



ELSEVIER

Contents lists available at ScienceDirect

## Pattern Recognition

journal homepage: [www.elsevier.com/locate/pr](http://www.elsevier.com/locate/pr)

## Fast and efficient narrow volume reconstruction from scattered data

Yibao Li<sup>a</sup>, Junseok Kim<sup>b,\*</sup><sup>a</sup> School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China<sup>b</sup> Department of Mathematics, Korea University, Seoul 136-713, Republic of Korea

## ARTICLE INFO

## Article history:

Received 13 August 2014

Received in revised form

20 May 2015

Accepted 23 June 2015

Available online 2 July 2015

## Keywords:

Offset surface reconstruction

Allen–Cahn equation

Unsigned distance function

Narrow band domain

Unconditional stability

## ABSTRACT

We describe a fast and efficient numerical algorithm for the process of three-dimensional narrow volume reconstruction from scattered data in three dimensions. The present study is an extension of previous research [Li et al., Surface embedding narrow volume reconstruction from unorganized points, *Comput. Vis. Image Underst.* 121 (2014) 100–107]. In the previous work, we modified the original Allen–Cahn equation by multiplying a control function to restrict the evolution within a narrow band around the given surface data set. The key idea of the present work is to perform the computations only on a narrow band around the given surface data set. In this way, we can significantly reduce the storage memory and CPU time. The proposed numerical method, based on operator splitting techniques, can employ a large time step size and exhibits unconditional stability. We perform a number of numerical experiments in order to demonstrate the efficiency of this method.

© 2015 Elsevier Ltd. All rights reserved.

## 1. Introduction

In this study, we consider a fast and efficient numerical algorithm for three-dimensional (3D) volume reconstruction from scattered data in three dimensions. Scattered data is defined as a set of data with no specified ordering and connectivity between data points. In practice, the unorganized sample points in  $\mathbb{R}^3$  for surface reconstruction can stem from a variety of sources, including medical imagery, laser range scanners, contact probe digitizers, radar and seismic surveys, or mathematical models such as implicit surfaces [1]. Volume reconstruction from scattered data represents an important task. For example, a reverse engineering problem involves reconstructing 3D models from unorganized points that are generated by 3D surface scanning devices [2]. However, volume reconstruction is a challenging problem because point clouds lack ordering information and connectivity, and are usually noisy [3].

Many approaches to surface reconstruction from scattered points exist. Most surface reconstruction methods for point clouds can be classified as either explicit or implicit surface methods, depending on the form of the representation of the surface [4]. In explicit surface representations, the surface location and the geometry are prescribed in an explicit manner. For example,

Boissonnat [5] suggested the use of Delaunay triangulations to construct a single connected shape of a point set. This method progressively eliminates tetrahedra from the Delaunay triangulation based on their circumspheres. Typically, in implicit surface reconstruction methods, a signed distance scalar function is constructed on a fixed rectangular grid such that the given scattered data are close to the zero level set of the function. The final 3D shape is the zero isosurface of the signed distance function [3,6–9]. Ye et al. [3] proposed a novel fast method for implicit surface reconstruction from unorganized point clouds. Their algorithm employs a computationally efficient multigrid solver on a narrow band of the level set function that represents the reconstructed surface.

In this study, we focus on implicit representations, and our approach is based on a phase-field model defined by the Allen–Cahn (AC) equation [10]. The AC equation has an intrinsic smoothing effect on interfacial transition layers and the motion by mean curvature. In the level set framework, an explicit time integration scheme is a general choice for the mean curvature flow, which requires small time steps in order to ensure numerical stability. However, for the AC equation a fast and accurate hybrid numerical solver is available [11], which is the main reason why we choose this equation. In our previous work [12], we presented a fast and accurate numerical method for surface embedding narrow volume reconstruction with a fixed distance from an unorganized surface data set. Fig 1(a), (b), (c), and (d) shows a given set of scattered data, its 3D reconstruction, and the cut view and cross view of the reconstructed volume, respectively.

\* Corresponding author.

E-mail addresses: [yibaoli@mail.xjtu.edu.cn](mailto:yibaoli@mail.xjtu.edu.cn) (Y. Li), [cfdkim@korea.ac.kr](mailto:cfdkim@korea.ac.kr) (J. Kim).URLS: <http://gr.xjtu.edu.cn/web/yibaoli> (Y. Li),<http://math.korea.ac.kr/~cfdkim> (J. Kim).

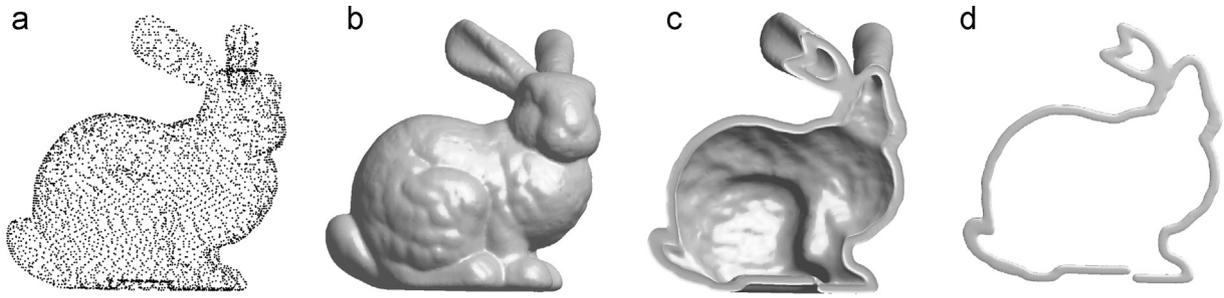


Fig. 1. (a) Point data set, (b) reconstruction, (c) cut view, and (d) cross view of a reconstructed volume.

