Computer-Aided Design 120 (2020) 102804

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Fast and Accurate Smoothing Method Using A Modified Allen–Cahn Equation ☆

Jian Wang [a], Yibao Li [b], Yongho Choi [c], Chaeyoung Lee [a], Junseok Kim [a,*]

[a] Department of Mathematics, Korea University, Seoul 02841, Republic of Korea
[b] School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China
[c] Department of Mathematics and Big Data, Daegu University, Gyeongsan-si, Gyeongsangbuk-do 38453, Republic of Korea

ARTICLE INFO

Article history:
Received 15 April 2019
Received in revised form 4 November 2019
Accepted 21 November 2019

Keywords:
Allen–Cahn equation
Image smoothing
Explicit numerical method

ABSTRACT

This paper presents a fast and accurate method using the Allen–Cahn (AC) equation with a fidelity term for curves smoothing of 2*D* shapes and volume smoothing of 3*D* shapes. The modified AC equation has a good smoothing dynamics and it is coupled with a fidelity term. The fidelity term forces the solution of the equation to be a close approximation to the original data. We use a hybrid explicit finite difference method to solve the equation. Therefore, we do not have any restriction on the shape of the computational domains. Several numerical tests for both the curve and surface smoothing problems are performed to demonstrate the robustness and efficiency of the proposed method. In particular, the proposed algorithm is useful for the 3D printing applications.

1. Introduction

In this study, we consider a fast and accurate smoothing method using a modified Allen–Cahn equation [1] in the two- and three-dimensional domains. Most existing smoothing methods suffer from a number of problems, and the most well-known problem is the volume shrinkage or shape distortion [2] as shown in Fig. 1(b).

Wei et al. [4] proposed an effective mesh smoothing algorithm which focuses on polygonized isosurfaces of inhomogeneous binary volumes. Guo et al. [5] introduced low-pass filtering based on non-uniform spectral synthesis to smooth 3D medical models while preserving features. A propagated mesh normal filtering model, introduced in [6], preserved prominent features when removing noises. For more details of the normal filtering model, please refer to Yadav's doctoral dissertation [7,8]. Yadav et al. [9] proposed a model to smooth the surfaces by using a normal voting tensor based on element. By optimizing the fuzzy vector median filters, the authors in [10] smoothed various surfaces represented by triangular meshes. Ohtake et al. [11] combined the Laplacian flow and a mesh evolution by a function of the mean curvature for smoothing polygonal surfaces, feature preserving, and crease enhancement.

Object feature can be preserved by a normal-diffusion process while dealing with mesh denoising [12]. In [13], the authors introduced feature preserving mesh denoising based on graph spectral processing, in the feature-aware fine step, they iteratively smooth face normals and vertices, while preserving geometric features. Liu et al. [14] proposed a volume-constrained smoothing algorithm for triangular meshes, which preserves exactly the mesh volume during the smoothing process. Cai et al. [15] proposed a structure preserving smoothing method via relativity-of-Gaussian. Zheng et al. [16] proposed a new point position updating formulation and adopt a multi-normal strategy to overcome sharp edge shrinkage. Chen et al. [17] studied shape preserving mesh texture smoothing by using joint low-rank matrix recovery. Wang et al. [18] collected similar patches and formulated them as a low-rank matrix recovery problem to estimate surface normals for dealing with different levels of noise and they suggested data-driven geometry-recovering mesh denoising. Zhao et al. [19] developed a novel L_0 sparse regularization method to robustly and reliably eliminate noises while preserving features with theoretic guarantee. The authors presented a simple and fast mesh denoising method, which can remove noise effectively while preserving mesh features such as sharp edges and corners in [20]. Lu et al. [21] presented a novel scheme for robust feature-preserving mesh denoising, through the experiments, the results show that the method can remove noise while maximally preserving geometric features. In [22], while preserving features, the authors generated an anisotropic point set denoising operation and remove noise and outliers from the input point set. Wei

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.cad.2019.102804.

* Corresponding author.
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).

https://doi.org/10.1016/j.cad.2019.102804
0010-4485/

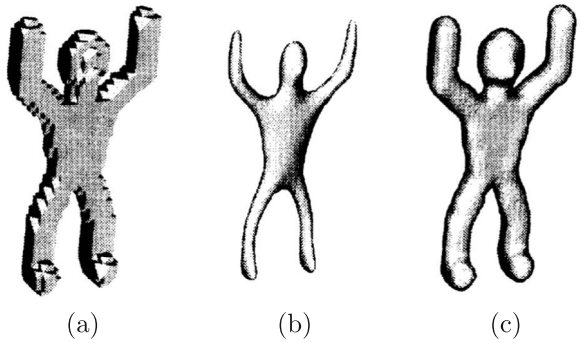

Fig. 1. (a) Original image (b) Result with Gaussian smoothing. (c) Result with a smoothing method. Reprinted from Taubin [3], with permission from Proceedings of IEEE International Conference on Computer Vision.

et al. [23] proposed tensor voting guided mesh denoising which demonstrates a outstanding performance on feature preservation and artifact suppression.

Although various smoothing techniques have been introduced, however, most of these methods focus on processing triangulated mesh smoothing, few studies have been done in the area of volume smoothing using a phase-field method. Choi et al. [24] proposed a phase-field method using the modified Cahn–Hilliard equation for smoothing piecewise linear shapes of two- and three-dimensional objects. In this paper, we present a new method using the modified AC equation for smoothing piecewise linear shapes of two- and three-dimensional objects. We compare our method with several state-of-the-arts in terms of visual and quantitative evaluations.

In the proposed method, let ϕ be an order parameter that takes the value $+1$ in the solid volume and the value -1 outside the volume. We define the zero-isosurface of ϕ as the surface of a volume. We consider the following energy functional for volume smoothing:

$$\mathcal{E}(\phi) = \int_{\Omega} \left[\frac{F(\phi)}{\epsilon^2} + \frac{|\nabla\phi|^2}{2} + \frac{\lambda}{2}(f(\mathbf{x}) - \phi)^2 \right] d\mathbf{x}, \tag{1}$$

where the second term $0.5|\nabla\phi|^2$ makes ϕ to be smooth and the first term $F(\phi) = 0.25(\phi^2 - 1)^2/\epsilon^2$ enforces ϕ to be sharp across the volume transition. Here, ϵ is a positive parameter related to the interfacial transition thickness. If only these two terms are considered, then the temporal evolution of the gradient flow of the functional (1) becomes motion by mean curvature [1], which results in volume shrinkage. The third term $0.5\lambda(f(\mathbf{x}) - \phi)^2$ is a fidelity term, which enforces ϕ to be the original scaled volume $f(\mathbf{x})$. Here, λ is the fidelity strength coefficient. The scaled given image is defined as

$$f(\mathbf{x}) = \frac{2\psi(\mathbf{x}) - f_{\max} - f_{\min}}{f_{\max} - f_{\min}}, \tag{2}$$

where $f_{\max}$ and $f_{\min}$ are the maximum and the minimum values of the given image $\psi(\mathbf{x})$, respectively. By applying gradient descent with respect to L^2 inner product for the energy $\mathcal{E}(\phi)$, we have the following evolution equation:

$$\frac{\partial\phi(\mathbf{x}, t)}{\partial t} = -\frac{F'(\phi(\mathbf{x}, t))}{\epsilon^2} + \Delta\phi(\mathbf{x}, t) + \lambda(f(\mathbf{x}) - \phi(\mathbf{x}, t)). \tag{3}$$

If $\lambda = 0$ in Eq. (3), then the proposed equation becomes the classical AC equation [1], which has been used as a mathematical model to investigate antiphase domain coarsening in a binary alloy.

The outline of this paper is the following. In Section 2, numerical solution algorithms for the curve and volume smoothing are presented. To demonstrate the efficiency and robustness of our proposed method, computational examples are presented in Section 3. In Section 4, we apply the method to 3D printing. Finally, conclusions are drawn in Section 5.

