

Research Article

Curve and Surface Smoothing Using a Modified Cahn-Hilliard Equation

Yongho Choi, Darea Jeong, and Junseok Kim

Department of Mathematics, Korea University, Seoul 02841, Republic of Korea

Correspondence should be addressed to Junseok Kim; cfdkim@korea.ac.kr

Received 19 April 2017; Revised 23 July 2017; Accepted 14 September 2017; Published 25 October 2017

Academic Editor: Ruben Specogna

Copyright © 2017 Yongho Choi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a new method using the modified Cahn-Hilliard (CH) equation for smoothing piecewise linear shapes of two- and three-dimensional objects. The CH equation has good smoothing dynamics and it is coupled with a fidelity term which keeps the original given data; that is, it does not produce significant shrinkage. The modified CH equation is discretized using a linearly stable splitting scheme in time and the resulting scheme is solved by using a Fourier spectral method. We present computational results for both curve and surface smoothing problems. The computational results demonstrate that the proposed algorithm is fast and efficient.

1. Introduction

For many computational purposes such as visualization of scientific data and registration of multimodal medical data, smooth curves and surfaces must be approximated by polygonal curves and polyhedral surfaces, respectively. However, these approximation algorithms generate faceted curves and surfaces [1]. In medical applications, Vollmer et al. [2] presented an improved Laplacian technique for smoothing polygonal surface meshes that avoids the well-known problem of deformation and shrinkage caused by many smoothing methods as shown in Figure 1. Using Bezier-smoothed boundary representation, Anflor and Marczak [3] studied topological optimization of anisotropic heat conducting devices.

To prevent the mesh from shrinking, Liu et al. [4] proposed a volume-constrained smoothing algorithm for triangular meshes, which preserves exactly the mesh volume during the smoothing process. Wei et al. [5] proposed an effective mesh smoothing algorithm which preserves morphology on polygonized isosurfaces of inhomogeneous binary volumes. Recently, various smoothing methods have been developed [6, 7].

In this paper, we propose a new method using a modified Cahn-Hilliard (CH) equation for smoothing piecewise linear

shapes of two- and three-dimensional objects. Let $\psi(\mathbf{x})$ be a given mosaic binary data, that is, either plus one or minus one. Here, $\mathbf{x} = (x, y)$ or $\mathbf{x} = (x, y, z)$. We want to construct smooth contour line or isosurface from the mosaic binary data by using the modified CH equation:

$$\begin{aligned} \phi_t(\mathbf{x}, t) = \Delta \left[F'(\phi(\mathbf{x}, t)) - \varepsilon^2 \Delta \phi(\mathbf{x}, t) \right] \\ + \lambda (\psi(\mathbf{x}) - \phi(\mathbf{x}, t)), \end{aligned} \quad (1)$$

where λ is the fidelity parameter and $F(\phi) = 0.25(\phi^2 - 1)^2$. When $\lambda = 0$, then (1) becomes the CH equation which was originally proposed by Cahn and Hilliard [8, 9] as a phenomenological model of phase separation in a binary alloy. See [10] and the references therein for physical, mathematical, and numerical derivations of the CH equation. The various applications of the CH equation can be seen in [11]. Equation (1) was used in image inpainting with a space-dependent $\lambda(\mathbf{x})$ [12].

This paper is organized as follows. In Section 2, the discrete equations for the governing equations are presented. In Section 3, we present various computational examples. Finally, in Section 4, conclusions are drawn.

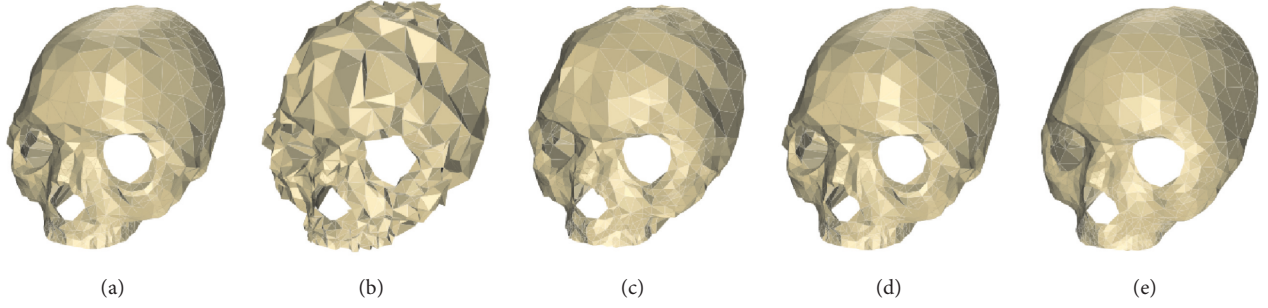


FIGURE 1: (a)–(e) Origin, noisy mesh, three iterations of the algorithm in [2]. Reprinted from Vollmer et al. [2] with permission from the Wiley Online Library.

2. Algorithm

We use an unconditionally stable Fourier spectral method for the modified CH equation (1) in two-dimensional space $\Omega = (a, b) \times (c, d)$. Fourier spectral method has been applied in many studies [13, 14]. The three-dimensional discretization is defined analogously. Let $h = (b - a)/N_x = (d - c)/N_y$ be the spatial step size, where N_x and N_y are positive even integers. Let ϕ_{mn}^k and μ_{mn}^k be approximations of $\phi(x_m, y_n, t_k)$ and $\mu(x_m, y_n, t_k)$, respectively, where $(x_m, y_n) = (a + (m - 0.5)h, b + (n - 0.5)h)$ and $t_k = k\Delta t$ with Δt the temporal step. Let $\hat{\phi}_{pq}^k = \sum_{m=1}^{N_x} \sum_{n=1}^{N_y} \phi_{mn}^k e^{-i(\xi_p x_m + \eta_q y_n)}$ be the discrete Fourier transform with a given data $\{\phi_{mn}^k \mid m = 1, \dots, N_x \text{ and } n = 1, \dots, N_y\}$, where $\xi_p = 2\pi p/(b - a)$ and $\eta_q = 2\pi q/(d - c)$. The inverse discrete Fourier transform is

$$\phi_{mn}^k = \frac{1}{N_x N_y} \sum_{p=-N_x/2}^{N_x/2-1} \sum_{q=-N_y/2}^{N_y/2-1} \hat{\phi}_{pq}^k e^{i(\xi_p x_m + \eta_q y_n)}. \quad (2)$$