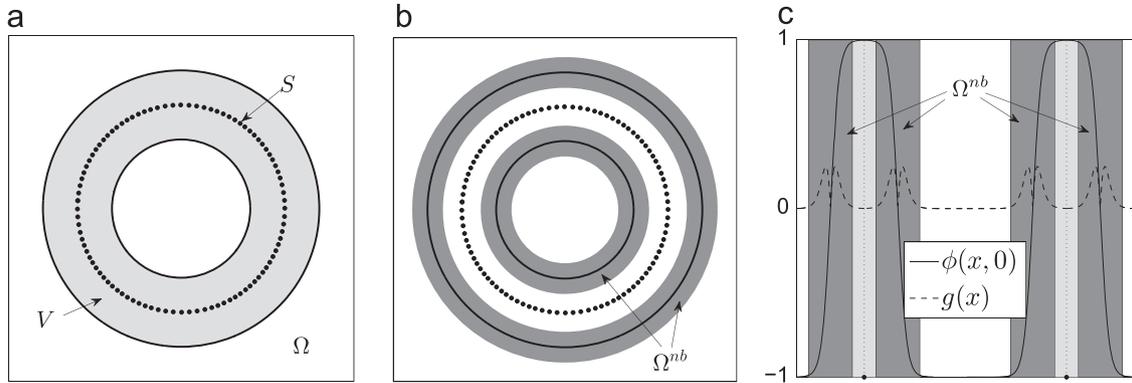


Fig. 2. (a)  $S$  is the given scattered data,  $V$  is the volume embedding  $S$ , and  $\Omega$  is the global domain containing  $V$ . (b) Narrow band domain  $\Omega^{nb} = \{\mathbf{x} \mid |\phi(\mathbf{x}, 0)| < 1 - \delta\} \cap \Omega$ . Here,  $\delta$  is a small positive value. (c) Sectional view of the new edge stopping function  $g(\mathbf{x})$ .

Note that our problem is also similar to the offset surface reconstruction problem, which can also be defined as a surface whose points are at a fixed normal distance from a given surface. Many algorithms have been proposed for solving offset surface reconstruction problems. For example, by using an implicit function, Liu and Wang [13] approximated both the zero-level surface and its offset surface. Subsequently, they also developed a fast offset surface generation method via a narrow band signed distance-field [14]. In addition, Chen and Wang proposed thickening operations for converting a surface to a solid [15], and introduced a uniform offsetting model that enables the generation of both enlarged and contracted models from an arbitrary offset distance [16]. Lien [17] and Varadhan and Manocha [18] proposed a method that generates point-based Minkowski sum boundaries. Curless and Levoy developed two important techniques for reconstructing complex and accurate models from scanned objects. The first is spacetime analysis, a ranging method based on analyzing the time evolution of the structured light reflections [19], and the second is a volumetric space carving technique for integrating several data into a single geometric model [20]. In order to obtain high-quality offsets, an adaptive octree-structure was used for distance bounds in [21]. Small and thin features were detected using subdivisional methods [22,23]. For some other methods of offset surface reconstruction, we also refer the reader to [24–30].

The main purpose of the present work is to perform the computations only on a narrow band around the given surface data set, in order to achieve significant reduction in both the storage memory and the CPU time. In addition, the proposed numerical method can use large time step sizes, and exhibits unconditional stability. The proposed numerical scheme has the advantage that the narrow domain can be theoretically defined, and its boundary condition can be defined simply as a Dirichlet boundary condition without loss of accuracy. It should be pointed

out that the proposed method is simpler and more efficient than the standard adaptive octree-structure method.

The rest of this paper is organized as follows. In Section 2, we briefly describe the main governing equation. We describe the numerical solution algorithm in Section 3. In Section 4, we perform some characteristic numerical experiments for volume reconstruction. Finally, our conclusions are presented in Section 5.

