the information of the known region at the equilibrium solution. In this work, to reduce the numerical iterations, we set $c(\mathbf{x}, 0) = 0.5$ in the inpainting domain.

3. Numerical solution algorithm

3.1. Proposed operator splitting algorithm

In this section, we propose an operator splitting method to get an unconditionally stable numerical method for solving the proposed algorithm. Let the computational domain Ω be $[1, N_x] \times$

$[1, N_y]$, h be the uniform grid size, and $\Omega_h = \{(x_i, y_j) | x_i = (i - 0.5)h,\ y_j = (j - 0.5)h,\ 1 \le i \le N_x,\ 1 \le j \le N_y\}$. Let c_{ij}^n be approximations of $c(x_i, y_j, n\Delta t)$, where $\Delta t = T/N_t$ is the time step, T is the final time, and N_t is the total number of time steps. Here, the superscript n denotes an evaluation at time level n. We split the original problem (8) into a sequence of simpler problems as

$$c_t = \Delta c, \tag{11}$$

$$c_t = -\frac{F'(c)}{\epsilon^2}. \tag{12}$$

As the first step, we solve Eq. (11) by applying the fully implicit method with an initial condition c^n, that is,

$$\frac{c_{ij}^{n+\frac{1}{2}} - c_{ij}^n}{\Delta t} = \Delta_d c_{ij}^{n+\frac{1}{2}}, \tag{13}$$

where Δ_d is the 9-point discrete Laplacian operator, i.e.,

$$\Delta_d c_{ij}^{n+\frac{1}{2}} = \frac{c_{i+1,j+1}^{n+\frac{1}{2}} + c_{i+1,j-1}^{n+\frac{1}{2}} + c_{i-1,j+1}^{n+\frac{1}{2}} + c_{i-1,j-1}^{n+\frac{1}{2}}}{6h^2} + \frac{2(c_{i,j-1}^{n+\frac{1}{2}} + c_{i,j+1}^{n+\frac{1}{2}} + c_{i+1,j}^{n+\frac{1}{2}} + c_{i-1,j}^{n+\frac{1}{2}}) - 10c_{ij}^{n+\frac{1}{2}}}{3h^2}.$$

The resulting implicit discrete Eq. (13) can be solved by the Gauss–Seidel method. Next, Eq. (12) is solved by the method of separation of variables [28–31] as

$$c^{n+1} = \frac{1}{2} + \frac{c^{n+\frac{1}{2}} - 0.5}{\sqrt{e^{\frac{-\Delta t}{2\epsilon^2}} + (2c^{n+\frac{1}{2}} - 1)^2 (1 - e^{\frac{-\Delta t}{2\epsilon^2}})}}. \tag{14}$$

For more details, we refer to [28] and note that in the previous numerical solution of Eq. (11), the authors used the Crank–Nicolson scheme, which is prone to oscillations when a large time step is used. However, in this paper, we do not have oscillations because we use the fully implicit scheme.

To solve Eqs. (11) and (12) in Ω_D, we introduce a control function g, which is defined as

$$g_{ij} = \begin{cases} 1 & \text{if } (x_i, y_j) \in \Omega_D, \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

The schematic diagram is shown in Fig. 3. c^{n+1} is updated from c^n by solving Eqs. (13) and (14) only when $g_{ij} = 1$. We note that the closest neighboring exterior cells to the boundary of Ω_D play a role as the Dirichlet boundary condition. Therefore, c_{ij}^n can have values between zero and one depending on the neighboring image values.

In summary, our discrete solution algorithm is described as following:

Step 1. Find the inpainting domain Ω_D using various methods: for example, either by inspection or by image segmentation methods [29,32].

Step 2. Define the control function g and initialize c^0 as

$$g_{ij} = \begin{cases} 1 & \text{if } (x_i, y_j) \in \Omega_D, \\ 0 & \text{otherwise.} \end{cases} \quad \text{and}$$

$$c_{ij}^0 = \begin{cases} 0.5 & \text{if } (x_i, y_j) \in \Omega_D, \\ \frac{f_{ij} - f_{\min}}{f_{\max} - f_{\min}} & \text{otherwise.} \end{cases}$$

Step 3. Update c^{n+1} from c^n by solving Eqs. (13) and (14).

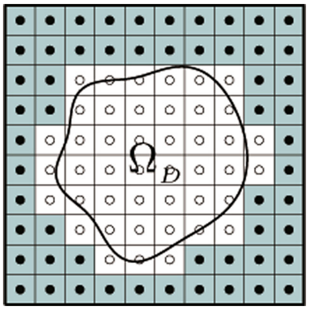


Fig. 3. Schematic illustration of the function g. The value of g is 1 (white circle, ∘) on the inpainting domain Ω_D and 0 (black circle, •) on the other domain.

An efficient preconditioned conjugate gradient (PCG) scheme is used to solve imaging problems [33].

3.2. Stability of the proposed method

Our proposed hybrid splitting method, Eqs. (13) and (14), is an unconditionally stable scheme. To confirm the unconditional stability of Eq. (13), we apply the von Neumann analysis. Let c_{ij}^n and $c_{ij}^{n+1/2}$ be decomposed into Fourier series as

$$c_{ij}^n = \sum_{\alpha=-N_x}^{N_x} \sum_{\beta=-N_y}^{N_y} \gamma_{\alpha\beta}^n e^{k(\alpha i + \beta j)\pi h} \quad \text{and}$$

$$c_{ij}^{n+1/2} = \sum_{\alpha=-N_x}^{N_x} \sum_{\beta=-N_y}^{N_y} \gamma_{\alpha\beta}^{n+1} e^{k(\alpha i + \beta j)\pi h},$$

respectively. Here, we denote n on γ as a power and k as the imaginary unit, i.e., $k = \sqrt{-1}$. Let $\tilde{c}_{ij}^n = \gamma^n e^{k(\alpha i + \beta j)\pi h}$ and $\tilde{c}_{ij}^{n+1/2} = \gamma^{n+1} e^{k(\alpha i + \beta j)\pi h}$ denote one summand of the series, respectively, where we drop the summand index in γ^n and γ^{n+1}. These formulas are now substituted into Eq. (13) and we obtain