Let $\phi(x, y, k\Delta t) = (1/N_x N_y) \sum_{p=-N_x/2}^{N_x/2-1} \sum_{q=-N_y/2}^{N_y/2-1} \hat{\phi}_{pq}^k e^{i(\xi_p x + \eta_q y)}$. Then, we have

$$\begin{aligned} \Delta \phi(x, y, k\Delta t) &= -\frac{1}{N_x N_y} \sum_{p=-N_x/2}^{N_x/2-1} \sum_{q=-N_y/2}^{N_y/2-1} (\xi_p^2 + \eta_q^2) \hat{\phi}_{pq}^k e^{i(\xi_p x + \eta_q y)}. \end{aligned} \quad (3)$$

By using the linearly stabilized splitting scheme [15] to (1), we have

$$\begin{aligned} \frac{\phi_{mn}^{k+1} - \phi_{mn}^k}{\Delta t} &= \Delta (2\phi_{mn}^{k+1} - \varepsilon^2 \Delta \phi_{mn}^{k+1} + f(\phi_{mn}^k)) \\ &\quad + \lambda (\psi_{mn} - \phi_{mn}^{k+1}), \end{aligned} \quad (4)$$

where $f(\phi) = \phi^3 - 3\phi$. Thus, (4) can be transformed into the discrete Fourier space as follows:

$$\begin{aligned} \frac{\hat{\phi}_{pq}^{k+1} - \hat{\phi}_{pq}^k}{\Delta t} &= -(\xi_p^2 + \eta_q^2) (2\hat{\phi}_{pq}^{k+1} + \varepsilon^2 (\xi_p^2 + \eta_q^2) \hat{\phi}_{pq}^{k+1} + \hat{f}_{pq}^k) \\ &\quad + \lambda (\hat{\psi}_{pq} - \hat{\phi}_{pq}^{k+1}). \end{aligned} \quad (5)$$

Therefore, we obtain the following discrete Fourier transform:

$$\hat{\phi}_{pq}^{k+1} = \frac{\hat{\phi}_{pq}^k - (\xi_p^2 + \eta_q^2) \Delta t \hat{f}_{pq}^k + \lambda \hat{\psi}_{pq}}{1 + \Delta t [\lambda + 2(\xi_p^2 + \eta_q^2) + \varepsilon^2 (\xi_p^2 + \eta_q^2)^2]}. \quad (6)$$

Then, using (2), we get the updated numerical solution ϕ_{mn}^{k+1} as

$$\phi_{mn}^{k+1} = \frac{1}{N_x N_y} \sum_{p=-N_x/2}^{N_x/2-1} \sum_{q=-N_y/2}^{N_y/2-1} \hat{\phi}_{pq}^{k+1} e^{i(\xi_p x_m + \eta_q y_n)}. \quad (7)$$

Standard fast Fourier transform (FFT) has been used to speed up the computations [16].

3. Computational Results and Discussion

3.1. Circular Mosaic Shape. In this test problem, we will show the smoothing effect of our proposed method. Therefore, we choose a coarse mesh grid 20×20 on the computational domain $\Omega = (-1, 1) \times (-1, 1)$. The initial configuration is

$$\phi(x, y, 0) = \begin{cases} 1 & \text{if } \sqrt{x^2 + y^2} \leq 0.6, \\ -1 & \text{otherwise,} \end{cases} \quad (8)$$

which represents a circular mosaic shape. The numerical parameters $h = 0.1$, $\varepsilon = h$, $\Delta t = 0.1$, and $\lambda = 1000$ are used.

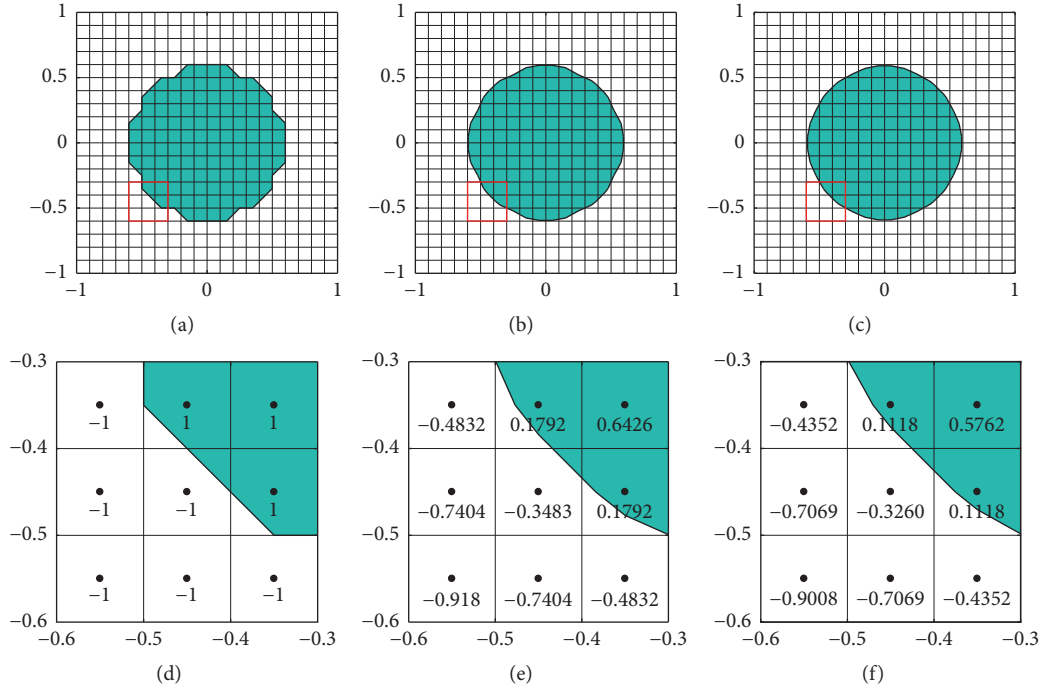


FIGURE 2: (a) Initial circular mosaic shape. (b)-(c) Numerical results after 1 and 10 iterations by the proposed method, respectively. (d)–(f) Values of $\phi(x, y)$ at the red box.