2. Numerical solution algorithm

In this paper, we use the hybrid explicit finite difference method to solve the proposed model. Let the computational domain Ω be $[x_L,\ x_R]\times[y_L, y_R]$ in the two-dimensional space, N_x and N_y be positive even integers, $h = (x_R - x_L)/N_x = (y_R - y_L)/N_y$ be the uniform grid size, Δt be the time step size, and $\Omega_h = \{(x_i, y_j) | x_i = x_L + h(i - 0.5),\ y_j = y_L + h(j - 0.5),\ 1 \le i \le N_x,\ 1 \le j \le N_y\}$ be the set of cell-centers. Let ϕ_{ij}^n be approximations of $\phi(x_i, y_j, n\Delta t)$. We formally split the original problem (3) into the following three equations:

$$\frac{\partial\phi}{\partial t} = \Delta\phi, \tag{4}$$

$$\frac{\partial\phi}{\partial t} = -\frac{F'(\phi)}{\epsilon^2}, \tag{5}$$

$$\frac{\partial\phi}{\partial t} = \lambda(f(\mathbf{x}) - \phi). \tag{6}$$

As the first step, we solve Eq. (4) by applying the fully explicit method with an initial condition ϕ^n, that is,

$$\frac{\phi_{ij}^{n+\frac{1}{3}} - \phi_{ij}^n}{\Delta t} = \Delta_d\phi_{ij}^n, \ \text{for } (x_i, y_j) \in \Omega_h, \tag{7}$$

where Δ_d is the discrete Laplacian operator, i.e.,

$$\Delta_d\phi_{ij} = (\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} - 4\phi_{ij})/h^2,$$

for $i = 1, \ldots, N_x$ and $j = 1, \ldots, N_y$. At the computational domain boundary, we use the zero Neumann boundary condition as follows:

$$\phi_{0,j} = \phi_{1,j}, \quad \phi_{N_x+1,j} = \phi_{N_x,j}, \quad \phi_{i,0} = \phi_{i,1}, \quad \phi_{i,N_y+1} = \phi_{i,N_y}.$$

Next, Eq. (5) is solved by the method of separation of variables [25] as

$$\phi_{ij}^{n+\frac{2}{3}} = \frac{\phi_{ij}^{n+\frac{1}{3}}}{\sqrt{e^{\frac{-2\Delta t}{\epsilon^2}} + (\phi_{ij}^{n+\frac{1}{3}})^2(1 - e^{\frac{-2\Delta t}{\epsilon^2}})}}. \tag{8}$$

Finally, we solve Eq. (6) by using the fully implicit method:

$$\frac{\phi_{ij}^{n+1} - \phi_{ij}^{n+\frac{2}{3}}}{\Delta t} = \lambda(f_{ij} - \phi_{ij}^{n+1}), \tag{9}$$

which can be explicitly rewritten as

$$\phi_{ij}^{n+1} = \frac{\phi_{ij}^{n+\frac{2}{3}} + \lambda\Delta t f_{ij}}{1 + \lambda\Delta t}. \tag{10}$$

Similarly to the two-dimensional space, we define the computational domain. Let the computational domain Ω be $[x_L, x_R]\times$

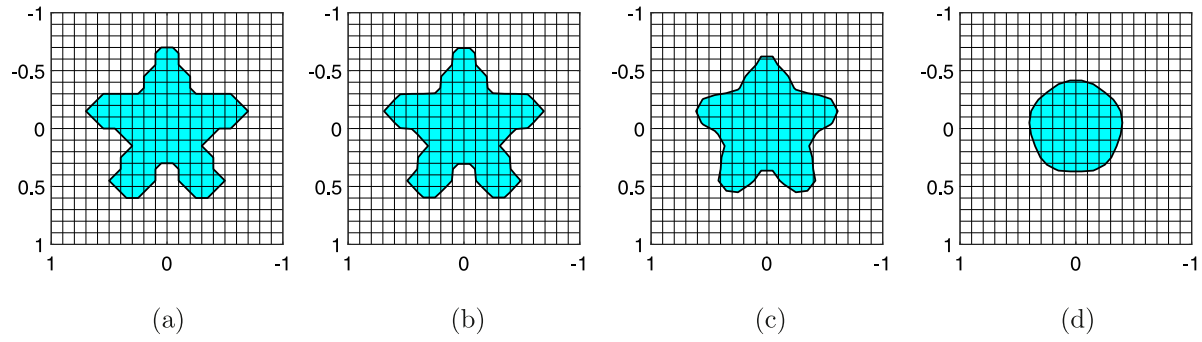

Fig. 2. Effect of λ on the smoothing dynamics. (a) Initial condition and numerical results at $t = 50\Delta t$ with (b) $\lambda = 1000$, (c) $\lambda = 100$, and (d) $\lambda = 10$.

$[y_L, y_R] \times [z_L, z_R]$ in the three-dimensional space, N_x, N_y, and N_z be positive even integers, $h = (x_R - x_L)/N_x = (y_R - y_L)/N_y = (z_R - z_L)/N_z$ be the uniform grid size, and $\Omega_h = \{(x_i, y_j, z_k) |\ x_i = x_L + h(i - 0.5),\ y_j = y_L + h(j - 0.5),\ z_k = z_L + h(k - 0.5),\ 1 \le i \le N_x,\ 1 \le j \le N_y,\ 1 \le k \le N_z\}$ be the set of cell-centers. Let ϕ^n_{ijk} be approximations of $\phi(x_i, y_j, z_k, n\Delta t)$, where Δt is the time step. At first, we solve Eq. (4) by using the fully explicit method with the initial condition ϕ^n, that is,

$$\frac{\phi^{n+\frac{1}{3}}_{ijk} - \phi^n_{ijk}}{\Delta t} = \Delta_d \phi^n_{ijk}, \text{ for } (x_i, y_j, z_k) \in \Omega_h, \tag{11}$$

where Δ_d is the discrete Laplacian operator, i.e.,

$$\Delta_d \phi_{ijk} = (\phi_{i+1,jk} + \phi_{i-1,j,k} + \phi_{i,j+1,k} + \phi_{i,j-1,k} + \phi_{ij,k+1} + \phi_{ij,k-1} - 6\phi_{ijk})/h^2,$$

for $i = 1, \ldots, N_x$, $j = 1, \ldots, N_y$, and $k = 1, \ldots, N_k$. At the boundary, we implement the zero Neumann boundary condition as follows:

$$\phi_{0,j,k} = \phi_{1,j,k}, \quad \phi_{i,0,k} = \phi_{i,1,k}, \quad \phi_{i,j,0} = \phi_{i,j,1},$$
$$\phi_{N_x+1,j,k} = \phi_{N_x,j,k}, \quad \phi_{i,N_y+1,k} = \phi_{i,N_y,k}, \quad \phi_{i,j,N_z+1} = \phi_{i,j,N_z}.$$

Next, Eq. (5) is solved by the method of separation of variables as

$$\phi^{n+\frac{2}{3}}_{ijk} = \frac{\phi^{n+\frac{1}{3}}_{ijk}}{\sqrt{e^{\frac{-2\Delta t}{\epsilon^2}} + (\phi^{n+\frac{1}{3}}_{ijk})^2 (1 - e^{\frac{-2\Delta t}{\epsilon^2}})}}. \tag{12}$$

Finally, we solve Eq. (6) by using the fully implicit method:

$$\frac{\phi^{n+1}_{ijk} - \phi^{n+\frac{2}{3}}_{ijk}}{\Delta t} = \lambda(f_{ijk} - \phi^{n+1}_{ijk}), \tag{13}$$

which can be explicitly rewritten as

$$\phi^{n+1}_{ijk} = \frac{\phi^{n+\frac{2}{3}}_{ijk} + \lambda \Delta t f_{ijk}}{1 + \lambda \Delta t}. \tag{14}$$

3. Numerical tests

3.1. Effect of parameters

In this section, we investigate the effect of parameters on the smoothing dynamics. We first consider the effect of the fidelity parameter λ on the minimum shrinkage. On the computational domain $\Omega = (-1, 1) \times (-1, 1)$ with a mesh grid $N \times N$, the initial configuration is given as

$$f(x, y) = \begin{cases} 1 & \text{if } \sqrt{x^2 + y^2} + 0.2\cos(5\theta) \le 0.5, \\ -1 & \text{otherwise,} \end{cases} \tag{15}$$

where $\theta = \tan^{-1}(y/x)$ if $x > 0$; $\theta = \pi + \tan^{-1}(y/x)$ otherwise. We define the interfacial length parameter ϵ_m as $\epsilon_m = mh/[\sqrt{2}\tanh^{-1}(0.9)]$. We use the following parameters: $N = 20$, $h = 2/N$, $\epsilon = \epsilon_2$, $\alpha = 1/7$, and $\Delta t = \alpha h^2$. With the initial configuration as shown in Fig. 2(a), we obtain the various numerical results (see Fig. 2(b)–(d)) at time $t = 50\Delta t$ with respect to different λ. In this test, we use three different $\lambda = 10,\ 100$, and 1000. When λ is large, $\phi(x, y, t)$ does not evolve and is close to the initial mosaic profile. However, if λ is too small, then the AC dynamics dominates and the profile becomes a circular shape which has the minimum interfacial length.

Now, we conduct a numerical experiment on the effect of the fidelity term. This term ensures that the solution ϕ approximates a given original data $f(\mathbf{x})$. For the test, we define a given data $f(\mathbf{x})$ and the phase-field ϕ on the computational domain $\Omega = (-1, 1) \times (-1, 1)$ as follows:

$$f(x, y) = \begin{cases} 1 & \text{if } \sqrt{(x + 0.3)^2 + y^2} \le 0.4, \\ -1 & \text{otherwise,} \end{cases}$$

$$\phi(x, y, 0) = \begin{cases} 1 & \text{if } |x - 0.3| \le 0.4 \text{ and } |y| \le 0.4, \\ -1 & \text{otherwise.} \end{cases}$$

With the initial conditions as shown in Fig. 3(a), we solve Eq. (6) using the fully implicit method. We use the following parameters: $N = 200$, $h = 2/N$, $\epsilon = \epsilon_2$, $\lambda = 100$, $\alpha = 1/7$, and $\Delta t = \alpha h^2$. Fig. 3(b) illustrates the zero-level contour of phases, $f(\mathbf{x})$ and ϕ, at $t = 0,\ 250\Delta t,\ 500\Delta t$, and $750\Delta t$ and Fig. 3(c) shows the mesh plots of the phases. We can find that the fidelity term keep the solution of (6) close to the given data.