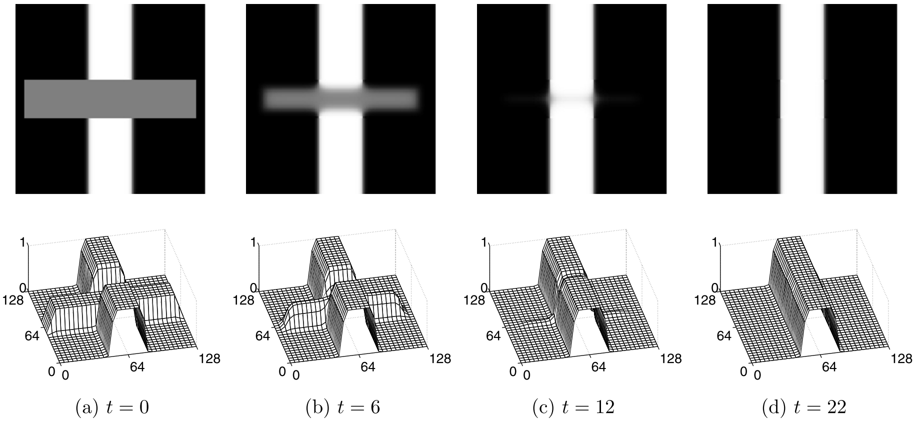
$$\gamma = \frac{3h^2}{3h^2 + 2\Delta t(\sin^2 \frac{(\alpha+\beta)\pi h}{2} + \sin^2 \frac{(\alpha-\beta)\pi h}{2} + 4\sin^2 \frac{\alpha\pi h}{2} + 4\sin^2 \frac{\beta\pi h}{2})}.$$

We can see that the above γ satisfies the property $|\gamma| \le 1$ for any α and β. Hence, the numerical scheme Eq. (13) is unconditionally stable. Furthermore, the inequality $\inf(c^n) \le c^{n+\frac{1}{2}} \le \sup(c^n)$ is satisfied by the discrete minimum and maximum principles for the heat equation [34]. Thus from Eq. (13), we get $0 \le c^{n+\frac{1}{2}} \le 1$ and this result induces $0 \le c^{n+1} \le 1$ from Eq. (14). Therefore our proposed scheme, Eqs. (13) and (14), is unconditionally stable for any time step.

4. Numerical tests

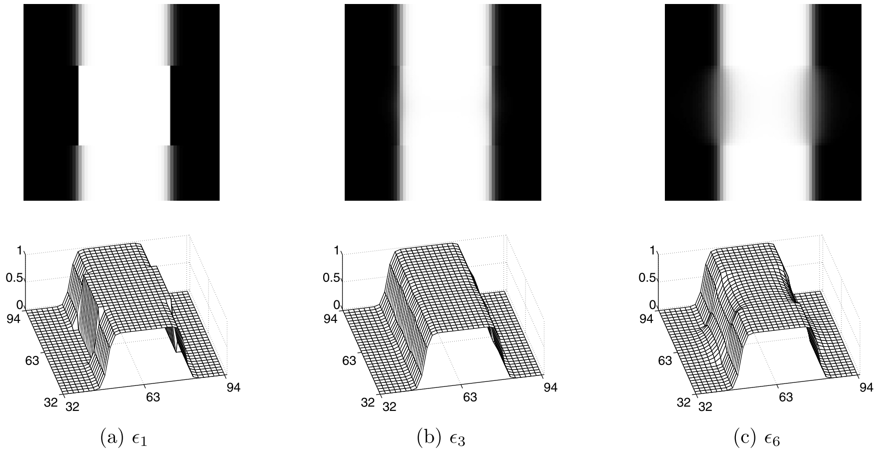
In this section, we present numerical results which are obtained by using the proposed numerical algorithm on various synthetic and real images. Unless otherwise specified, we use uniform grid size $h = 1$ and time step $\Delta t = 2$. The ϵ value is given as $\epsilon_m = hm/[4\sqrt{2}\tanh^{-1}(0.9)]$ [35], where hm is approximately the transition length of c from 0.05 to 0.95. We also will present the CPU times in seconds of our calculations which are performed in C++ on a 3 GHz with 3 G RAM.

We stop the numerical computations when the difference between the $(n+1)$th and nth time step energies becomes less than



(a) $t = 0$ (b) $t = 6$ (c) $t = 12$ (d) $t = 22$

Fig. 4. Evolution of inpainting an image. Top row: image plots. Bottom row: mesh plots. The times are shown below each figure.



(a) ϵ_1 (b) ϵ_3 (c) ϵ_6

Fig. 5. Inpainting results with different ϵ parameter values. The top row shows the final inpainting results at time $T = 1000$. The bottom row shows mesh plots of the final inpainting results.

a given tolerance, *tol*. In this paper, we set $tol = 0.1$. The termination criterion algorithm is listed as follows:

```
Set a maximum iteration number $N$, a tolerance $tol$, and $k = 1$.
While $(k \le N)$ do Steps 1–2
    Step 1 Compute $\phi^{n+1}$ from $\phi^n$ by solving Eqs. (13) and
        (14).
    Step 2 If $|\mathcal{E}_{B1}(\phi^{n+1}) - \mathcal{E}_{B1}(\phi^n)| < tol$, then stop the calcu-
        lation.
        Else $k = k + 1$.
```

4.1. Basic mechanism of the algorithm

We start with an example which illustrates the basic mechanism of the algorithm, Eqs. (8)–(10). Let us consider a synthetic image as shown in Fig. 4(a) on the computational domain $\Omega = [1, 128] \times [1, 128]$. Here, white regions are close to 1 and black regions are close to 0. Gray region is an inpainting domain and is close to 0.5. Interfacial energy coefficient ϵ_3 is used. The top and bottom rows in Fig. 4 show the evolution of the gray region and mesh plots of the numerical solutions, respectively. The evolution times are shown below each figure. The inpainting region evolves according to neighboring values, i.e., either 0 or 1.