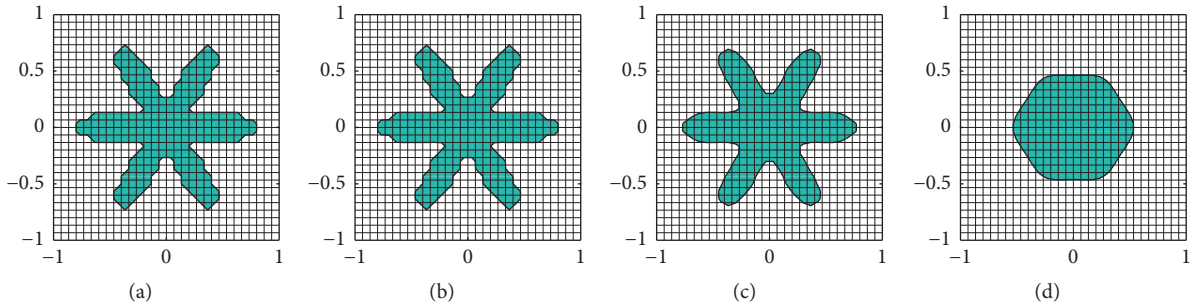


FIGURE 3: Effect of λ on the smoothing dynamics. (a) Initial condition and numerical results at $T = 20\Delta t$ with (b) $\lambda = 10^6$, (c) $\lambda = 10^4$, and (d) $\lambda = 10^2$.

Figure 2 shows the temporal evolution at $t = 0, \Delta t$, and $10\Delta t$. Here, the shaded area is the filled contour with $\phi \geq 0$. Figures 2(d)–2(f) show the values of $\phi(x, y)$ at the part of the red box in Figures 2(a)–2(c). As shown in Figure 2(c), we can see the smooth interface after 10 iterations.

3.2. *Effect of λ .* Next, we investigate the effect of parameter λ on the smoothing dynamics. On the computational domain $\Omega = (-1, 1) \times (-1, 1)$, the initial configuration is given as

$$\phi(x, y, 0) = \begin{cases} 1 & \text{if } \sqrt{x^2 + y^2} + 0.3 \cos(6\theta) \leq 0.5, \\ -1 & \text{otherwise,} \end{cases} \quad (9)$$

where $\theta = \tan^{-1}(y/x)$ if $x > 0$; $\theta = \pi + \tan^{-1}(y/x)$ otherwise. We use the following parameters: $h = 2/30$, $\varepsilon = h$, and $\Delta t = 0.1$. With the initial configuration as shown in Figure 3(a), we obtain the various numerical results (see Figures 3(b)–3(d)) at time $T = 20\Delta t$ with respect to different λ . In this test, we use three different $\lambda = 10^6, 10^4$, and 10^2 . As λ is large, $\phi(x, y)$ does not evolve and is close to the initial mosaic profile. However, if λ is too small, then the CH dynamics dominates and the profile becomes a circular shape which has the minimum interfacial length.

3.3. *Effect of ε .* In our proposed model, parameter ε determines the thickness of the interface of two phases, that is, interfacial region of transition profile. Therefore, it is one of

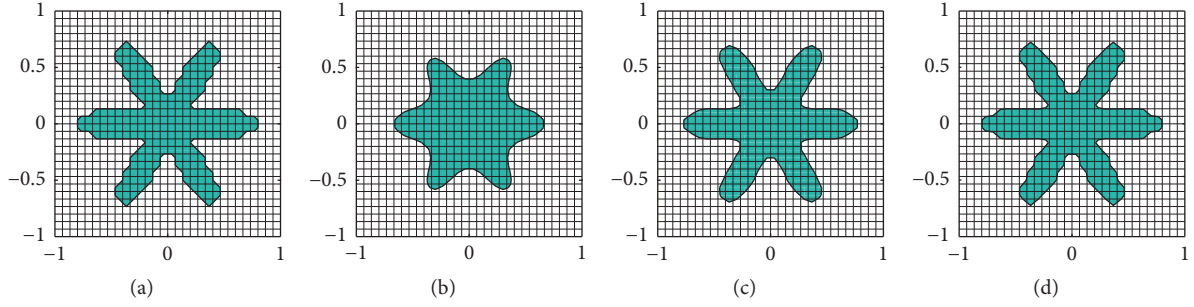


FIGURE 4: Effect of ε on the smoothing dynamics. (a) Initial condition and numerical results at $T = 20\Delta t$ with (b) $\varepsilon = 4h$, (c) $\varepsilon = h$, and (d) $\varepsilon = h/4$.