Next, we investigate the effect of ϵ. In our proposed model, the parameter ϵ determines the thickness of the interface of two phases, that is, interfacial region of transition profile. Therefore, it is one of 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 Eq. (15) as the initial condition as shown in Fig. 4(a). The other parameters are used as $N = 20$, $h = 2/N$, $\alpha = 1/7$, $\Delta t = \alpha h^2$, and $\lambda = 100$. Fig. 4(b)–(d) represent the numerical configurations at $t = 50\Delta t$ with $\epsilon = \epsilon_8$, $\epsilon = \epsilon_2$, and $\epsilon = \epsilon_{0.4}$, respectively. As shown in Fig. 4(d), if ϵ is too small, then the phase-field has abrupt transition and the profile becomes mosaic. On the other hand, according to Eq. (8), as ϵ increases, $\phi^{n+2/3}_{ijk}$ converges to 1, i.e., the large value of ϵ cannot make a big difference in our numerical solution. As shown in Fig. 4(b) and (c), it is hard to find significant difference between them.

3.2. 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 $N = 64$, $h = 2/N$,

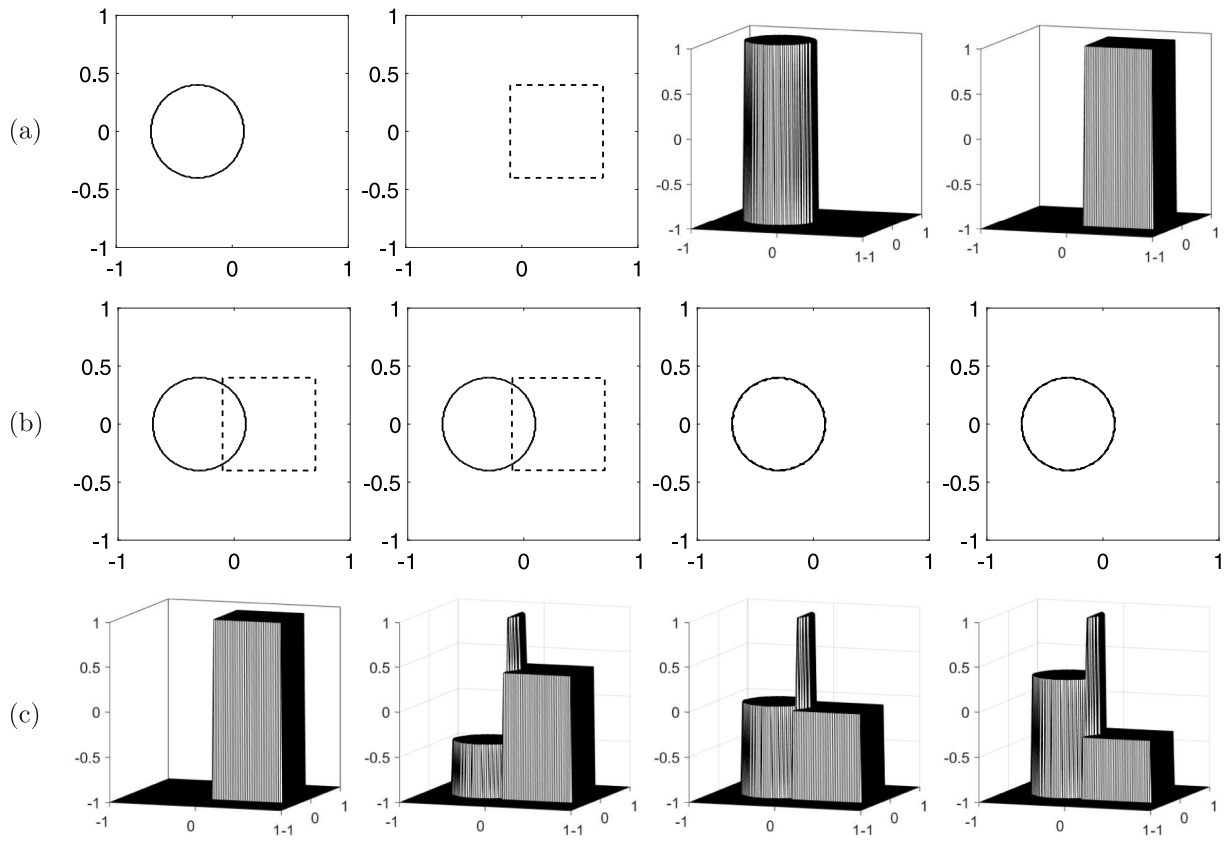

Fig. 3. Effect of the fidelity term. (a) shows the initial conditions of the given data $f(\mathbf{x})$ and the phase-field ϕ. Here, the first two columns represent the zero-level contour of phases and the third and fourth columns are the mesh plots of the phases. (b) and (c) show the evolution of ϕ at $t = 0,\ 250\Delta t,\ 500\Delta t$, and $750\Delta t$.

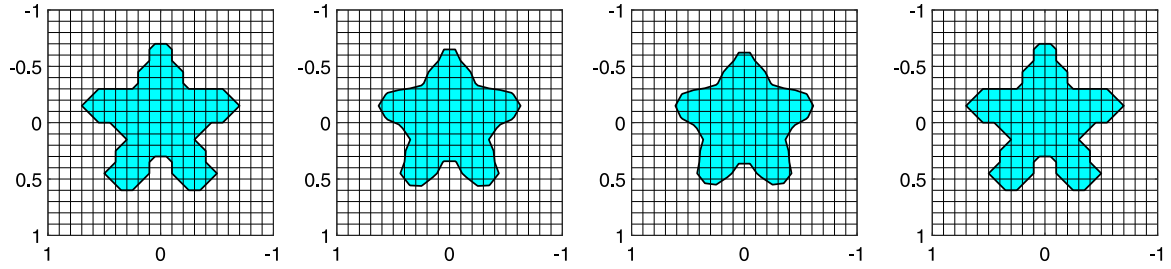

Fig. 4. Effect of ϵ on the smoothing dynamics. (a) Initial condition and numerical results at $t = 50\Delta t$ with (b) ϵ_8, (c) ϵ_2, and (d) $\epsilon_{0.4}$.

$\epsilon = \epsilon_2$, $\alpha = 1/7$, $\Delta t = \alpha h^2$ and $\lambda = 100$. From left to right, Fig. 5(a) represents the initial configurations of numbers '2', '3', and '5', respectively. Fig. 5(b) shows their corresponding smooth results after 5 iterations. Observing these results, we can see that our proposed algorithm works well to remove staircases while preserving the initial volumes.

Fig. 6 shows the temporal evolution of bunny (first column), armadillo (second row), and horse (third row) at $t = 0$ and $t = 5\Delta t$. Here we take the parameters as same as the above test.

3.3. 3D various shapes and volume preservation

In this section, we will present several numerical results on various synthetic and real images. Let the initial data of the sphere and the pyramid be

$$f(x,y,z) = \begin{cases} 1 & \text{if } \sqrt{x^2+y^2+z^2} \le 0.8, \\ -1 & \text{otherwise,} \end{cases}$$

and for some $m = 1, 2, \ldots, 27$,

$$f(x,y,z) = \begin{cases} 1 & \text{if } x,\ y \in [-1+h(3m+21.5),\ 1-h(187.5-3m)], \\ & \quad z \in [-1+h(6m+13.5),\ 1-h(6m+19.5)], \\ -1 & \text{otherwise,} \end{cases}$$

on computational domain $\Omega = (-1,1)\times(-1,1)\times(-1,1)$ with a mesh grid $200 \times 200 \times 200$. The initial data of the ellipsoid are taken as

$$f(x,y,z) = \begin{cases} 1 & \text{if } \sqrt{\frac{x^2}{4}+y^2+z^2} \le 0.8, \\ -1 & \text{otherwise,} \end{cases}$$

on computational domain $\Omega = (-2,2)\times(-1,1)\times(-1,1)$ with a mesh grid $400 \times 200 \times 200$. The other parameters $h = 0.01$, $\alpha = 1/7$, $\lambda = 100$, $\Delta t = \alpha h^2$, and $\epsilon = \epsilon_2$ are chosen. Fig. 7 shows the temporal evolution of sphere, ellipsoid, and pyramid at $t = 0$