4.2. Effect of parameter ϵ

We perform a parameter ϵ sensitivity analysis for the model. The role of ϵ is interface thickness of a transition layer of the separated region which represents two different states. We take the same initial condition except for different interface parameter values ϵ_1, ϵ_3, and ϵ_6. From the results shown in Fig. 5, we can observe that when ϵ is too small, interfacial transition is too sharp. On the other hand, if ϵ is too large, the inpainting result is blurry in appearance.

To show the relation of the parameter ϵ and the time step Δt, we perform the same test as the above with different time steps, $\Delta t = 0.1\epsilon^2$, $2\epsilon^2$, and $10\epsilon^2$. Here, $\epsilon = \epsilon_3$ is used. From Fig. 6, we can see that when the time step is larger, the interfacial transition is getting sharper. Therefore, too large time step is not good for getting smooth results.

4.3. Sensitivity to initial conditions

Fig. 7 shows the evolution of inpainting an image with respect to three different initial conditions on $\Omega = [1, 128] \times [1, 128]$. Here, ϵ_3 is employed. As shown in Fig. 7(a), we set $c(x, y, 0) = 1$ for $(x, y) \in \Omega_D$. Even though $F(c) = 0$ and $|\nabla c|^2/2 = 0$ for $(x, y) \in \Omega_D$, our method also works well because of $\mathbf{n} \cdot \nabla c \neq 0$ on $\partial\Omega_D$. The inpainting result is obtained after 340 iterations. Another initial condition is with random perturbations on Ω_D, i.e., $c(x, y, 0) = \text{rand}(x, y)$, where $\text{rand}(x, y)$ is a random number between 0 and 1. The numerical solutions obtained from 200 iterations are shown in

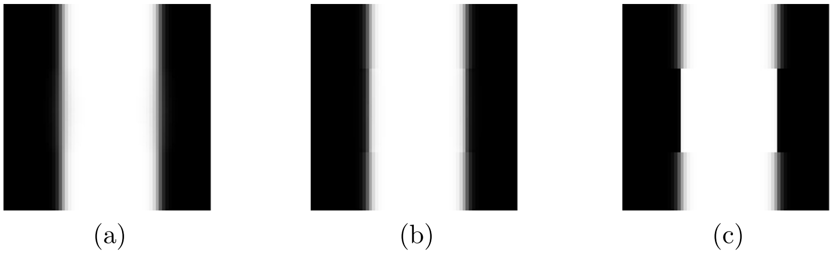


Fig. 6. Inpainting results with different time steps. (a) $\Delta t = 0.1\epsilon^2$, (b) $\Delta t = 2\epsilon^2$, and (c) $\Delta t = 10\epsilon^2$.

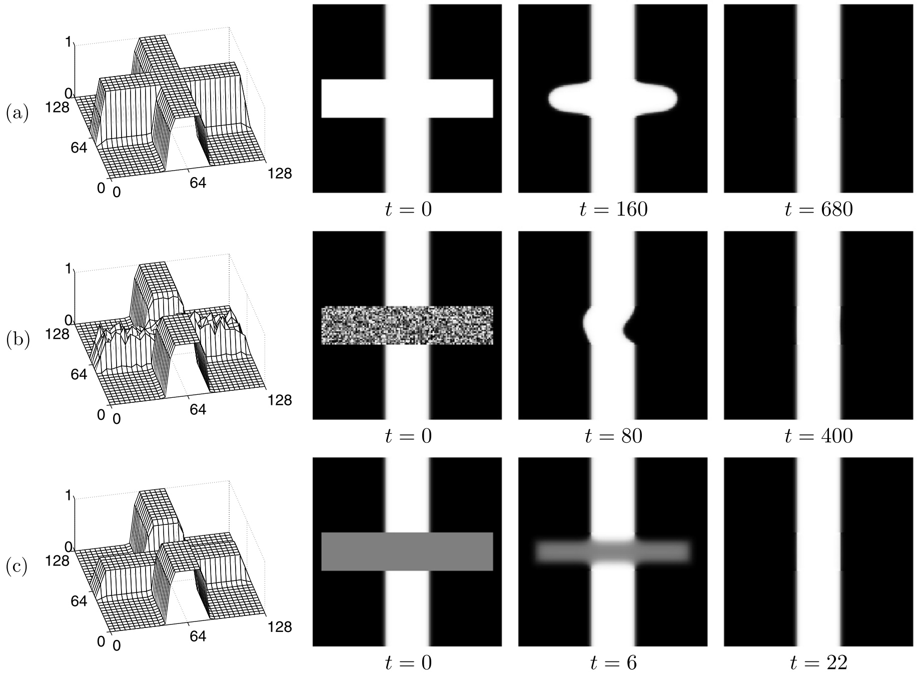


Fig. 7. Evolution of inpainting an image with three different initial conditions with (a) $c(x, y, 0) = 1$, (b) $c(x, y, 0) = \text{rand}(x, y)$, and (c) $c(x, y, 0) = 0.5$ on Ω_D. The first column shows the mesh plot of the initial condition. The other columns are the temporal evolutions with the given initial conditions. The computational times are listed below each figure.

Table 1
Comparison of our proposed method with the method from Bertozzi et al. [18] for the number of iterations and CPU time in seconds.