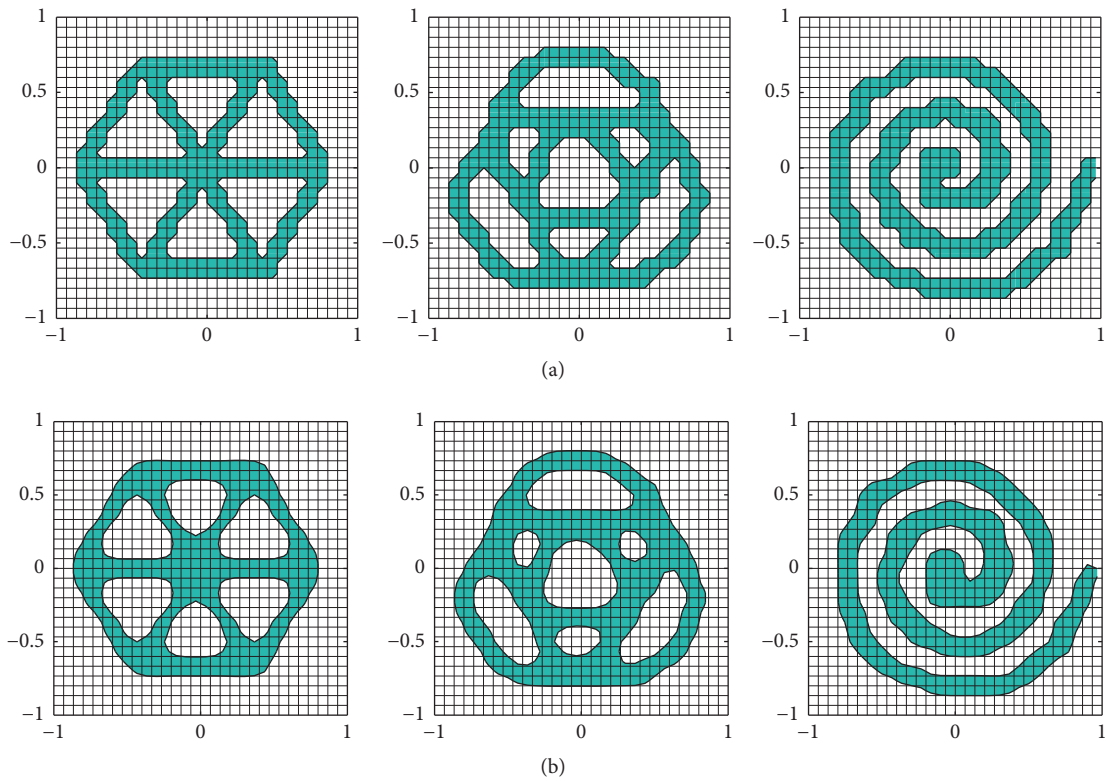


FIGURE 5: (a) Three initial configurations. (b) Corresponding smooth results after 10 iterations.

the important factors on the smoothing process. Now, we study the effect of ε on the smoothing dynamics.

On the computational domain $\Omega = (-1, 1) \times (-1, 1)$, we use the initial condition (9); see Figure 4(a). The other parameters are used as $h = 2/30$, $\Delta t = 0.1$, and $\lambda = 10000$.

Figures 4(b)–4(d) represent the numerical configurations at $T = 20\Delta t$ with $\varepsilon = 4h$, $\varepsilon = h$, and $\varepsilon = h/4$, respectively. As shown in Figure 4(b), large value of ε causes wide transition layer. Therefore, it overly smooths the curve. On the other hand, if ε is too small, then the phase-field has abrupt transition and the profile becomes mosaic.

3.4. 2D Various Shapes. To demonstrate the robustness of the proposed algorithm, we consider various complex shapes on two-dimensional space $\Omega = (-1, 1) \times (-1, 1)$. For the numerical test, we use $N = 30$, $h = 2/N$, $\varepsilon = h$, $\lambda = 10000$, and $\Delta t = 0.1$. The initial configurations are represented in Figure 5(a) and its corresponding smooth results in Figure 5(b). Through these results, we can see that our proposed algorithm is a good strategy to remove staircases while preserving the initial feature.

3.5. 3D Various Shapes. In this section, we consider various morphologies on the three-dimensional space.

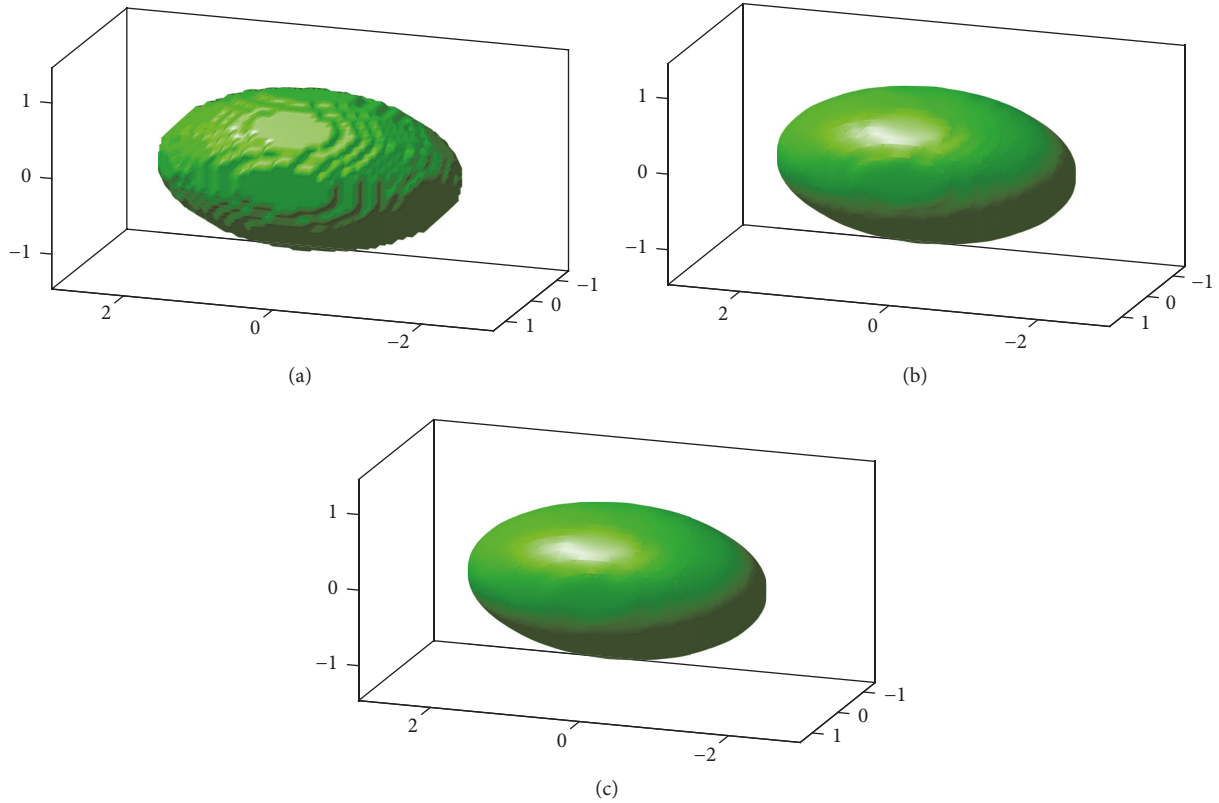


FIGURE 6: (a) Initial condition. Numerical results after (b) 1 and (c) 10 iterations.