## 2. Phase-field method

For scattered surface data points  $S = \{\mathbf{x}_p = (X_p, Y_p, Z_p) \in \mathbb{R}^3 \mid p = 1, \dots, M\}$ , where  $M$  is the number of data points, we want to reconstruct a uniform narrow volume with a distance  $l$  from the given unorganized surface data. In order to find a smooth narrow volume, we presented the following partial differential equation in our previous study [12]:

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = g(\mathbf{x}) \left( -\frac{F'(\phi(\mathbf{x}, t))}{\epsilon^2} + \Delta \phi(\mathbf{x}, t) \right), \tag{1}$$

$$\phi(\mathbf{x}, 0) = \tanh\left(\frac{l - d(\mathbf{x})}{\sqrt{2}\xi}\right), \tag{2}$$

$$g(\mathbf{x}) = 1 - \phi^2(\mathbf{x}, 0), \tag{3}$$

where  $\phi \in [-1, 1]$  is the order parameter with  $\phi = 1$  and  $\phi = -1$  inside and outside of the reconstructed narrow volume, respectively.  $\phi = 0$  is interpreted as the surface of the volume.  $F(\phi) = 0.25(\phi^2 - 1)^2$ ,  $\epsilon$  and  $\xi$  are positive constants.  $d(\mathbf{x})$  is the unsigned distance function from the surface. From (2), we can see that if  $d(\mathbf{x}) = l$ , then  $\phi(\mathbf{x}, 0) = 0$ , which means that the initial guess of  $\phi$  is sufficiently close to the exact solution. If  $g(\mathbf{x}) \equiv 1$ , then (1) becomes the classical AC equation [10].

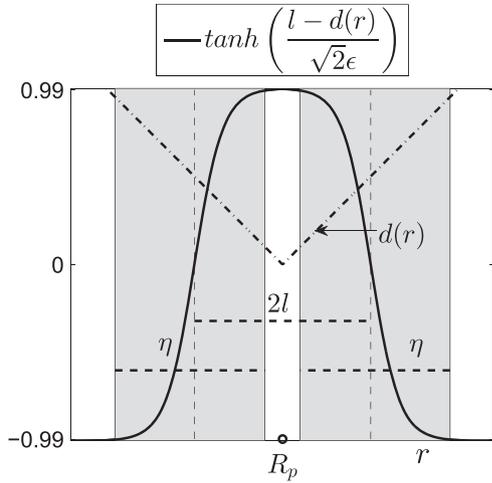


Fig. 3. A phase-field profile,  $\tanh[(l-d(r))/(\sqrt{2}\epsilon)]$ . Here,  $\eta = 2mh$  and  $R_p$  is a given data point in one-dimensional space. The gray region represents  $-0.99 \leq \phi \leq 0.99$ .

In this study, we propose a narrow band computation of (1) with the following new edge stopping function:

$$g(\mathbf{x}) = (1 - |\phi(\mathbf{x}, 0)|) |\phi(\mathbf{x}, 0)|. \tag{4}$$

For practical computations, (4) yields better results than (3). Let  $S$  be the given scattered data,  $V$  be the volume embedding  $S$ , and  $\Omega$  be the global domain containing  $V$  (see Fig. 2(a)). To reduce the computational cost, we only perform the evolution within a narrow band domain  $\Omega^{nb} = \{\mathbf{x} \mid |\phi(\mathbf{x}, 0)| < 1 - \delta\} \cap \Omega$  (Fig. 2(b)). Here,  $\delta$  is a small positive value, and in particular we choose  $\delta = 0.01$ . Fig. 2(c) shows the new edge stopping function  $g(\mathbf{x})$ . The following Dirichlet boundary condition is used on  $\partial\Omega^{nb}$ :

$$\phi(\mathbf{x}, t) = \begin{cases} 1 - \delta & \text{if } \mathbf{x} \in \{\mathbf{x} \mid \phi(\mathbf{x}, 0) > 0\} \cap \partial\Omega^{nb}, \\ -1 + \delta & \text{if } \mathbf{x} \in \{\mathbf{x} \mid \phi(\mathbf{x}, 0) < 0\} \cap \partial\Omega^{nb}. \end{cases} \tag{5}$$

This narrow band calculation will reduce the computational cost, without compromising the resolution.

### 3. Numerical solution algorithm

For simplicity of exposition, let us first consider one-dimensional space. For an equilibrium phase-field profile,  $\tanh[r/(\sqrt{2}\epsilon)]$ , of (1), the phase-field varies from  $-0.99$  to  $0.99$  over a distance of approximately  $\eta = 2\sqrt{2}\epsilon \tanh^{-1}(0.99)$  [12,31] (refer to Fig. 3). Therefore, if we want this value to constitute  $2m$  grid points, i.e.,  $\eta = 2mh$ , then the  $\epsilon$  value needs to be taken as  $\epsilon_m = mh/[\sqrt{2} \tanh^{-1}(0.99)]$ , where  $h$  is a space step size [31]. From the definition of  $\phi(r, 0)$  in (2), we see that if  $d(r) = |r - R_p|$  is larger than  $l + \eta/2$  or smaller than  $l - \eta/2$ , then  $\phi(r, 0)$  takes values of  $-0.99$  or  $0.99$ , respectively. Here,  $R_p$  is a given data point. In one-dimensional space, the narrow band region is  $(R_p - l - \eta/2, R_p + l + \eta/2) - (R_p - l + \eta/2, R_p + l - \eta/2)$  (see the gray region in Fig. 3).