Case	Image size	Our proposed method		Bertozzi et al.	
		Iterations	CPU time	Iterations	CPU time
Fig. 8(a)	128 × 128	14	0.25	700	11.5
Fig. 8(b)	128 × 128	11	0.21	1000	15.6

Fig. 7(b). In the last test, we consider our proposed initialization algorithm, i.e., $c(x, y, 0) = 0.5$. It takes only 11 iterations to solve the image inpainting problem. Note that, with the three different initial conditions, image inpaintings are successfully done as shown in the last column of Fig. 7.

4.4. Inpainting of a synthetic image

In this section, we show our proposed method can inpaint various synthetic images such as double stripes, crosses, and disks. In this simulation, ϵ_4 is used. The initial configurations are shown in the first two columns of Fig. 8. The gray region in the initial configuration denotes the inpainting region. The inpainted results are shown in the last two columns. As shown in Fig. 8, our proposed method works well. Note that Bertozzi et al. [18] have taken similar numerical tests for double stripes and crosses by using the Cahn–Hilliard equation with an adaptive ϵ. The comparisons with the results in [18] for iterations and CPU time are listed in Table 1. From the results, we can see that our proposed method is more efficient than theirs.

4.5. Gray-valued image inpainting

Figs. 9(a), (b), (c), (d), and (e) show an initial image, the inpainting domain, and the recovered images by using the Chan and Shen's method [12], the Li et al.'s method [17], and the proposed method, respectively. Interface parameter ϵ_{15} and time step $\Delta t = 5$ are used. It took only 4 iterations, which is much faster than Li et al.'s method with 200 iterations and Chan and Shen's method with 5000 iterations for inpainting with Fig. 9(a).

Also, we conduct the inpainting of a zebra and a QR code images, which are considered by Bosch et al. [1]. In Ref. [1], the authors developed a fast solver using the Moreau–Yosida regularization technique for the Cahn–Hilliard inpainting. The results by Bosch et al. [1] are presented in second column of Fig. 10. And we represent the our results in third column for comparison. As shown in Fig. 10, we obtain more clear recovered images of a zebra and QR code with only 8 and 10 iterations than Bosch et al. [1], respectively. These facts imply that our numerical method is also performed well in gray valued inpainting images.

Figs. 11(a) and (b) show two removal of texts. Here, ϵ_{15} and $\Delta t = 5$ are used. As shown in Fig. 11, image inpaintings are suc-

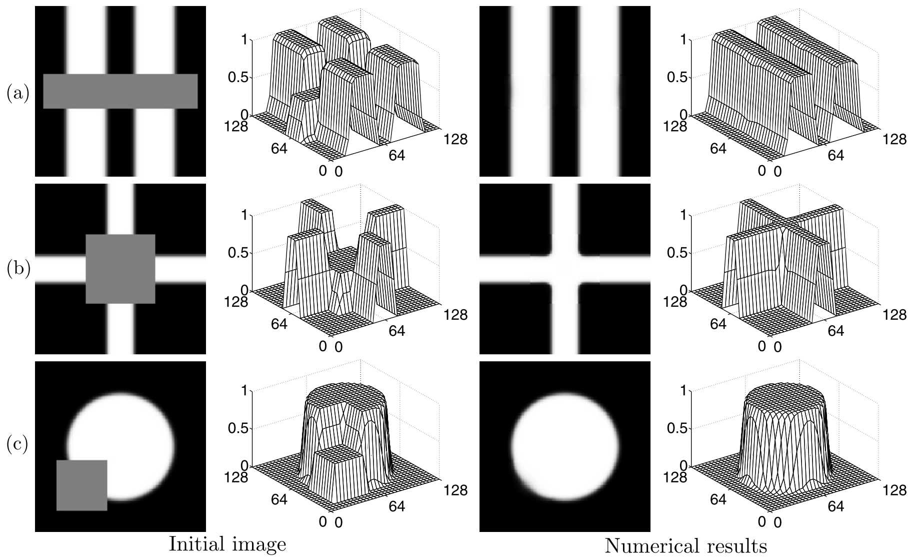


Fig. 8. Inpainting of several synthetic images with a double stripe, cross, and disk deteriorated region. The first two columns are initial images and the last two are inpainted images. Note that the gray region respects inpainting domain.

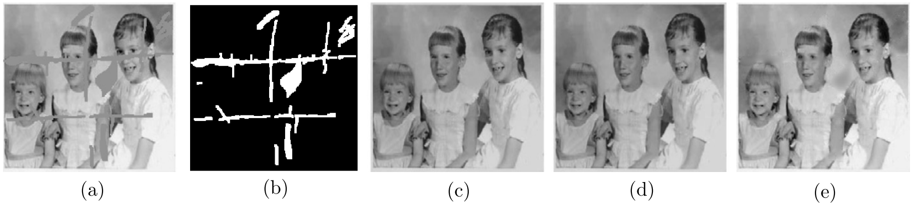


Fig. 9. Inpainting of gray images. (a), (b), (c), (d), and (e) show an initial image, the inpainting domain, and the recovered images by using the Chan and Shen's method [12], the Li et al.'s method [17], and the proposed method, respectively. The initial image is reprinted from F. Li et al. [17].
© 2011 Elsevier. Reprinted with permission. All rights reserved.