3.6. Ellipsoidal Shape. We consider an ellipsoidal shape on the computational domain $\Omega = (-3, 3) \times (-1.5, 1.5) \times (-1.5, 1.5)$. The initial condition is set to

$$\phi(x, y, z, 0) = \begin{cases} 1 & \text{if } \sqrt{\frac{x^2}{4} + y^2 + z^2} \leq 1, \\ -1 & \text{otherwise.} \end{cases} \quad (10)$$

In the test problem, we use $h = 0.075$, $\varepsilon = 5h$, $\Delta t = 0.1$, and $\lambda = 1000$ on an $80 \times 40 \times 40$ mesh grid. Figures 6(a)–6(c) show the temporal evolutions of the surface at $T = 0$, Δt , and $10\Delta t$. As shown in Figure 6, the proposed method leads to a smooth ellipsoidal surface.

3.7. Volume Preservation. Laplacian smoothing is a well-known algorithm to smooth a polygonal mesh. However, this method can cause volume shrinkage or shape distortion [5]. However, our proposed method is a feature-preserving approach without severe volume changes. In this section, we demonstrate that our smoothing approach preserves the initial volume. We first define the volume of an object using triangles on the surface. For $s = 1, 2, \dots, M$, let $\mathbf{Tri}_s = (\mathbf{X}_l, \mathbf{X}_m, \mathbf{X}_n) = ((X_l, Y_l, Z_l), (X_m, Y_m, Z_m), (X_n, Y_n, Z_n))$ be a triangle consisting of three points on a triangular mesh. Here,

M is the number of triangles. We have the following formula for the volume of the polyhedron as

$$V(\mathbf{X}) = \frac{1}{6} \sum_{s=1}^N [X_n (Y_l Z_m - Y_m Z_l) - Y_n (X_l Z_m - X_m Z_l) + Z_n (X_l Y_m - X_m Y_l)]. \quad (11)$$

See [17] for more detailed explanation about the polyhedron volume. Also, we define the percent of relative error of volume $V(\mathbf{X})$ with the initial volume $V(\mathbf{X}^0)$ as

$$V_{\text{err}}(\mathbf{X}) = \frac{V(\mathbf{X}) - V(\mathbf{X}^0)}{V(\mathbf{X}^0)} \times 100 (\%). \quad (12)$$

Now, we consider three-dimensional morphologies such as to holed tetrahedron, many spheres, and holed sphere on $\Omega = (-1, 1) \times (-1, 1) \times (-1, 1)$; see Figure 7(a). On a $40 \times 40 \times 40$ mesh grid, we perform the numerical test with $h = 0.1$, $\varepsilon = h$, $\Delta t = 0.1$, and $\lambda = 5000$. We obtain smooth results after 10 iteration steps by our proposed method as shown in Figure 7(b). Also, to show the effectiveness of our method, we compute the temporal evolutions of the relative percent error; see Figure 7(c).

Next, to show the robustness of the proposed method, we perform surface smoothing problem with noise in Figure 8. The initial conditions are the same as in Figure 7 with 30