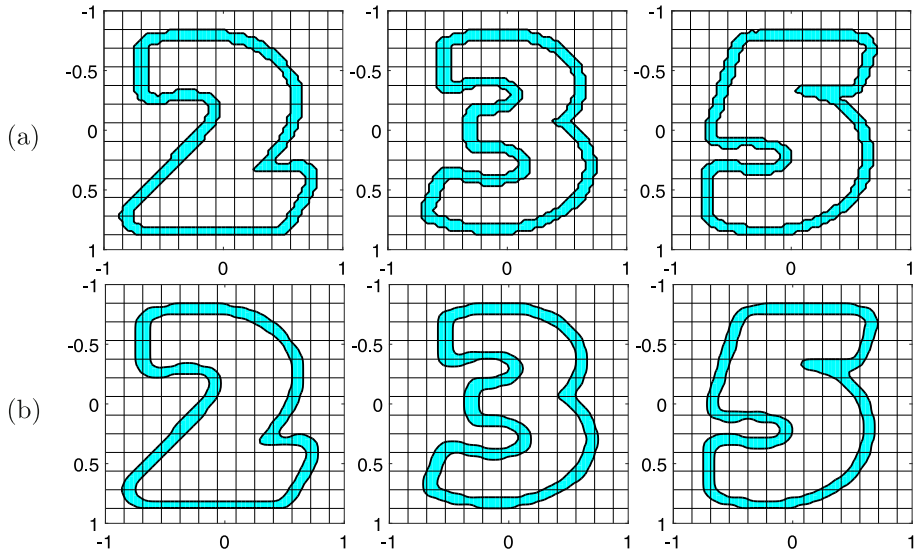

Fig. 5. (a) Three initial configurations. (b) Corresponding smooth results after 5 iterations.

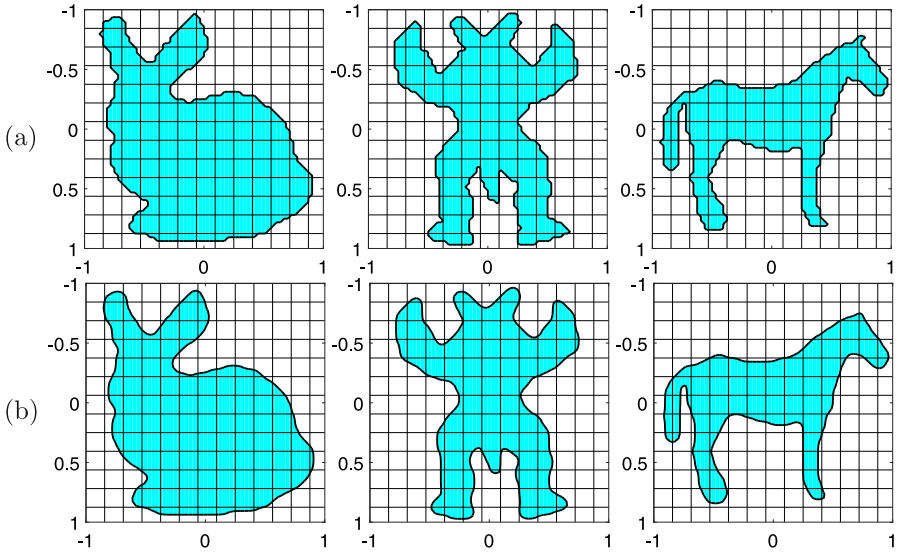

Fig. 6. (a) Three initial configurations. (b) Corresponding smooth results after 5 iterations.

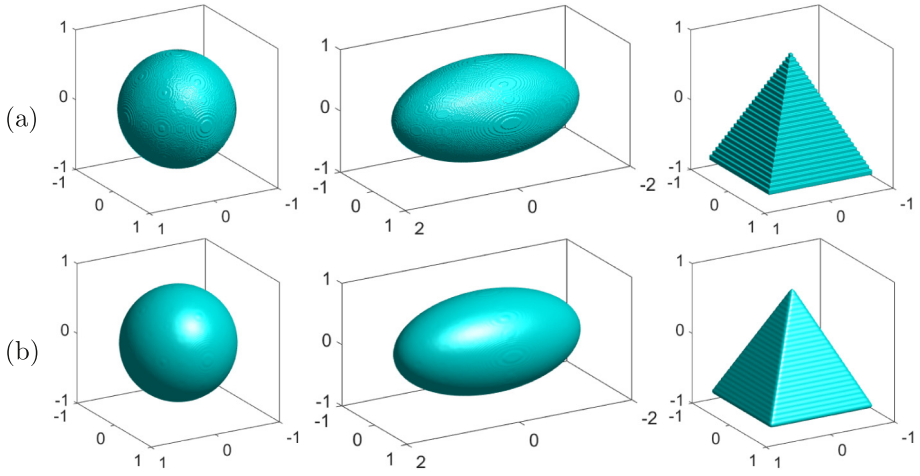

Fig. 7. Smoothing morphologies of sphere, ellipsoid, and pyramid models at (a) $t = 0$ and (b) $t = 20\Delta t$.

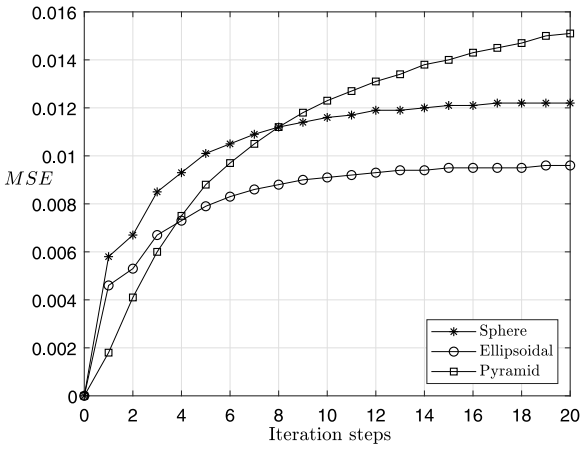

Fig. 8. Mean square error against iteration steps.

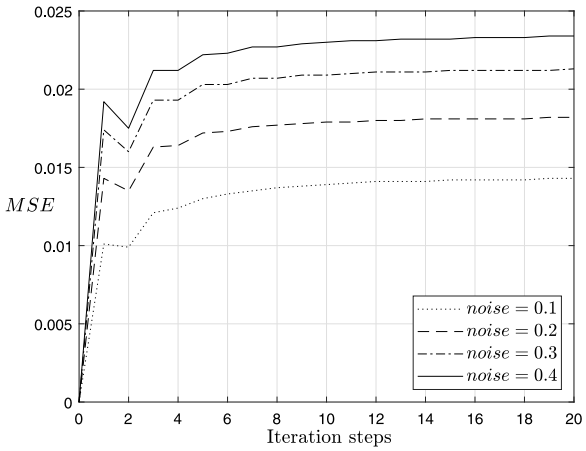

Fig. 10. Mean square error for different noise values.

and $20\Delta t$. We can see that our method works well with smooth volumes.

Laplacian smoothing is a well-known algorithm to smooth a polygonal mesh. This method can cause volume shrinkage or shape distortion [4]. However, our proposed method is a volume-preserving approach without severe volume changes. In this section, we demonstrate that our smoothing approach preserves the initial volume. We use the mean square error (*MSE*) to estimate the shrinkage error.

$$MSE(g,f) = \frac{\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\sum_{k=1}^{N_z}(g(i,j,k)-f(i,j,k))^2}{N_xN_yN_z}. \tag{16}$$

MSE is first calculated as an evaluation metric for image segmentation [26,27]. The range of *MSE* is between 0 and 4. While the smoothing process reaches a stable state, the lower value shows a better performance.

We use the above parameters to show the effectiveness of our method. In Fig. 8, we compute the shrinkage error against iteration steps. They are 1.76, 1.74, and 1.84, responding to sphere, ellipsoid, and pyramid, respectively.

Next, to show the robustness of the proposed method, we perform surface smoothing problem with noises. We assess the capability of our method on the sphere model, which is corrupted by synthetic noise. The initial phase value of sphere and parameters are taken as the same as the sphere model in Fig. 7. We perform the simulation test in the three-dimensional space $\Omega = (-1,1)^3$. We use different levels of random noise on the surface of sphere model as shown in Fig. 9(a)–(d). We include the noise in the phase-field model as follows:

$$f(x,y,z) = 1, \quad \text{if } \text{rand}(x,y,z) \le N_{val},$$
$$0.8 - 2h \le \sqrt{x^2+y^2+z^2} \le 0.8 + 2h,$$

where $\text{rand}(x,y,z)$ is a random number between 0 and 1, and N_{val} is the noise value. Here, we take noise values 10%, 20%, 30%, and 40% to the surface of the sphere. Shrinkage effect analysis during the smoothing process in the proposed method is calculated using Eq. (16). Fig. 9(e)–(h) show the effective results of the corresponding initial noised conditions. Fig. 10 shows the temporal evolution of *MSE* with different percentages of the noise at $20\Delta t$.