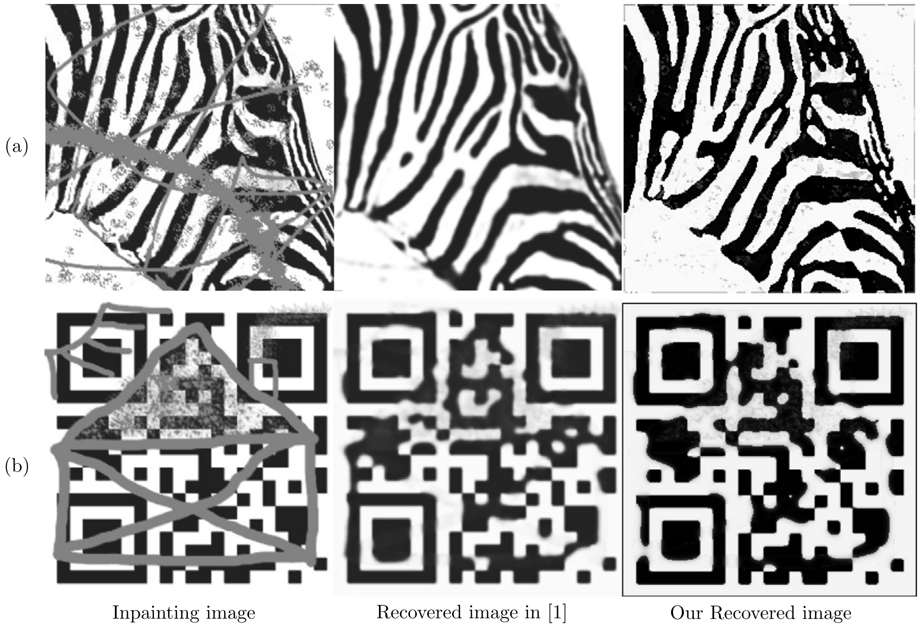


Fig. 10. Image inpainting of (a) zebra and (b) QR code. Inpainting and recovered images are reprinted from Bosch et al. [1].
© 2014 Society for Industrial and Applied Mathematics. Reprinted with permission.

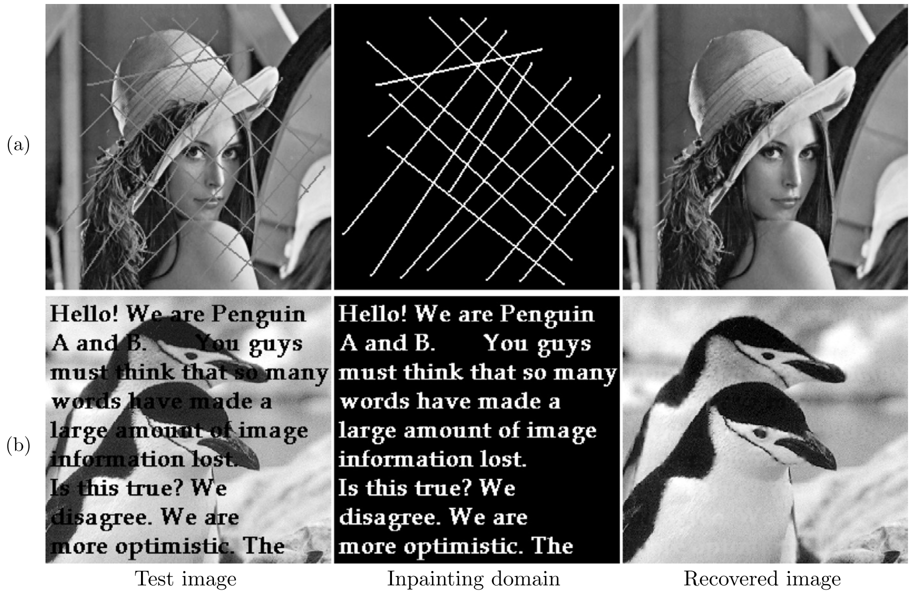


Fig. 11. Examples of (a) Lena and (b) texture images. The initial image of (b) is reprinted from T.F. Chan and J. Shen [12]. © 2002 Society for Industrial and Applied Mathematics. Reprinted with permission. All rights reserved.

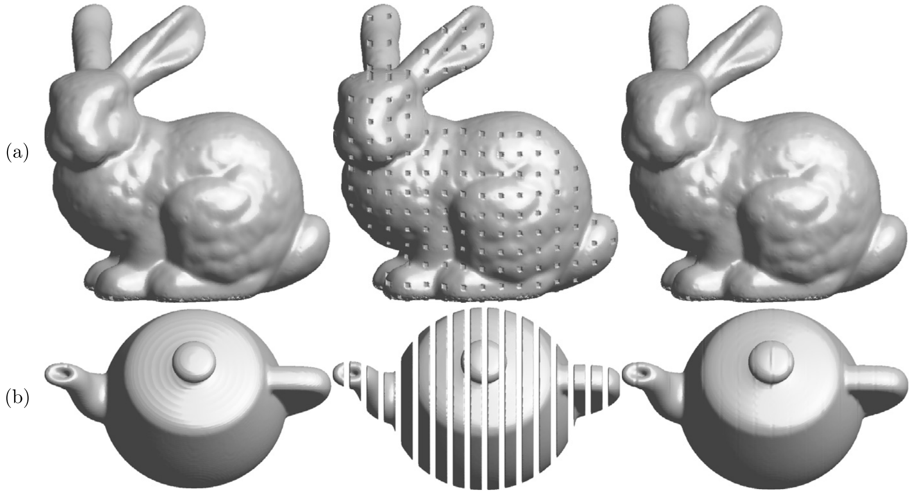


Fig. 12. Examples of (a) bunny and (b) teacup in three-dimensional space. From left to right, they are original, damaged, and recovered surfaces, respectively.

cessfully done. We refer to [36] for an automatic image inpainting algorithm based on a fuzzy C-mean (FCM).