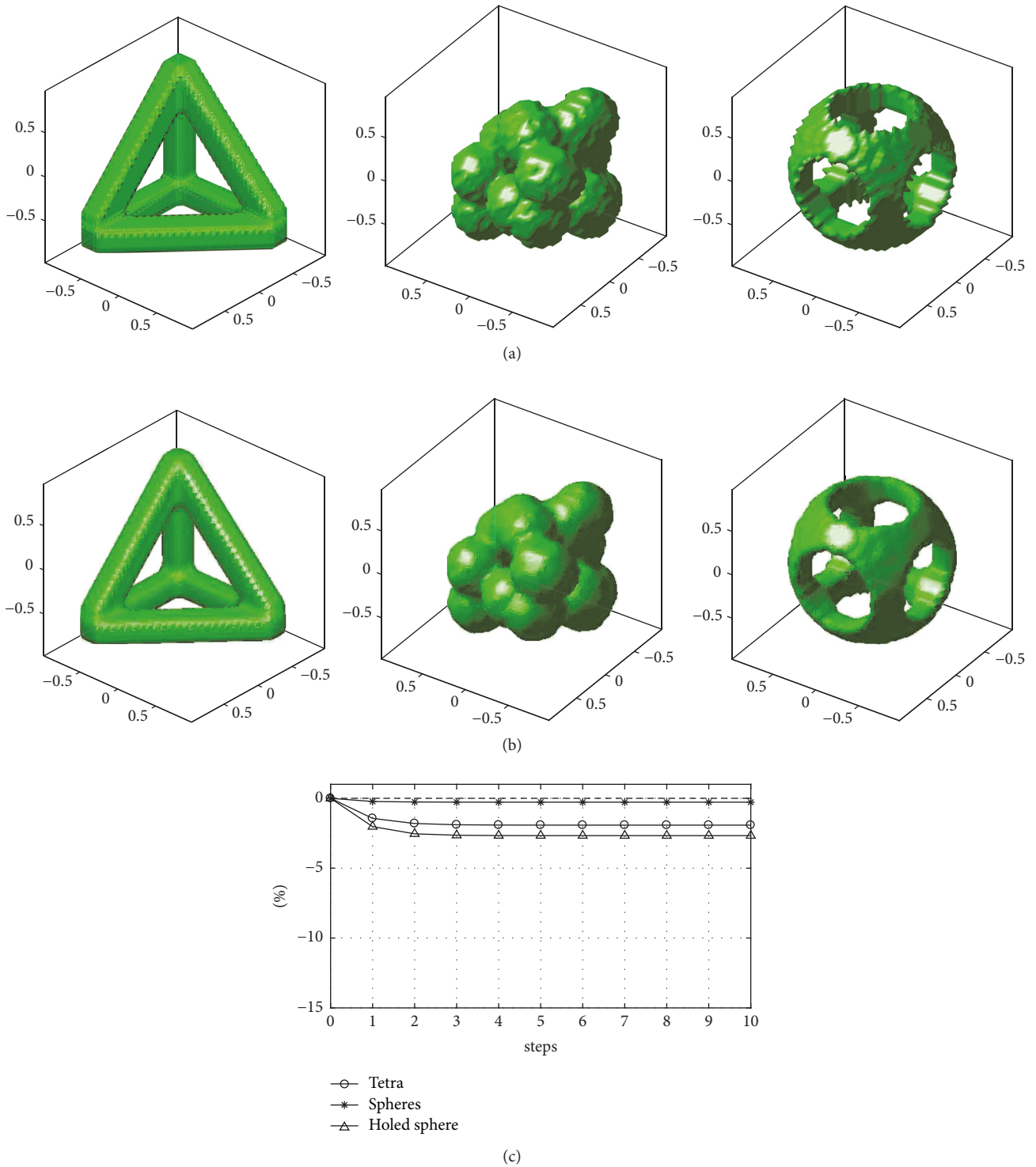


FIGURE 7: (a) Three different initial conditions. (b) Smoothed results after 10 iterations by the proposed method. (c) Percentage of relative error of volume against iteration steps.

percent noise. The other parameters are the same as the previous test. Figure 8 also shows the smooth numerical results. Even with noise, our smoothing approach has robust and effective results.

3.7.1. Complex 3D Morphology. As the final test, we take Armadillo, Stanford bunny, and happy Buddha models as the initial conditions. For all tests, the parameters $\lambda = 10^4$, $\Delta t = 0.01$, and $\varepsilon = h$ are used. The others are stated in Table 1.

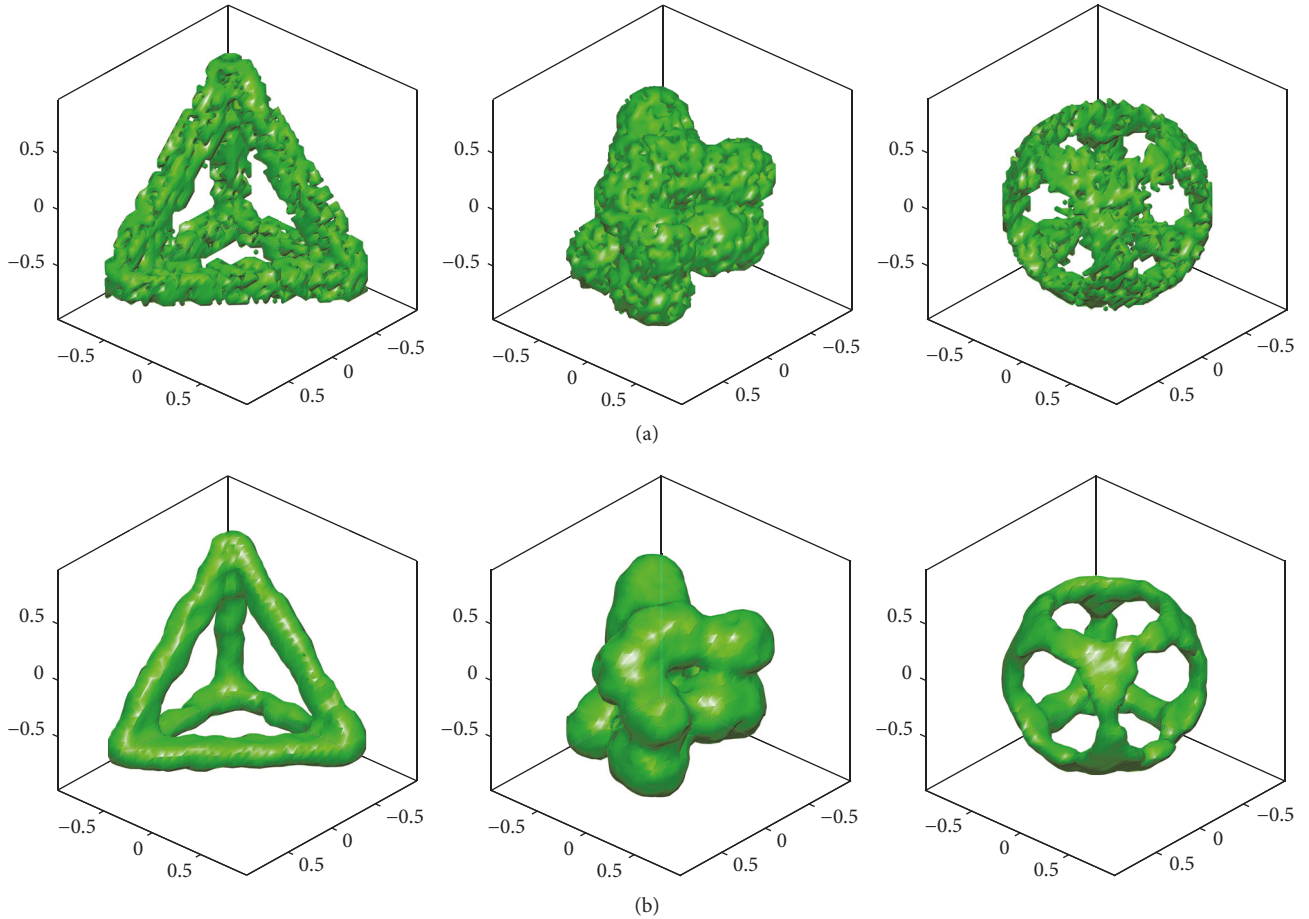


FIGURE 8: (a) Three different initial condition with 30% noise. (b) Smoothed results after 10 iterations by the proposed method.

TABLE 1: Domain size, mesh grid points, and spatial step size used in Armadillo, stanford bunny, and happy buddha models.

Case	(a) Armadillo	(b) Stanford bunny	(c) Happy buddha
Domain size	$(0, 2) \times (0, 2.4) \times (0, 2)$	$(0, 1) \times (0, 1) \times (0, 1)$	$(0, 1) \times (0, 2.2) \times (0, 1)$
Mesh grid points	$50 \times 60 \times 50$	$60 \times 60 \times 50$	$100 \times 220 \times 100$
Spatial step size	$h = 0.04$	$h = 0.0167$	$h = 0.01$

The columns in Figure 9 represent the initial condition and numerical results after 1 and 3 iterations. As we expected, the numerical results are smooth and preserve the initial feature.