3.3.1. Irregular 3D morphology

In this section, we consider various irregular morphologies on the three-dimensional space. We consider three-dimensional

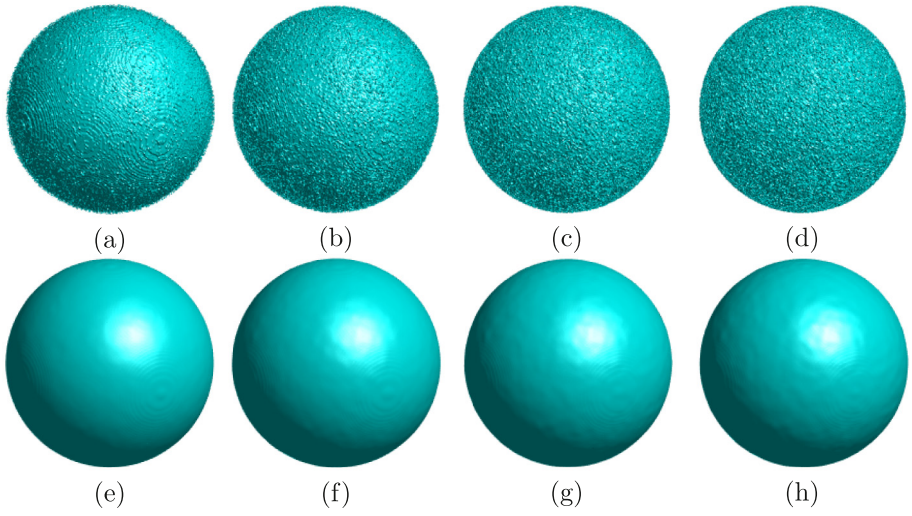

Fig. 9. Shrinkage analysis during the smoothing process for different noise values. (a), (b), (c), and (d) are the initial conditions with 10%, 20%, 30%, and 40%, noises, respectively. (e), (f), (g), and (h) are the corresponding smoothed results.

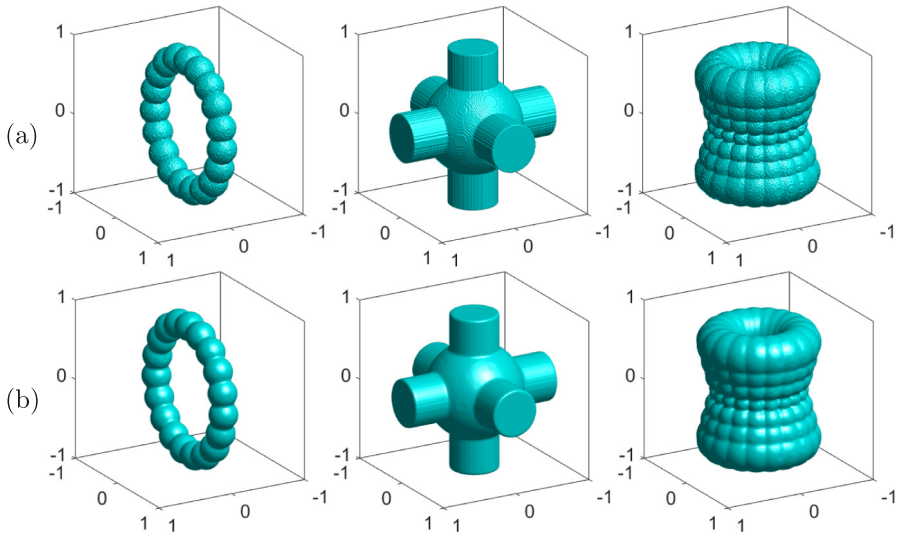

Fig. 11. Smoothing morphologies at (a) $t = 0$ and (b) $t = 10\Delta t$.

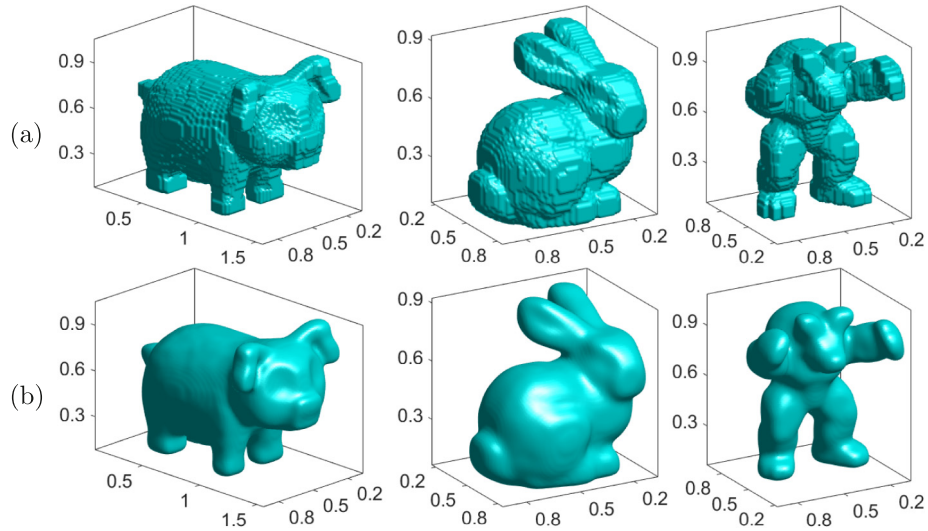

Fig. 12. Smoothing morphologies at (a) $t = 0$ and (b) $t = 20\Delta t$. Here, Piggy, Stanford bunny, and Armadillo models are used.

morphologies shown in the first row of Fig. 11 on $\Omega = (-1, 1) \times (-1, 1) \times (-1, 1)$. On a $200 \times 200 \times 200$ mesh grid, we perform the numerical test with $h = 2/N$, $\alpha = 1/7$, $\lambda = 100$, $\Delta t = \alpha h^2$, and $\epsilon = \epsilon_2$. We obtain smooth results after 10 iteration steps using our proposed method as shown in the second row in Fig. 11.

3.3.2. Complex 3D morphology

In this section, we consider various complex 3*D* morphologies on the three-dimensional space. We take Piggy, Stanford bunny, and Armadillo models as the initial conditions. We take the parameters $\lambda = 100$, $\alpha = 1/7$, $\Delta t = \alpha h^2$, $\Omega = (0, 2)^3$, $N = 200$, $h = 2/N$, and $\epsilon = \epsilon_2$.

The rows in Fig. 12 represent the initial condition and the numerical results after 20 iterations. As we expected, the numerical results are smooth and preserve the initial volumes.

Furthermore, in Fig. 13, we represent the shrinkage error which is stated in Eq. (16). The MSEs of volume are 0.78, 0.45, and 0.52, responding to Piggy, Stanford bunny, and Armadillo models. Through these results, we can see that the proposed method generate smooth results without significant volume loss.

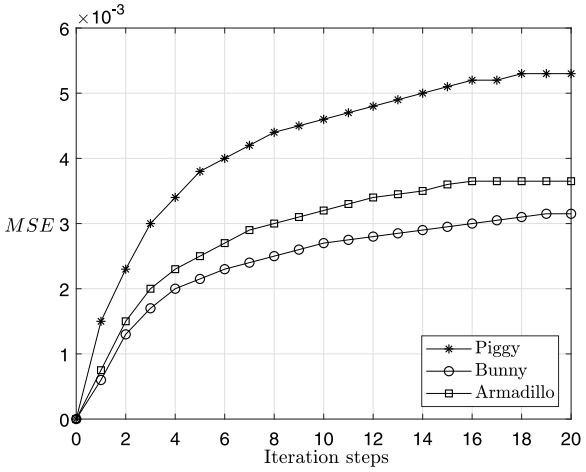

Fig. 13. Mean square error of Fig. 12.