4.6. Three-dimensional inpainting

In this section, we perform an image inpainting in three-dimensional space with our numerical method. It should be noted that in order to use more information from the original image, we use the 27-point discrete Laplacian operator in three-dimensional space [37], i.e.,

$$
\begin{aligned}
\Delta_d c_{ijk} = \big[& 14(c_{i+1,j,k} + c_{i-1,j,k} + c_{i,j+1,k} + c_{i,j-1,k} + c_{i,j,k+1} \\
& + c_{i,j,k-1}) + 3(c_{i+1,j+1,k} + c_{i+1,j-1,k} + c_{i+1,j,k+1} \\
& + c_{i+1,j,k-1} + c_{i-1,j+1,k} + c_{i-1,j-1,k} + c_{i-1,j,k+1} \\
& + c_{i-1,j,k-1} + c_{i,j+1,k+1} + c_{i,j-1,k-1} + c_{i,j+1,k-1} \\
& + c_{i,j-1,k+1}) + c_{i-1,j+1,k+1} + c_{i-1,j-1,k-1} \\
& + c_{i-1,j+1,k-1} + c_{i-1,j-1,k+1} + c_{i+1,j+1,k+1} \\
& + c_{i+1,j-1,k-1} + c_{i+1,j+1,k-1} + c_{i+1,j-1,k+1} \\
& - 128 c_{ijk} \big] / \big(30h^2\big).
\end{aligned}
$$

Figs. 12(a) and (b) show two 3D examples. From left to right, they are original, damaged, and recovered surfaces, respectively. Here, ϵ_4 and $\Delta t = 2$ are used. As can be seen, our proposed method performs well for inpainting in three-dimensional space.

4.7. Computational cost

Now, we show the performance of all test problems which were done in previous sections. Table 2 shows the image size (N), the inpainting domain size (N_{is}), the number of iterations, and the CPU time in each test. The result indicates that our proposed method is robust and efficient. It should be noted that since our proposed model is defined in the inpainting domain, the computational cost is not $O(N \log N)$ but $O(N_{is} \log N_{is})$. To demonstrate it, in Fig. 13, we plot the average CPU time (total CPU time over total iterations) and $O(N_{is} \log N_{is})$, which are summarized in Table 2. Here, the linear fitting is done using the MATLAB function "polyfit". We define the error as the difference between the average CPU time and the linear fit result. In this test, the l_2 error is 0.89%. Therefore the results suggest that the convergence rate of the computational cost is linear with respect to $N_{is} \log N_{is}$. To show the effect and accuracy of our proposed method, we will compute PSNR (Peak Signal to Noise Ratio) [38–41], which is commonly used to measure the quality of reconstruction of image inpainting. Here, PSNR is defined as

$$\mathrm{PSNR} = -10 \log_{10}\left(\frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \left[c_{ij}^n - (f_{ij} - f_{\min}) \right.\right.$$
$$\left.\left. /(f_{\max} - f_{\min}) \right]^2 \right).$$

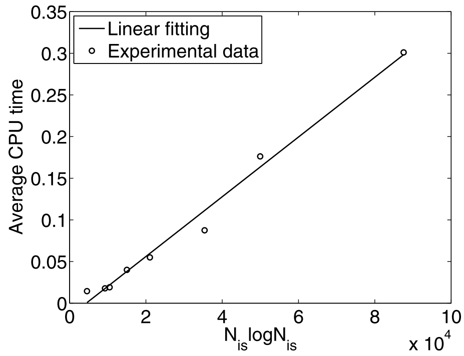


Fig. 13. Experimental data and linear fitting average CPU time versus $N_{is} \log N_{is}$.

Table 2
Performance of our proposed method in image inpainting for several cases. Note that N_{is} means the number of pixels which are on the inpainting domain Ω_D.

Case	Image size (N)	N_{is}	Iterations	CPU time (s)	PSNR
Fig. 8(a)	128 × 128	3016	14	0.25	43.29
Fig. 8(b)	128 × 128	2704	11	0.21	35.52
Fig. 8(c)	128 × 128	1444	11	0.16	35.43
Fig. 9	231 × 229	5614	4	0.35	
Fig. 10(a)	256 × 256	12,223	8	1.41	
Fig. 10(b)	294 × 293	20,321	10	3.01	
Fig. 11(a)	240 × 256	8952	4	0.22	32.93
Fig. 11(b)	256 × 256	4152	4	0.16	
Fig. 12(a)	254 × 250 × 200	819,200	9	116.89	44.21
Fig. 12(b)	277 × 183 × 147	1,371,951	11	163.54	37.39

Since the pixel values in the inpainting domain are obtained by curvature-driven diffusions and the pixel values outside the inpainting region are the same as those in the original input image, we can expect the higher PSNR obtained by our proposed method.

4.8. Disconnected stripe image with large gap

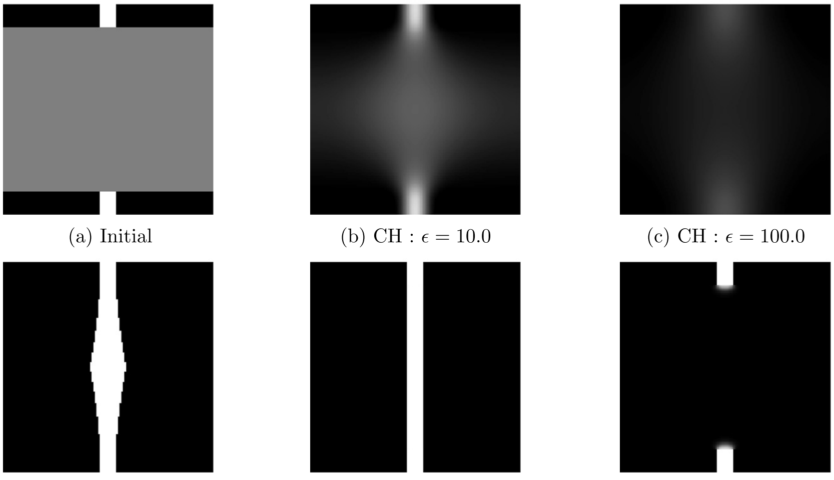
In this section, we test a disconnected stripe image with large inpainting region as the benchmark example [42]. First, we recover the example (see Fig. 14(a)) by using the modified Cahn–Hilliard equation. For this test, we use $\lambda = 1$, $\Delta t = 1$, and $h = 1$. Numerical results with $\epsilon = 10$ and 100 are shown in Figs. 14(b) and (c), respectively.