Let  $\Omega = (X_{min}, X_{max}) \times (Y_{min}, Y_{max}) \times (Z_{min}, Z_{max})$  be a 3D domain that embeds the given scattered data  $X_p = (X_p, Y_p, Z_p)$  for  $p = 1, \dots, M$ . The space step size  $h$  is defined as  $h = \frac{X_{max} - X_{min}}{N_x} = \frac{Y_{max} - Y_{min}}{N_y} = \frac{Z_{max} - Z_{min}}{N_z}$ . Here,  $N_x$ ,  $N_y$ , and  $N_z$  are the numbers of cells in the  $x$ -,  $y$ - and  $z$ -directions, respectively. The center of each cell is located at  $\mathbf{x}_{ijk} = (x_i, y_j, z_k)$  where

$x_i = X_{min} + (i - 1)h$ ,  $y_j = Y_{min} + (j - 1)h$ , and  $z_k = Z_{min} + (k - 1)h$ , for  $i = 1, \dots, N_x, j = 1, \dots, N_y$ , and  $k = 1, \dots, N_z$ . Let  $\phi_{ijk}$  be an approximation to  $\phi(x_i, y_j, z_k)$ . Then, let us define a truncated unsigned distance function  $d_{ijk}$  on  $\Omega_h = \{\mathbf{x}_{ijk} \mid i = 1, \dots, N_x, j = 1, \dots, N_y, \text{ and } k = 1, \dots, N_z\}$  as

$$d_{ijk} = \begin{cases} \min_{1 \leq p \leq M} \|\mathbf{x}_{ijk} - \mathbf{X}_p\| & \text{if } \min_{1 \leq p \leq M} \|\mathbf{x}_{ijk} - \mathbf{X}_p\| \leq l + \eta/2, \\ h\sqrt{N_x^2 + N_y^2 + N_z^2} & \text{otherwise.} \end{cases} \tag{6}$$

The discrete narrow band domain is defined as  $\Omega_h^{nb} = \{\mathbf{x}_{ijk} \mid l - \eta/2 \leq d_{ijk} \leq l + \eta/2\}$ . In order to avoid the computation of the distance function in the whole domain, we define  $\Omega^{nb}$  as the union of a collection of domains  $\Omega_p^{nb}$ , i.e.,  $\Omega^{nb} = \bigcup_{p=1}^M \bar{\Omega}_p^{nb}$ . Here, we have

$$\begin{aligned} & \left\{ (x_i, y_j, z_k) \mid |x_i - X_p| \leq l + \frac{\eta}{2}, |y_j - Y_p| \leq l + \frac{\eta}{2}, |z_k - Z_p| \leq l + \frac{\eta}{2} \right\} \\ & - \left\{ (x_i, y_j, z_k) \mid \|\mathbf{x}_{ijk} - \mathbf{x}_p\| \leq l - mh \right\} \\ & \subset \left\{ (x_i, y_j, z_k) \mid \left| \frac{X_p - X_{min} - l - (m-1)h}{h} \leq i \leq \frac{X_p - X_{min} + l + (m+1)h}{h}, \right. \right. \\ & \quad \left. \frac{Y_p - Y_{min} - l - (m-1)h}{h} \leq j \leq \frac{Y_p - Y_{min} + l + (m+1)h}{h}, \right. \\ & \quad \left. \frac{Z_p - Z_{min} - l - (m-1)h}{h} \leq k \leq \frac{Z_p - Z_{min} + l + (m+1)h}{h} \right\} \\ & - \left\{ (x_i, y_j, z_k) \mid \|\mathbf{x}_{ijk} - \mathbf{x}_p\| \leq l - mh \right\} \\ & \subset \left\{ (x_i, y_j, z_k) \mid \left[ \frac{X_p - X_{min} - l - (m-1)h}{h} \right] \leq i \leq \left[ \frac{X_p - X_{min} + l + (m+2)h}{h} \right], \right. \\ & \quad \left[ \frac{Y_p - Y_{min} - l - (m-1)h}{h} \right] \leq j \leq \left[ \frac{Y_p - Y_{min} + l + (m+2)h}{h} \right], \\ & \quad \left[ \frac{Z_p - Z_{min} - l - (m-1)h}{h} \right] \leq k \leq \left[ \frac{Z_p - Z_{min} + l + (m+2)h}{h} \right] \right\} \\ & - \left\{ (x_i, y_j, z_k) \mid \|\mathbf{x}_{ijk} - \mathbf{x}_p\| \leq l - mh \right\} = \bar{\Omega}_{h,p}^{nb}. \end{aligned} \tag{7}$$

In the above, we have used the definition of  $\mathbf{x}_{ijk}$  and  $[x]$  is the largest integer not greater than  $x$ . We order  $\phi_{ijk}$  on  $\Omega_h^{nb}$  by using lexicographical ordering for  $(i, j, k)$ , i.e.,  $\phi_s = \phi_{ijk}$  for  $s = 1, \dots, L$  and some  $ijk$ . Here,  $L$  is the total number of computational cells in  $\Omega_h^{nb}$ . Next, we present an operator splitting-based hybrid numerical scheme for (1) on the discrete narrow band domain  $\Omega_h^{nb}$ . We split the original problem (1) into a sequence of simpler problems as