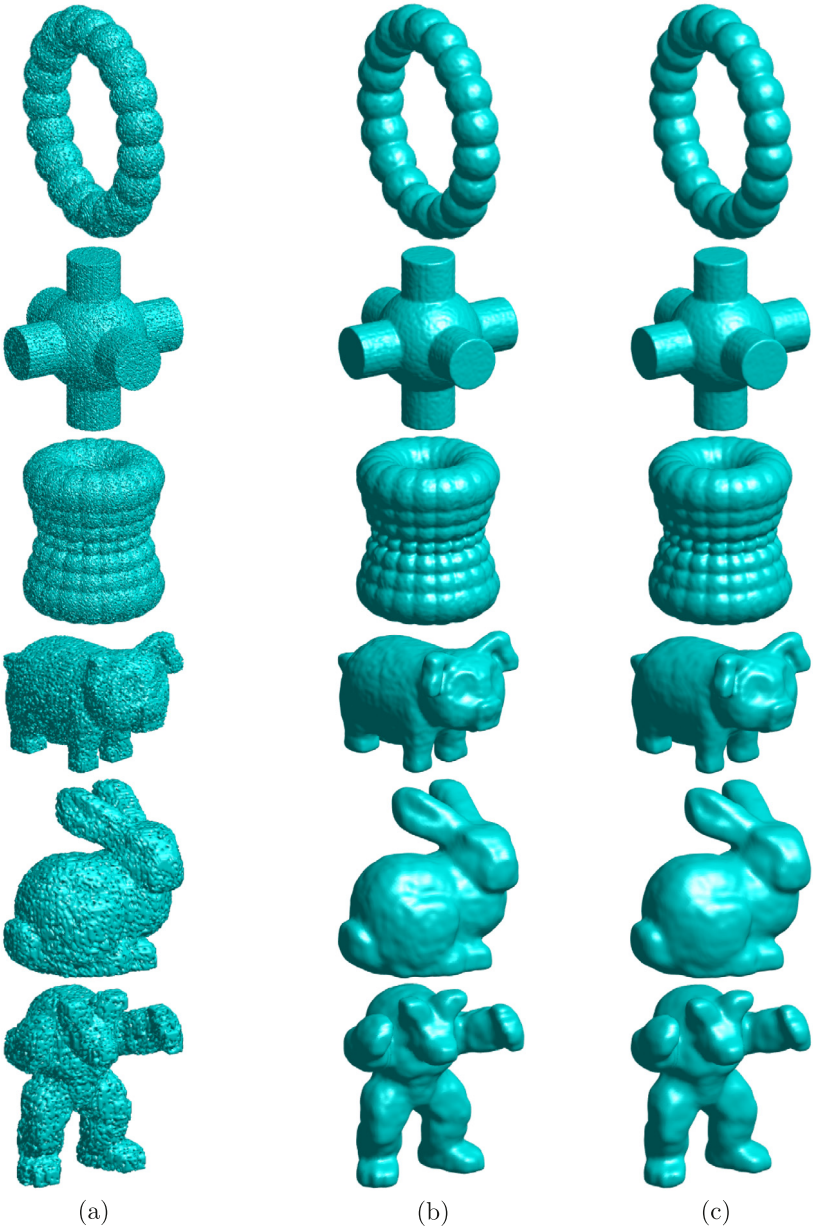

Fig. 14. Surface smoothing for six models of: (a) initial condition with 20% noise; (b) the results of the modified Cahn–Hilliard equation, and (c) the results of the proposed method.

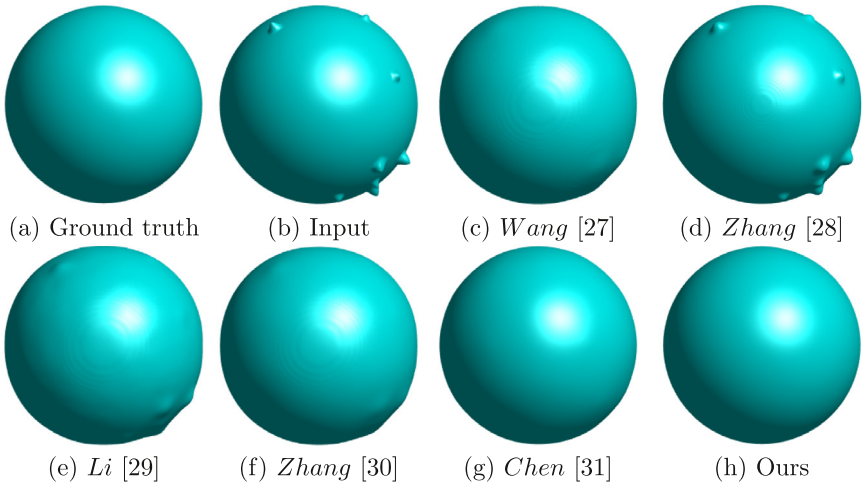

Fig. 15. Comparison of smoothing on the sphere with the state-of-the-arts.

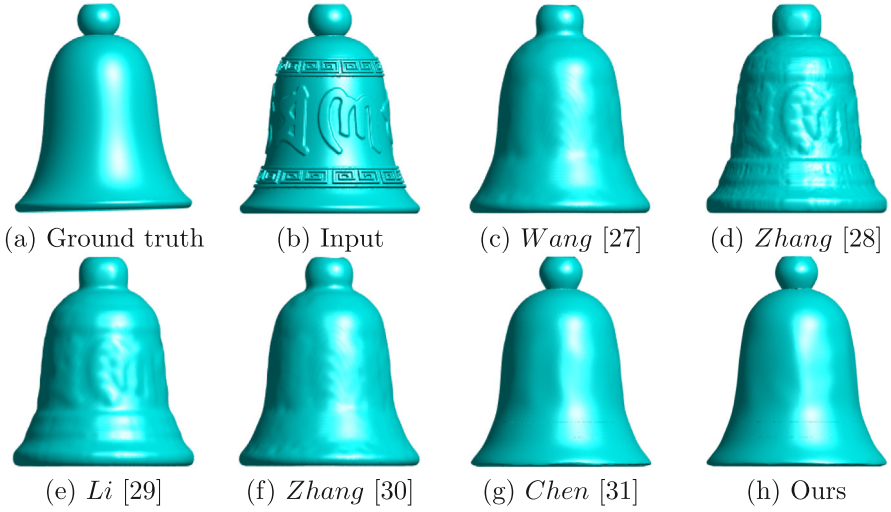

Fig. 16. Comparison of smoothing on the bell with the state-of-the-arts.

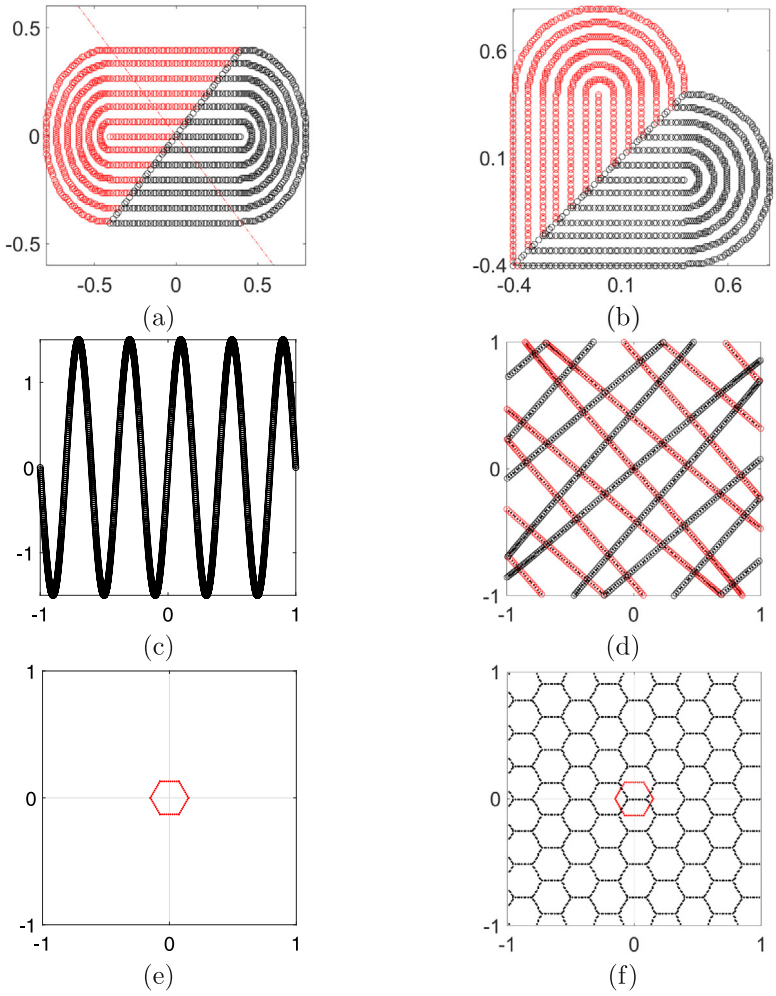

Fig. 17. Initial point data of concentric-heart, zigzag, and honeycomb. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.4. Comparison with the previous work

To demonstrate the performance of the proposed algorithm, we compare our method with recently developed methods [24, 28–32]. The comparison covers the qualitative and quantitative evaluations. To do a comparison with [24] which is also a phase-field based smoothing method, six models with 20% noise are taken as the initial conditions, as shown in Fig. 14(a). Both [24] and our method are tested in the computational domain $\Omega = (0, 2)^3$, and $N_x = N_y = N_z = 200$. Fig. 14(b) and (c) show the