To show the effectiveness of the proposed smoothing approach, we compare results in [5] with those from our proposed method. Figure 10 illustrates CPU time in seconds against the total number of vertices of a model. As shown in Figure 10, under the similar number of vertex points, our proposed method gets smooth results and a reduction of over two orders of magnitude in CPU time.

Also, in Figure 11, we represent the percent of the relative error of volume ($V_{err}(\mathbf{X})$) which is stated in (12). Through these results, we can see that the proposed method implements the smoothing procedure almost without total volume loss.

4. Conclusion

In this paper, we have introduced a novel method using the modified CH equation for smoothing piecewise linear shapes of two- and three-dimensional objects without producing significant shrinkage. The main advantages of the proposed method are very fast and easy to implement because we use a fast Fourier transform. Various computational experiments for both curve and surface smoothing problems were presented. The results demonstrate that the proposed algorithm is effective and efficient for smoothing curves and surfaces.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

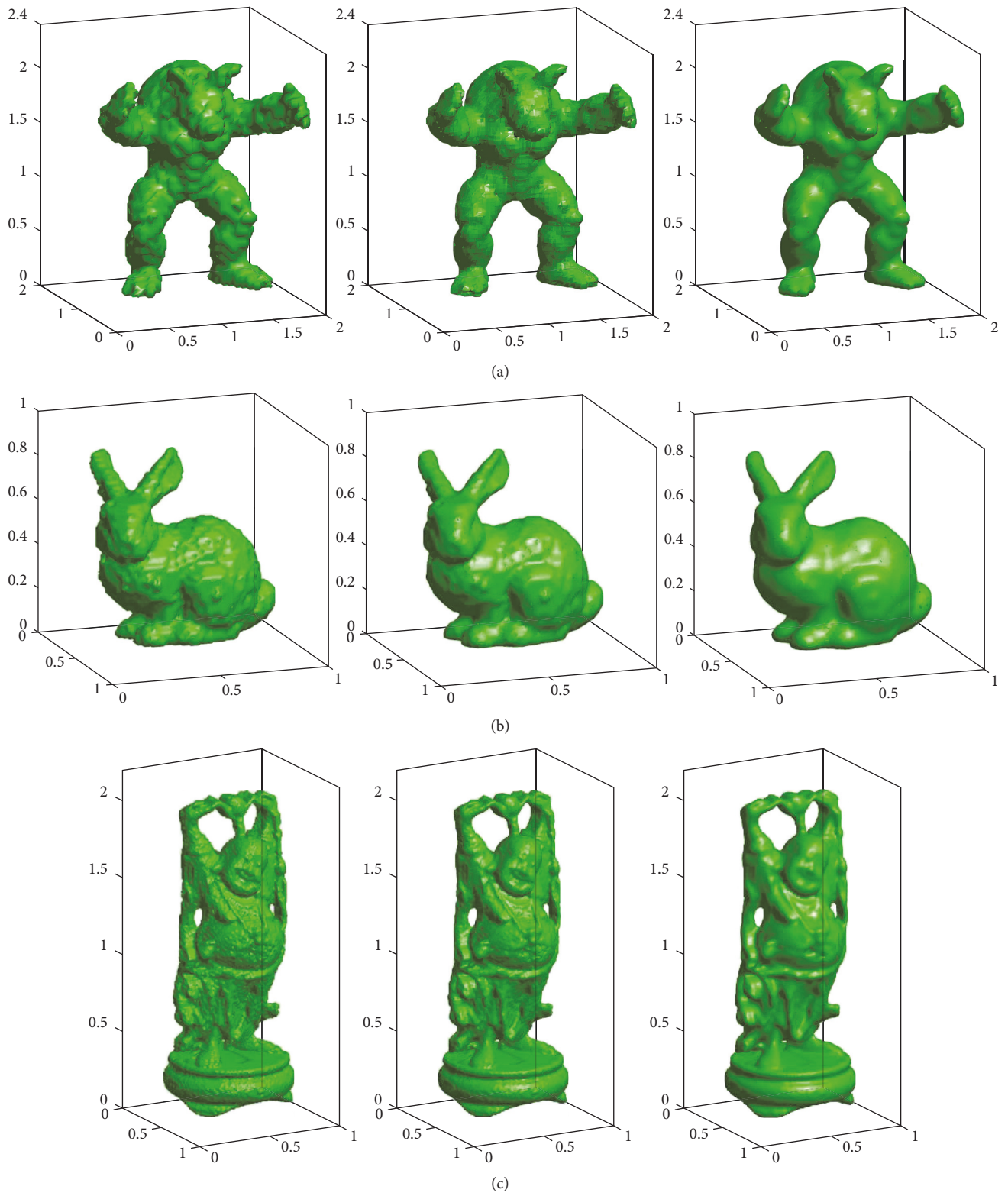


FIGURE 9: Smoothing morphologies at $T = 0$ (first column), $T = \Delta t$ (second column), and $T = 3\Delta t$ (third column). Here, (a) Armadillo, (b) Stanford bunny, and (c) happy Buddha models are used.

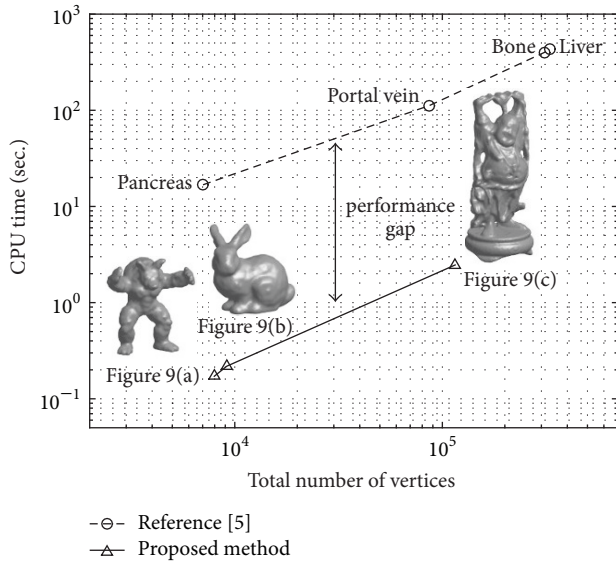


FIGURE 10: Total number of vertices versus CPU time.