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = g(\mathbf{x}) \Delta \phi(\mathbf{x}, t), \tag{8}$$

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = -\frac{g(\mathbf{x})F'(\phi(\mathbf{x}, t))}{\epsilon^2}. \tag{9}$$

As a first step, we solve Eq. (8) by applying an implicit method and Dirichlet boundary condition (5). That is,

$$\frac{\phi_s^{n+1/2} - \phi_s^n}{\Delta t} = g_s \Delta_d \phi_s^{n+1/2} \quad \text{for } s = 1, \dots, L, \tag{10}$$

where  $g_s = (1 - |\phi_s^0|) |\phi_s^0|$  and the discrete Laplacian operator is given as

$$\Delta_d \phi_s = (\phi_{x_l s} + \phi_{x_h s} + \phi_{y_l s} + \phi_{y_h s} + \phi_{z_l s} + \phi_{z_h s} - 6\phi_s)/h^2. \tag{11}$$

Here,  $\phi_{x_l s} = \phi_{i-1, j, k}$  and  $\phi_{x_h s} = \phi_{i+1, j, k}$  are the values of the left and right cells of  $\phi_s$  in the  $x$ -direction, respectively. The other terms are similarly defined. For the boundary values, we use either  $0.99$  or  $-0.99$  depending on the sign of  $\phi_s$ . The data structure and initialization steps are described in the box below.

In the first step, we set  $f(\mathbf{x}) = 0$  and  $l(\mathbf{x}) = 0$  everywhere in the computational domain  $\Omega$ . Here, the index function  $l$  is used to store

Let  $ijklloop$  denote

for ( $k=1; k \leq N_z; k++$ ) for ( $j=1; j \leq N_y; j++$ ) for ( $i=1; i \leq N_x; i++$ ).

Initialize  $d$ ,  $f$ , and  $I$  as  $ijklloop$   $d_{ijk} = h\sqrt{N_x^2 + N_y^2 + N_z^2}$ ,  $f_{ijk} = I_{ijk} = 0$ .

And then, we take the following three steps:

*Step 1:* for ( $p=1; p \leq M; p++$ )

for ( $i = \lfloor \frac{X_p - X_{min} - l - (m-1)h}{h} \rfloor; i \leq \lfloor \frac{X_p - X_{min} + l + (m+2)h}{h} \rfloor; i++$ )

for ( $j = \lfloor \frac{Y_p - Y_{min} - l - (m-1)h}{h} \rfloor; j \leq \lfloor \frac{Y_p - Y_{min} + l + (m+2)h}{h} \rfloor; j++$ )

for ( $k = \lfloor \frac{Z_p - Z_{min} - l - (m-1)h}{h} \rfloor; k \leq \lfloor \frac{Z_p - Z_{min} + l + (m+2)h}{h} \rfloor; k++$ ) {

if ( $d_{ijk} > \|\mathbf{x}_{ijk} - \mathbf{X}_p\|$ )  $d_{ijk} = \|\mathbf{x}_{ijk} - \mathbf{X}_p\|$ ;

$ijklloop$  if ( $l - \eta/2 \leq d_{ijk} \leq l + \eta/2$ )  $f_{ijk} = 1$ ;

*Step 2:* Let  $s = 0$ .

$ijklloop$  { if ( $f_{ijk} == 1$ ) {  $s++$ ;  $\phi_s^0 = \tanh((l - d_{ijk})/(\sqrt{2}\xi))$ ;

$g_s = (1 - |\phi_s^0|)|\phi_s^0|$ ;  $I_{ijk} = s$ }; Set  $L = s$ ;

*Step 3:* Set  $s = 0$ ,  $\phi_{L+1}^0 = 0.99$ , and  $\phi_{L+2}^0 = -0.99$ .

$ijklloop$  { if ( $f_{ijk} == 1$ ) {  $s++$ ;  $xl_s = I_{i-1,j,k}$ ;  $xh_s = I_{i+1,j,k}$ ;

$yl_s = I_{i,j-1,k}$ ;  $yh_s = I_{i,j+1,k}$ ;  $zl_s = I_{i,j,k-1}$ ;  $zh_s = I_{i,j,k+1}$ ;

if ( $xl_s == 0$ ) { if ( $\phi_s^0 > 0$ )  $xl_s = L+1$ ; else  $xl_s = L+2$ };

if ( $xh_s == 0$ ) { if ( $\phi_s^0 > 0$ )  $xh_s = L+1$ ; else  $xh_s = L+2$ };

if ( $yl_s == 0$ ) { if ( $\phi_s^0 > 0$ )  $yl_s = L+1$ ; else  $yl_s = L+2$ };

if ( $yh_s == 0$ ) { if ( $\phi_s^0 > 0$ )  $yh_s = L+1$ ; else  $yh_s = L+2$ };

if ( $zl_s == 0$ ) { if ( $\phi_s^0 > 0$ )  $zl_s = L+1$ ; else  $zl_s = L+2$ };

if ( $zh_s == 0$ ) { if ( $\phi_s^0 > 0$ )  $zh_s = L+1$ ; else  $zh_s = L+2$ };}