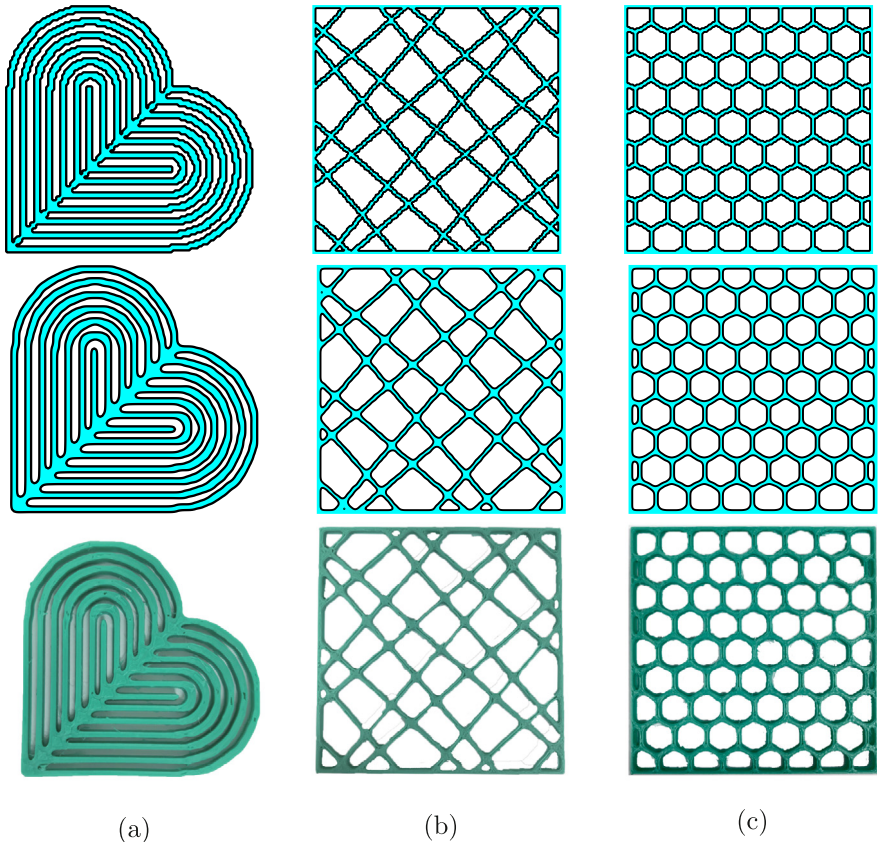

Fig. 18. (a) Concentric-heart, (b) zigzag, and (c) honeycomb.

Table 1
Quantitative evaluations. Here, parameters for [24] are selected by $h = 0.01$, $\Delta t = 0.01$, $\lambda = 10^4$, $\epsilon = h$, parameters for our proposed method are selected by $h = 0.01$, $\Delta t = h^2/7$, $\lambda = 100$, $\epsilon = \epsilon_2$. The computation complexity T_n for [24] and ours are $O(Nlog(N))$ and $O(N)$, respectively, where $N = N_x \times N_y \times N_z$.

Methods	Case	MSE	CPU time	Vertice number
Choi et al. [24]	Fig. 14a(row 1)	0.082	5.16	567 996
	Fig. 14a(row 2)	0.233	15.71	1 637 942
	Fig. 14a(row 3)	0.209	17.43	1 841 826
	Piggy	0.061	3.68	431 448
	Bunny	0.038	1.79	201 488
	Armadillo	0.032	1.66	171 652
mAC	Fig. 14a(row 1)	0.053	2.93	567 996
	Fig. 14a(row 2)	0.162	8.96	1 637 942
	Fig. 14a(row 3)	0.181	9.31	1 841 826
	Piggy	0.039	2.36	431 448
	Bunny	0.019	1.15	201 488
	Armadillo	0.016	1.02	171 652

steady phase state at $10\Delta t$ for two methods, respectively, and the proposed method is highly competitive with the method in [24].

To compare the performance of the two methods in terms of quantitative evaluations, we consider the MSE of volume as stated in Eq. (16), which represents a volume shrinkage condition of the geometric model. We list parameters of each method, computation complexity (T_n), shrinkage errors (MSE), CPU time in seconds, and vertice numbers in Table 1. The computational results show that our method generates a good visual quality and computational cost.

Table 2
Quantitative evaluations.

Task	Sphere			Bell		
	σ	d_{max}	d_{mean}	σ	d_{max}	d_{mean}
Wang et al. [28]	0.0044	**0.0355**	0.0027	0.6367	2.6661	0.7309
Zhang et al. [29]	0.0071	0.1797	0.0022	0.6238	2.6851	0.7880
Li et al. [30]	0.0064	0.1159	0.0023	0.6487	2.9672	0.7924
Zhang et al. [31]	0.0077	0.0499	0.0024	0.6880	3.2592	0.7437
Chen et al. [32]	0.0045	**0.0355**	**0.0021**	0.6147	2.6714	**0.7214**
Ours	**0.0043**	0.0358	**0.0021**	**0.6113**	**2.6634**	0.7286

Moreover, we compare our method with state-of-the-art filtering methods by smoothing the sphere and bell models, including rolling guidance normal filter [28], guided mesh normal filter [29], non-local low-rank normal filter [30], and static/dynamic filter [31]. All the data and results of the sphere and bell models are provided by Chen [32] and Meshlab. We present the visual comparison in Figs. 15 and 16, in addition, the quantitative evaluations are conducted in terms of error statistics such as the distance standard deviation (σ), the maximal of distances (d_{max}), and the mean distance (d_{mean}) between the vertices of the smoothed surface and the ground truth, which are shown in Table 2. The best metric value of shape-preserving for each model is highlighted in bold. All methods are computed using Matlab R2018a on an Intel(R) Core(TM) i5-4430 CPU @ 3.00 GHz processor.

Through the comparison with the previous works, our method outperforms most other state-of-the-art algorithms, and both the smoothing results of sphere and bell are close to the ground truths.

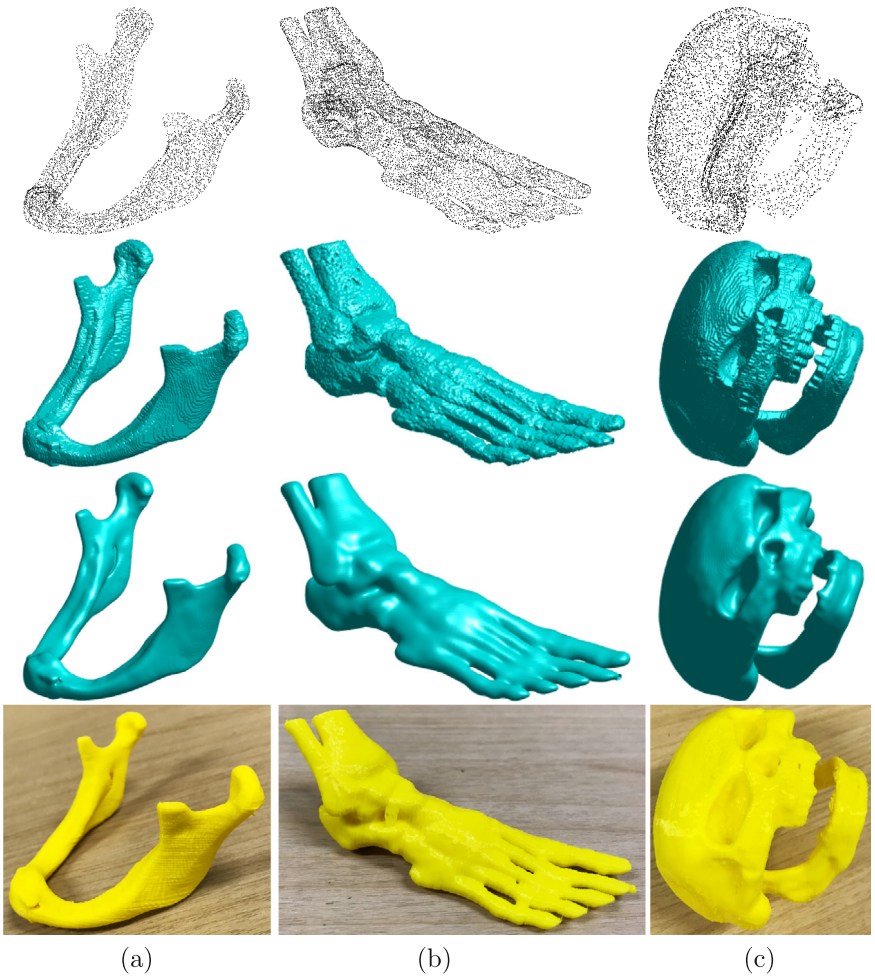

Fig. 19. The columns (a), (b), and (c) are the results of a dry mandible model of adult human, a human foot bone model, and a human skull model, respectively. The first, second, third, and fourth rows are the initial point data, zero-isosurface of ϕ at $t = 0$, zero-isosurface of ϕ at $t = 30\Delta t$, and 3D printed models, respectively.

4. 3D printing application

In this section, we introduce a 3D printing application which is processed by the proposed method. We simulate three shapes such as concentric-heart, zigzag, and honeycomb to verify the effectiveness of the proposed method. These basic patterns can be applied to the shape filling the inside of objects when performing 3D printing. The 3D models can be made by the following steps.

Step1: Take a two-dimensional point data. For the initial point data of concentric-heart, we take the two-dimensional point data as shown in Fig. 17(a), then we flip the red point data along the dash-dotted line to get the concentric-heart shape as shown in Fig. 17(b). For the initial point data of zigzag, we generate points using $y = 1.5\sin(5\pi x)$ (Fig. 17(c)), then we rotate the points 45 degree to both left and right (Fig. 17(d)). For the initial point data of honeycomb, we generate a regular hexagon with center $(0, 0)$ (Fig. 17(e)), then make a set of regular hexagons by tessellation (Fig. 17(f)).