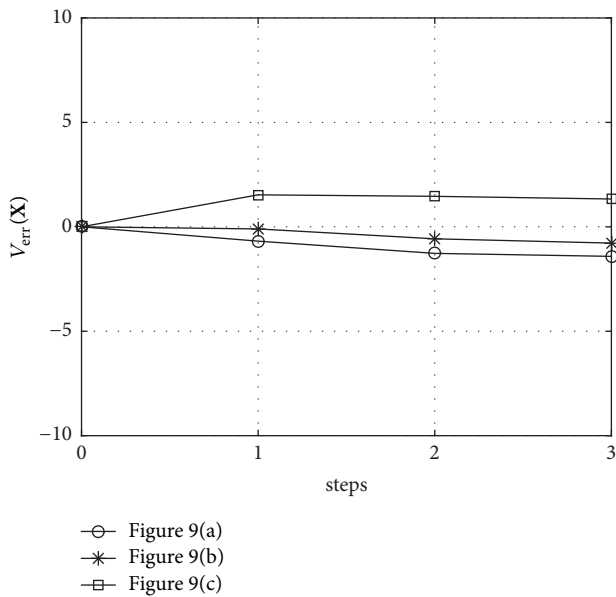


FIGURE 11: Temporal percent of relative error of volume with initial volume of Figure 9.

Acknowledgments

The first author (Y. Choi) was supported by the BK21 PLUS program. 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 Education (NRF-2016R1D1A1B03933243).

References

[1] G. Taubin, “Curve and surface smoothing without shrinkage,” in *Proceedings of the 5th International Conference on Computer Vision*, pp. 852–857, June 1995.

[2] J. Vollmer, R. Mencl, and H. Muller, “Improved Laplacian Smoothing of Noisy Surface Meshes,” *Computer Graphics Forum*, vol. 18, no. 3, pp. 131–138, 1999.

[3] C. T. M. Anflor and R. J. Marczak, “Topological optimization of anisotropic heat conducting devices using bezier-smoothed boundary representation,” *CMES: Computer Modeling in Engineering & Sciences*, vol. 78, no. 3-4, pp. 151–168, 2011.

[4] X. Liu, H. Bao, H.-Y. Shum, and Q. Peng, “A novel volume constrained smoothing method for meshes,” *Graphical Models*, vol. 64, no. 3-4, pp. 169–182, 2002.

[5] M. Wei, L. Zhu, J. Yu et al., “Morphology-preserving smoothing on polygonized isosurfaces of inhomogeneous binary volumes,” *Computer-Aided Design*, vol. 58, pp. 92–98, 2015.

[6] X. Zhang, X. Chen, Y. Liu, B. Han, T. Zhuang, and W. Zuo, “An Effective Approach of Teeth Segmentation within the 3D Cone Beam Computed Tomography Image Based on Deformable Surface Model,” *Mathematical Problems in Engineering*, vol. 2016, Article ID 9505217, 2016.

[7] H. Sima, A. Mi, Z. Wang, and Y. Zou, “Objectness Supervised Merging Algorithm for Color Image Segmentation,” *Mathematical Problems in Engineering*, vol. 2016, Article ID 3180357, 2016.

[8] J. W. Cahn and J. E. Hilliard, “Free energy of a nonuniform system. I. Interfacial free energy,” *The Journal of Chemical Physics*, vol. 28, no. 2, pp. 258–267, 1958.

[9] J. W. Cahn, “Free energy of a nonuniform system. II. Thermodynamic basis,” *The Journal of Chemical Physics*, vol. 30, no. 5, pp. 1121–1124, 1959.

[10] D. Lee, J.-Y. Huh, D. Jeong, J. Shin, A. Yun, and J. Kim, “Physical, mathematical, and numerical derivations of the Cahn-Hilliard equation,” *Computational Materials Science*, vol. 81, pp. 216–225, 2014.

[11] J. Kim, S. Lee, Y. Choi, S.-M. Lee, and D. Jeong, “Basic principles and practical applications of the Cahn-Hilliard equation,” *Mathematical Problems in Engineering*, Article ID 9532608, 11 pages, 2016.

[12] A. L. Bertozzi, S. Esedoglu, and A. Gillette, “Inpainting of binary images using the Cahn-Hilliard equation,” *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 285–291, 2007.

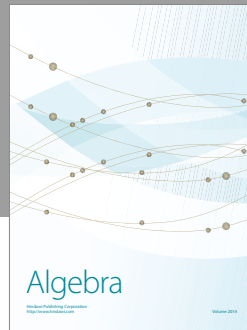
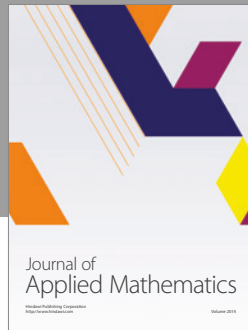
[13] C.-M. Chang and H.-D. Yeh, “Variability quantification of excess pressure head in heterogeneous deformable aquifers,” *Applied Mathematical Modelling: Simulation and Computation for Engineering and Environmental Systems*, vol. 40, no. 19-20, pp. 8580–8591, 2016.

[14] S. Zhai, Z. Weng, and X. Feng, “Fast explicit operator splitting method and time-step adaptivity for fractional non-local Allen-Cahn model,” *Applied Mathematical Modelling: Simulation and Computation for Engineering and Environmental Systems*, vol. 40, no. 2, pp. 1315–1324, 2016.

[15] D. J. Eyre, “An unconditionally stable one-step scheme for gradient systems,” *Unpublished Article*.

[16] J. Shen, T. Tang, and L. Wang, *Spectral Methods: Algorithms, Analysis and Applications*, vol. 41 of *Springer Series in Computational Mathematics*, Springer, New York, NY, USA, 2011.

[17] Y. Li, A. Yun, D. Lee, J. Shin, D. Jeong, and J. Kim, “Three-dimensional volume-conserving immersed boundary model for two-phase fluid flows,” *Computer Methods Applied Mechanics and Engineering*, vol. 257, pp. 36–46, 2013.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