the index of  $\phi_s$  and the flag control function  $f$  is used to determine which point is located in the narrow domain. To make sure that the initialized function  $d$  is larger than the longest distance from the surface points in  $\Omega_h$ ,  $p^{nb}$ , we set  $d = h\sqrt{N_x^2 + N_y^2 + N_z^2}$  in the initial condition. The value of the distance function at a point located in the domain  $\Omega_{h,p}$  is determined only by its neighboring input data points  $\mathbf{X}_p = (X_p, Y_p, Z_p)$  that have smaller distance values. Meanwhile, by setting a flag control function  $f(\mathbf{x})$ , we tag the cells that are located in  $\Omega_h^{nb}$ . The total computation requires  $O(M(m+l/h)^3)$  operations. Subsequently, by using the control function  $f(\mathbf{x})$ , we store the grids and record the index of grids that are located in domain  $\Omega_{h,p}^{nb}$ . Finally, we define the order parameter  $\phi$  and the control function  $g$ .

It is clear that if  $s$  is connected with  $(i, j, k)$ , then  $xl_s$  is connected with  $(i-1, j, k)$ . Therefore, our connecting method is simple, and this is described in step 3. There are two points that should be

noted. First, the order of the stored grids in steps 2 and 3 should be same. Second, there are some points whose connected grids are not stored. In fact, the distances from these points are located at the boundary of  $\Omega_h^{nb}$  (see the circle-mark in Fig. 4). It can be seen from the definition of  $\phi$  in (2) that its value is approximately equal to either one or negative one, and its sign depends on the sign of  $\phi$  (refer to Fig. 4).

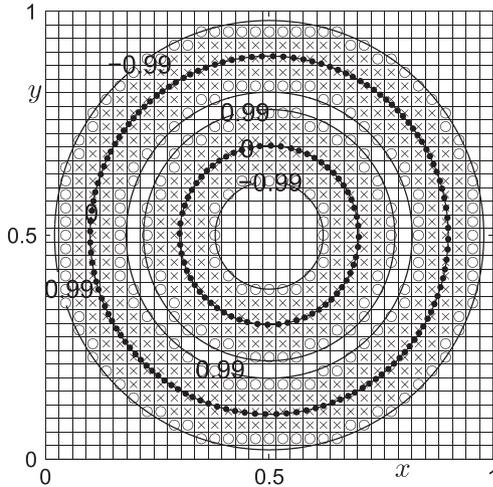
The linear system given by (10) can be solved using the Gauss-Seidel relaxation method, with a tolerance of  $1E-6$ . Then, with  $\phi_s^{n+1/2}$  as the solution at  $t = n\Delta t$ , we solve (9) at  $t = (n+1)\Delta t$  as

$$\phi_s^{n+1} = \phi_s^{n+1/2} \sqrt{e^{-\frac{2g_s \Delta t}{\epsilon^2}} + (\phi_s^{n+1/2})^2 (1 - e^{-2g_s \Delta t / \epsilon^2})}. \quad (12)$$

For more details about the numerical solution algorithm, see [11,12,32–34]. Our proposed hybrid splitting method, Eqs. (10) and (12), is an unconditionally stable scheme, which was proved in our previous work [12].

### 4. Numerical results

In this section, we present numerical results on various synthetic and real data sets, using the proposed numerical algorithm. We will stop the numerical computations when the relative error of the order parameter,  $\|\phi^{n+1} - \phi^n\|_2 / \|\phi^n\|_2$ , attains a value that is less than a given tolerance,  $tol = 1E-5$ . Here,  $\|\phi^n\|_2$  denotes the discrete  $l_2$ -norm of  $\phi^n$ . Throughout the rest of paper, unless otherwise specified, we will use  $m = 2, \xi = \epsilon_2, \epsilon = \epsilon_6, l = 4h$ , and the time step  $\Delta t = \epsilon^2$ . Note that for the purpose of achieving a better



**Fig. 4.** Schematic illustration of the computational domain and the mesh used in two-dimensional space. Here, the symbols ‘x’ and lines represent offset surfaces and contours of the order parameter  $\phi(\mathbf{x}, 0)$ , respectively. Symbols ‘o’, and the x-mark are used to denote mesh grids.