From these results, we know that the inpainting problem with large gap is difficult to recover with the modified CH equation. In order to resolve this problem which is arising in Cahn–Hilliard model, authors in [18,19,42] proposed the two-step method. This method is first performed with a large ϵ, then used with small ϵ to regulate the diffuse-interface motion. For more details about the two-step method, see the reference [18,19,42].

Also, we conduct the test with new method proposed in this paper. In this test, we use $\Delta t = 0.01$ and $h = 1$. And we do test with three different values of ϵ. Figs. 14(d)–(f) show the numerical results with $\epsilon = 0.01, 0.1,$ and 1 at steady states, respectively. Here, we define the steady state as $\|\mathcal{E}_{B1}(\phi^{n+1}) - \mathcal{E}_{B1}(\phi^n)\| < 10^{-5}$. When we use $\epsilon = 0.1$, we get the good numerical result as the one in [42].

5. Conclusion

In this paper, we proposed a new unconditionally stable hybrid numerical method for image inpainting. The model is based on the local Allen–Cahn equation and an operator splitting technique was used to solve the model numerically. We proved the unconditional stability of the proposed scheme. Various numerical results on real and synthetic images were presented to demonstrate the efficiency and robustness of the proposed method. Since our proposed model is defined in the inpainting domain, the computational cost is not $O(N \log N)$ but $O(N_{is} \log N_{is})$. This indicates that our proposed method is very fast for image inpainting process. Through the computational cost comparisons with three models in [12,17,18], we confirmed that our proposed model is much faster than the other existing models.


(a) Initial (b) CH : $\epsilon = 10.0$ (c) CH : $\epsilon = 100.0$

(d) AC : $\epsilon = 0.01$ (e) AC : $\epsilon = 0.1$ (f) AC : $\epsilon = 1.0$

Fig. 14. Disconnected stripe image with large gap inpainting region. (a) Initial condition and (b)–(f) numerical results by modified CH model and AC model with respect to ϵ.

Conflict of interest statement

The authors declare that there is no conflict of interests regarding the publication of this article.

Acknowledgments

The second author (D. Jeong) was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2014R1A6A3A01009812). The corresponding author (J.S. Kim) was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and future Planning (NRF-2014R1A2A2A01003683). The authors are grateful to the reviewers whose valuable suggestions and comments significantly improved the quality of this paper.