Step2: Obtain a grid data from the initial points. We initialize the grid data as -1. For the closest grid point of an initial point, we set value one for 3×3 grid points with center of the closest grid point. As there is no gradient in the horizontal and vertical directions, there will be no noise affecting the horizontal and vertical structures of concentric-heart and the outer square frames of zigzag and honeycomb models. However, on the curved structures of concentric-heart and the inner parts of zigzag and honeycomb models, there will be noise due to the gradient. The first row of Fig. 18 shows the initial grid data.

Step3: Smoothing by using the modified Allen–Cahn equation. The second row of Fig. 18 shows the smoothing state after 10 iterations.

Step4: To construct volume data, stack the values obtained by smoothing. Add layers having -1 value to the top and bottom layers. Take zero-isosurface of the 3D grid data and make STL file. Finally, print out the constructed 3D models. The printed models of concentric-heart, zigzag, and honeycomb are shown in the third row of Fig. 18.

Manufacturing accurate medical models is one of the most useful techniques for complex medical surgery. The proposed smoothing algorithm is tested for a dry mandible model of adult human, a human foot bone model, and a human skull model. First, we start with the point data as shown in the first row of Fig. 19. The second, third, and fourth rows are the zero-isosurface of ϕ at $t = 0$, zero-isosurface of ϕ at $t = 30\Delta t$, and 3D printed models, respectively. Using the "stlread" function from file exchange of MathWorks https://kr.mathworks.com/matlabcentral/fileexchange/22409-stl-file-reader), we obtain the point data by "STL" files from website (http://www.dayin.la). For the reconstructed surfaces in second row, we reconstruct a uniform narrow volume with a distance function from the given scattered surface data, for more details, please refer to [33].

The computations are done on the domains $\Omega = (0, 3.45) \times (0, 3.65) \times (0, 2.65)$, $(0, 1.2) \times (0, 3.2) \times (0, 1.2)$, and $(0, 1.1) \times (0, 1.1) \times (0, 1.3)$ with parameters $h = 0.01$, $\Delta t = h^2/7$, $\lambda = 100$, and $\epsilon = \epsilon_2$. We obtain the smooth results after 30 iteration steps using our proposed method as shown in the third row of Fig. 19.

5. Conclusion

In this paper, we proposed a modified AC equation for curves and surfaces smoothing, which is the AC equation with a fidelity term. We demonstrated the proposed algorithm is effective and efficient for smoothing curves and surfaces. As we expected, the numerical results are smooth and preserve the initial volumes. We also computed the MSE of volume against iteration steps and compare our results with the current state of the art method. Through these results, we can see that the proposed method is able to perform smoothing procedure almost without total volume loss but with the high surface smoothing quality of visual effect. We also applied the method to smoothing 3D printed models. We introduced detailed 3D model generation methods for first-time users. Our proposed method can be useful for smoothing 3D models with rough surface.

Acknowledgments

The first author (Jian Wang) was supported by the China Scholarship Council (201808260026). Y.B. Li is supported by National Natural Science Foundation of China (No. 11601416, No. 11631012) and by the China Postdoctoral Science Foundation (No. 2018M640968). The corresponding author (J.S. Kim) was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF), Republic of Korea funded by the Ministry of Education (NRF-2016R1D1A1B03933243). The authors greatly appreciate the reviewers for their constructive comments and suggestions, which have improved the quality of this paper.

References

[1] Allen SM, Cahn JW. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. Acta Mater 1979;27(6):1085–95.
[2] Ko H, Shim H, Choi O, Kuo CCJ. Robust uncalibrated stereo rectification with constrained geometric distortions (USR-CGD). Image Vis Comput 2017;60:98–114.
[3] Taubin G. Curve and surface smoothing without shrinkage. In: Proceedings of IEEE international conference on computer vision. 1995. p. 852–57.
[4] Wei M, Zhu L, Yu J, Wang J, Pang WM, Wu J, et al. Morphology-preserving smoothing on polygonized isosurfaces of inhomogeneous binary volumes. Comput Aided Des 2015;58:92–8.
[5] Guo Y, Su Z, Lin S, Lu J, Zhong X, Luo X. 3D medical model low-pass filtering based on non-uniform spectral synthesis. Comput Aided Des 2018;104:27–35.
[6] Liu B, Cao J, Wang W, Ma N, Li B, Liu L, et al. Propagated mesh normal filtering. Comput Graph 2018;74:119–25.
[7] Yadav SK. Surface denoising based on the variation of normals and retinal shape analysis (Ph.D. thesis), Berlin: Freie Universitat; 2018.
[8] Yadav SK, Reitebuch U, Polthier K. Robust and high fidelity mesh denoising. IEEE Trans Vis Comput Graphics 2018.
[9] Yadav SK, Reitebuch U, Polthier K. Mesh denoising based on normal voting tensor and binary optimization. IEEE Trans Vis Comput Graphics 2017;24(8):2366–79.
[10] Shen Y, Barner KE. Fuzzy vector median-based surface smoothing. IEEE Trans Vis Comput Graphics 2004;10(3):252–65.
[11] Ohtake Y, Belyaev A, Bogaevski I. Mesh regularization and adaptive smoothing. Comput Aided Des 2001;33(11):789–800.
[12] Centin M, Signoroni A. Mesh denoising with (geo) metric fidelity. IEEE Trans Vis Comput Graphics 2017;24(8):2380–96.
[13] Arvanitis G, Lalos AS, Moustakas K, Fakotakis N. Feature preserving mesh denoising based on graph spectral processing. IEEE Trans Vis Comput Graphics 2019;25(3):1513–27.
[14] Liu X, Bao H, Shum HY, Peng Q. A novel volume constrained smoothing method for meshes. Graph Models 2002;64:169–82.
[15] Cai B, Xing X, Xu X. Edge/structure preserving smoothing via relativity-of-Gaussian. In: 2017 IEEE international conference on image processing (ICIP). IEEE; 2017, p. 250–4.
[16] Zheng Y, Li G, Xu X, Wu S, Nie Y. Rolling normal filtering for point clouds. Comput Aided Geom Design 2018;62:16–28.
[17] Chen H, Huang J, Remil O, Xie H, Qin J, Guo Y, et al. Structure-guided shape-preserving mesh texture smoothing via joint low-rank matrix recovery. Comput Aided Des 2019;115:122–34.
[18] Wang J, Huang J, Wang FL, Wei M, Xie H, Qin J. Data-driven geometry-recovering mesh denoising. Comput Aided Des 2019;114:133–42.
[19] Zhao Y, Qin H, Zeng X, Xu J, Dong J. Robust and effective mesh denoising using L0 sparse regularization. Comput Aided Des 2018;101:82–97.
[20] Sun X, Rosin P, Martin R, Langbein F. Fast and effective feature-preserving mesh denoising. IEEE Trans Vis Comput Graphics 2007;5:925–38.
[21] Lu X, Deng Z, Chen W. A robust scheme for feature-preserving mesh denoising. IEEE Trans Vis Comput Graphics 2016;22(3):1181–94.
[22] Yadav SK, Reitebuch U, Skrodzki M, Zimmermann E, Polthier K. Constraint-based point set denoising using normal voting tensor and restricted quadratic error metrics. Comput Graph 2018;74:234–43.
[23] Wei M, Liang L, Pang WM, Wang J, Li W, Wu H. Tensor voting guided mesh denoising. IEEE Trans Auto Sci Eng 2017;14(2):931–45.
[24] Choi Y, Jeong D, Kim J. Curve and surface smoothing using a modified Cahn–Hilliard equation. Math Probl Eng 2017;1–9.
[25] Li Y, Lee HG, Jeong D, Kim J. An unconditionally stable hybrid numerical method for solving the Allen–Cahn equation. Comput Math Appl 2010;60:1591–606.
[26] Valliammal N, Geethalakshmi SN. Hybrid image segmentation algorithm for leaf recognition and characterization. In: 2011 international conference on process automation, control and computing. 2011, p. 1–6.
[27] Mythili C, Kavitha V. Color image segmentat ion using ERKFCM. Int J Comput Appl 2012;41(20):21–8.
[28] Wang P, Fu X, Liu Y, Tong X, Liu S, Guo B. Rolling guidance normal filter for geometric processing. ACM Trans Graph 2015;34(6):173.
[29] Zhang W, Deng B, Zhang J, Bouaziz S, Liu L. Guided mesh normal filtering. Comput Graph Forum 2015;34(7):23–34.
[30] Li X, Zhu L, Fu C, Heng P. Non-local lowrank normal filtering for mesh denoising. Comput Graph Forum 2018;37(7):155–66.
[31] Zhang J, Deng B, Hong Y, Peng Y, Qin W, Liu L. Static/dynamic filtering for mesh geometry. IEEE Trans Vis Comput Graphics 2019;25(4):1774–87.
[32] Chen H, Huang J, Remil O, Xie H, Qin J, Guo Y, et al. Structure-guided shape-preserving mesh texture smoothing via joint low-rank matrix recovery. Comput Aided Des 2019;115:122–34.
[33] Li Y, Kim J. Fast and efficient narrow volume reconstruction from scattered data. Pattern Recognit 2015;48(12):4057–69.