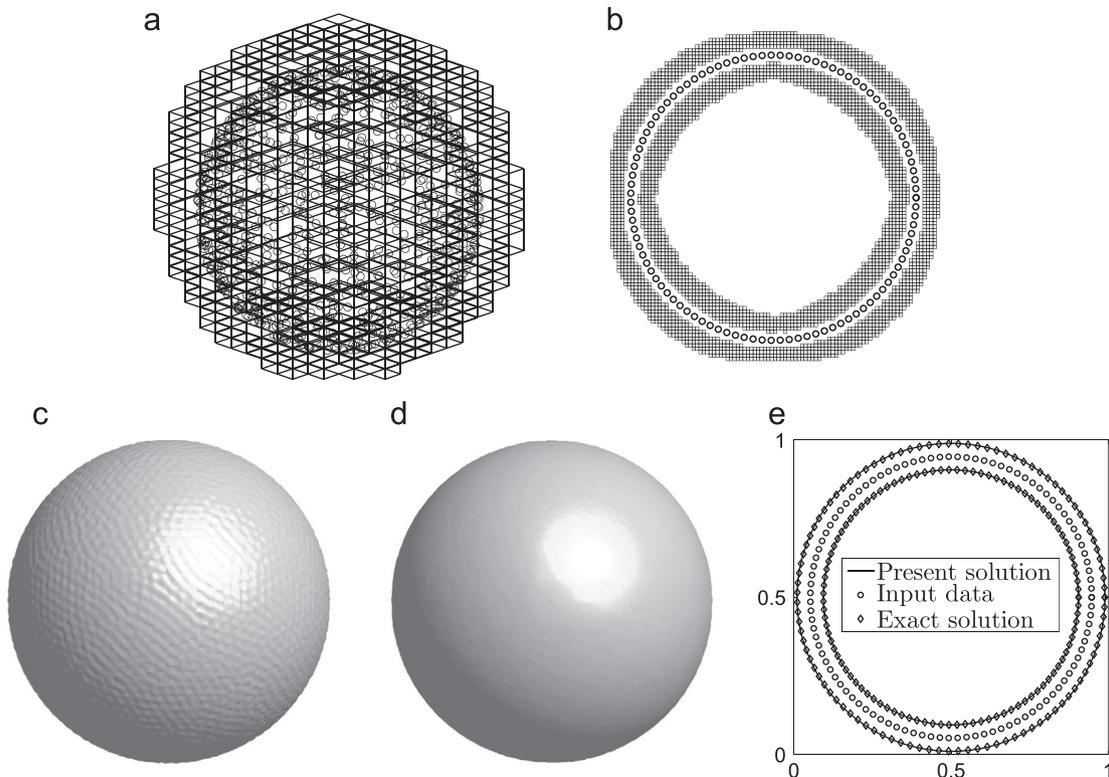
visualization, the input data points are displayed more sparsely than the real density.

In order to demonstrate the basic mechanism of the algorithm, we start with the simple example of a sphere placed at the center of the unit domain, with a radius 0.45. The computation is run over six iterations, with  $h = 0.01$ . Fig. 5(a) and (b) shows the input data set (circles) with bounding boxes in the whole view and the cut view, respectively. Fig. 5(c–e) shows the initial reconstruction, the final reconstruction, and a comparison between the numerical solution and the exact solution. As can be seen, our method performs well for the reconstruction of the offset surface. The CPU times required for the initial processing and the surface reconstruction were 0.202 s and 1.607 s, respectively. Our method achieves the reconstruction very quickly, because the number of the grid points used is 362,715, which is only about 35.6% of all grid points.

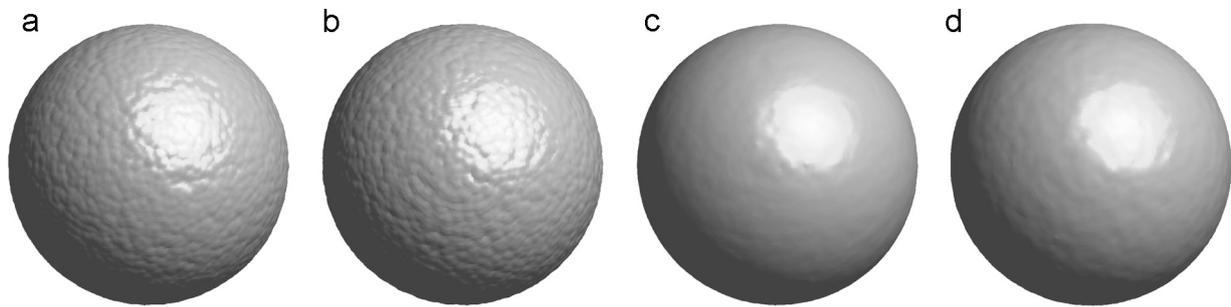
To test the accuracy of our proposed method, we perform numerical tests using a cube, a sphere, and a cylinder, where we could actually compute the errors using theoretical analysis. We