References

[1] J. Bosch, D. Kay, M. Stoll, A. Wathen, Fast solvers for Cahn–Hilliard inpainting, SIAM J. Imaging Sci. 7 (1) (2014) 67–97.
[2] C. Guillemot, O. Le Meur, Image inpainting: overview and recent advances, IEEE Signal Process. Mag. 31 (1) (2014) 127–144.
[3] C. Qin, F. Cao, X.P. Zhang, Efficient image inpainting using adaptive edge-preserving propagation, Imaging Sci. J. 59 (4) (2011) 211–218.
[4] C. Qin, S. Wang, X. Zhang, Simultaneous inpainting for image structure and texture using anisotropic heat transfer model, Multimed. Tools Appl. 56 (3) (2012) 469–483.
[5] Y. Liu, V. Caselles, Exemplar-based image inpainting using multiscale graph cuts, IEEE Trans. Image Process. 22 (5) (2013) 1699–1711.
[6] J. Patel, T.K. Sarode, Exemplar based image inpainting with reduced search region, Int. J. Comput. Appl. 92 (12) (2014) 27–33.
[7] J. Wang, K. Lu, D. Pan, N. He, B.-K. Bao, Robust object removal with an exemplar-based image inpainting approach, Neurocomputing 123 (2014) 150–155.
[8] S.R. Gaonkar, P.D. Hire, P.S. Pimple, Y.R. Kotwal, B.A. Ahire, Image inpainting using robust exemplar-based technique, Int. J. Comput. Sci. Eng. 2 (4) (2014) 176–179.
[9] K.N. Ramamurthy, J.J. Thiagarajan, A. Spanias, Recovering non-negarive and combined sparse representations, Digit. Signal Process. 26 (2014) 21–35.
[10] J.S. Turek, I. Yavneh, M. Elad, On MAP and MMSE estimators for the co-sparse analysis model, Digit. Signal Process. 28 (2014) 57–74.
[11] M. Burger, L. He, C. Schönlieb, Cahn–Hilliard inpainting and a generalization for gray-value images, SIAM J. Imaging Sci. 2 (4) (2009) 1129–1167.
[12] T.F. Chan, J. Shen, Mathematical models for local non-texture inpaintings, SIAM J. Appl. Math. 62 (3) (2002) 1019–1043.
[13] S. Esedoḡlu, J. Shen, Digital inpainting based on the Mumford–Shah–Euler image model, Eur. J. Appl. Math. 13 (4) (2002) 353–370.
[14] T.F. Chan, J. Shen, Nontexture inpainting by curvature-driven diffusions, J. Vis. Commun. Image Represent. 12 (4) (2001) 436–449.
[15] D. Mumford, J. Shah, Optimal approximation by piecewise smooth functions and associated variational problems, Commun. Pure Appl. Math. 42 (5) (1989) 577–685.
[16] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, J. Verdera, Filling-in by joint interpolation of vector fields and gray levels, IEEE Trans. Image Process. 10 (8) (2001) 1200–1211.
[17] F. Li, C. Shen, R. Liu, J. Fan, A fast implementation algorithm of TV inpainting model based on operator splitting method, Comput. Electr. Eng. 37 (5) (2011) 782–788.
[18] A. Bertozzi, S. Esedoḡlu, A. Gillette, Inpainting of binary images using the Cahn–Hilliard equation, IEEE Trans. Image Process. 16 (1) (2007) 285–291.
[19] A. Bertozzi, S. Esedoḡlu, A. Gillette, Analysis of a two-scale Cahn–Hilliard model for binary image inpainting, Multiscale Model. Simul. 6 (3) (2007) 913–936.
[20] J.W. Cahn, J.E. Hilliard, Free energy of a nonuniform system, I: interfacial free energy, J. Chem. Phys. 28 (2) (1958) 258–267.
[21] D. Lee, J.Y. Huh, D. Jeong, J. Shin, A. Yun, J. Kim, Physical, mathematical, and numerical derivations of the Cahn–Hilliard equation, Comput. Mater. Sci. 81 (2014) 216–225.
[22] S.M. Allen, J.W. Cahn, A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening, Acta Mater. 27 (6) (1979) 1085–1095.
[23] Y. Li, J. Kim, An unconditionally stable hybrid method for image segmentation, Appl. Numer. Math. 82 (2014) 32–43.
[24] T. Chan, S. Esedoḡlu, M. Nikolova, Algorithms for finding global minimizers of image segmentation and denoising models, SIAM J. Appl. Math. 66 (5) (2006) 1632–1648.
[25] S. Esedoḡlu, Y.-H.R. Tsai, Threshold dynamics for the piecewise constant Mumford–Shah functional, J. Comput. Phys. 211 (1) (2006) 367–384.
[26] J. Lie, M. Lysaker, X.C. Tai, A binary level set model and some applications to Mumford–Shah image segmentation, IEEE Trans. Image Process. 15 (5) (2006) 1171–1181.
[27] L. Blanc-Feraud, G. Aubert, J. Zerubia, A variational model for image classification and restoration, IEEE Trans. Pattern Anal. Mach. Intell. 22 (5) (2000) 460–472.
[28] Y. Li, H.G. Lee, D. Jeong, J. Kim, An unconditionally stable hybrid numerical method for solving the Allen–Cahn equation, Comput. Math. Appl. 60 (6) (2010) 1591–1606.
[29] Y. Li, J. Kim, Multiphase image segmentation with a phase-field model, Comput. Math. Appl. 62 (2) (2011) 737–745.
[30] A. Stuart, A.R. Humphries, in: Dynamical System and Numerical Analysis, Cambridge University Press, Cambridge, UK, 1998.
[31] Y. Li, H.G. Lee, J. Kim, A fast, robust, and accurate operator splitting method for phase-field simulations of crystal growth, J. Cryst. Growth 321 (1) (2011) 176–182.
[32] Y. Li, J. Kim, A fast and accurate numerical method for medical image segmentation, J. Korea SIAM. 14 (4) (2010) 201–210.
[33] B. Zhang, S. Makram-Ebeid, R. Prevost, G. Pizaine, Fast solver for some computational imaging problems: a regularized wighted least-squares approach, Digit. Signal Process. 27 (2014) 107–118.
[34] K.W. Morton, D.E. Mayers, in: Numerical Solution of Partial Differential Equations, Cambridge University Press, Cambridge, UK, 1996.
[35] J. Kim, Phase-field models for multi-component fluid flows, Commun. Comput. Phys. 12 (3) (2012) 613–661.
[36] J. Liu, H. Liu, S. Qiao, G. Yue, An automatic image inpainting algorithm based on FCM, Sci. World J. 2014 (2014) 201704.
[37] W.F. Spotz, G.F. Carey, A high-order compact formulation for the 3D Poisson equation, Numer. Methods Partial Differ. Equ. 12 (1996) 235–243.
[38] P.A. Hernandez-Avalos, C. Feregrino-Uribe, R. Cumplido, Watermarking using similarities based on fractal codification, Digit. Signal Process. 22 (2013) 324–336.
[39] A. Pizurica, W. Philips, Estimating the probability of the presence of a signal of interest in multiresolution single- and multibrand image denoising, IEEE Trans. Image Process. 15 (3) (2006) 654–665.
[40] C. Qin, C.-C. Chang, Y.-P. Chiu, A novel joint data-hiding and compression scheme based on SMVQ and image inpainting, IEEE Trans. Image Process. 23 (3) (2014) 969–978.
[41] M. Yan, Restoration of images corrupted by impulse noise and mixed Gaussian impulse noise using blind inpainting, SIAM J. Imaging Sci. 6 (3) (2013) 1227–1245.
[42] A. Gillette, Image inpainting using a modified Cahn-Hilliard equation, doctoral dissertation, University of California Los Angeles, 2006.

Yibao Li received the M.S. and Ph.D. degrees in Applied Mathematics from Korea University, Korea, in 2011 and 2013, respectively. He held a research position in Department of Computational Science and Engineering, Yonsei University, Korea (2013–2014). Currently he is an assistant professor at the School of Mathematics and Statistics of Xi'an Jiaotong University, China. His research interests include image processing, phase field model, computational fluid dynamics, and scientific computing.

Darae Jeong received her Ph.D. in Applied Mathematics from Korea University, Korea, in 2013. And she is received his M.S. degree in Applied Mathematics and his B.S. degree in Mathematics from Korea University in 2011 and Dongguk University in 2008, respectively. She is currently working as a post-doctoral research fellow at the Department of Mathematics, Korea University, Korea. Her research interests include computational finance, computational fluid dynamics, and scientific computation.

Jung-il Choi received the B.Sc, M.Sc. and Ph.D. in mechanical engineering from KAIST in 1994, 1996, and 2002, respectively. He is currently an associate professor at the Department of Computational Science and Engineering, Yonsei University, Korea. His research interests lie in the field of computational fluid dynamics and its application to various thermo-fluid engineering problems.

Seunggyu Lee received the B.S. degree in mathematical sciences in 2011 from KAIST. Now, he is a Ph.D. candidate in the Department of Mathematics, Korea University. His research interests include numerical analysis and scientific computing in fluid dynamics, bioscience, and image processing.

Junseok Kim received the received his Ph.D. in Applied Mathematics from the University of Minnesota, U.S.A. in 2002. He also received his B.S. degree from the Department of Mathematics Education, Korea University, Korea in 1995. He joined the faculty of Korea University, Korea in 2008 where he is currently an associate professor at the Department of Mathematics. His research interests are in image processing and scientific computation.