**Table 1**  
Accuracy test for our proposed method. Here, we use the offset distance  $l = 0.1$ .

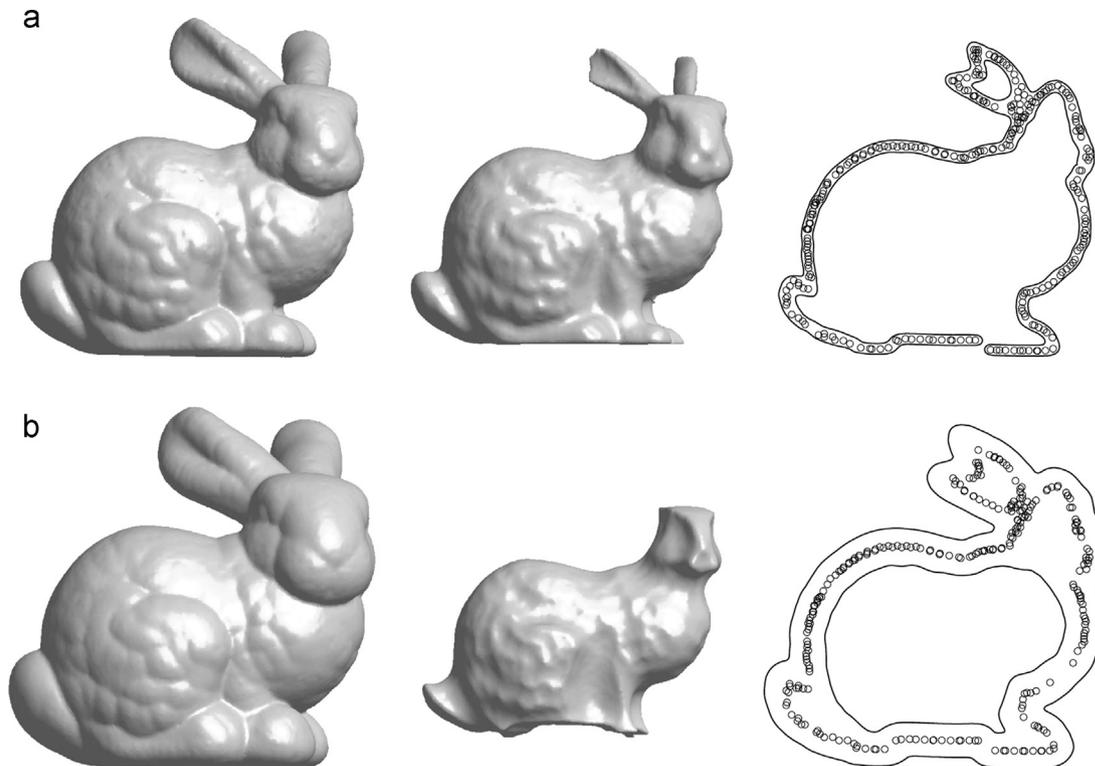
Case	Space size	Narrow domain		Results in [16]	
		Maximum error	Average error	Maximum error	Average error
Cube	$h = 0.01$	$5.241E-2$	$9.517E-3$	$5.0E-6$	$1.0E-6$
	$h = 0.005$	$4.054E-2$	$2.456E-3$		
Sphere	$h = 0.01$	$1.307E-3$	$9.938E-4$	$1.311E-3$	$2.224E-4$
	$h = 0.005$	$3.723E-4$	$2.173E-4$		
Cylinder	$h = 0.01$	$9.346E-4$	$6.460E-4$	$1.263E-3$	$2.46E-4$
	$h = 0.005$	$3.155E-4$	$1.527E-4$		



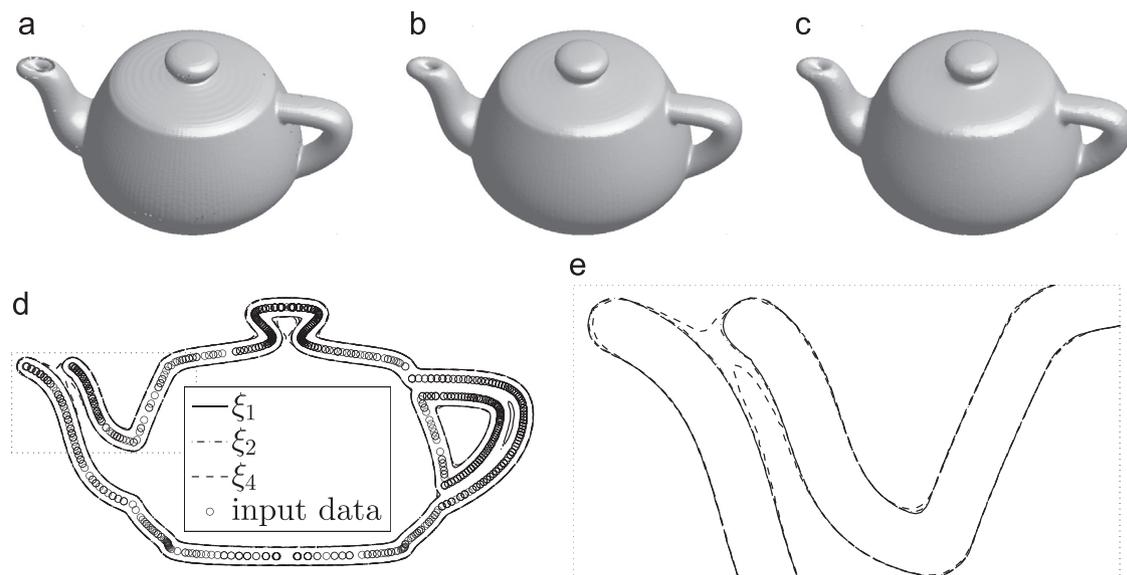
**Fig. 5.** Volume reconstruction for the sphere. (a) Input data set (circles) with computational domain (bounding boxes). (b) Cross view of (a). (c) Initial reconstruction. (d) Final reconstruction. (e) Comparison between numerical solution and exact solution. Note that for a better visualization, the points and mesh points used in (a) are displayed more sparsely than the real density.



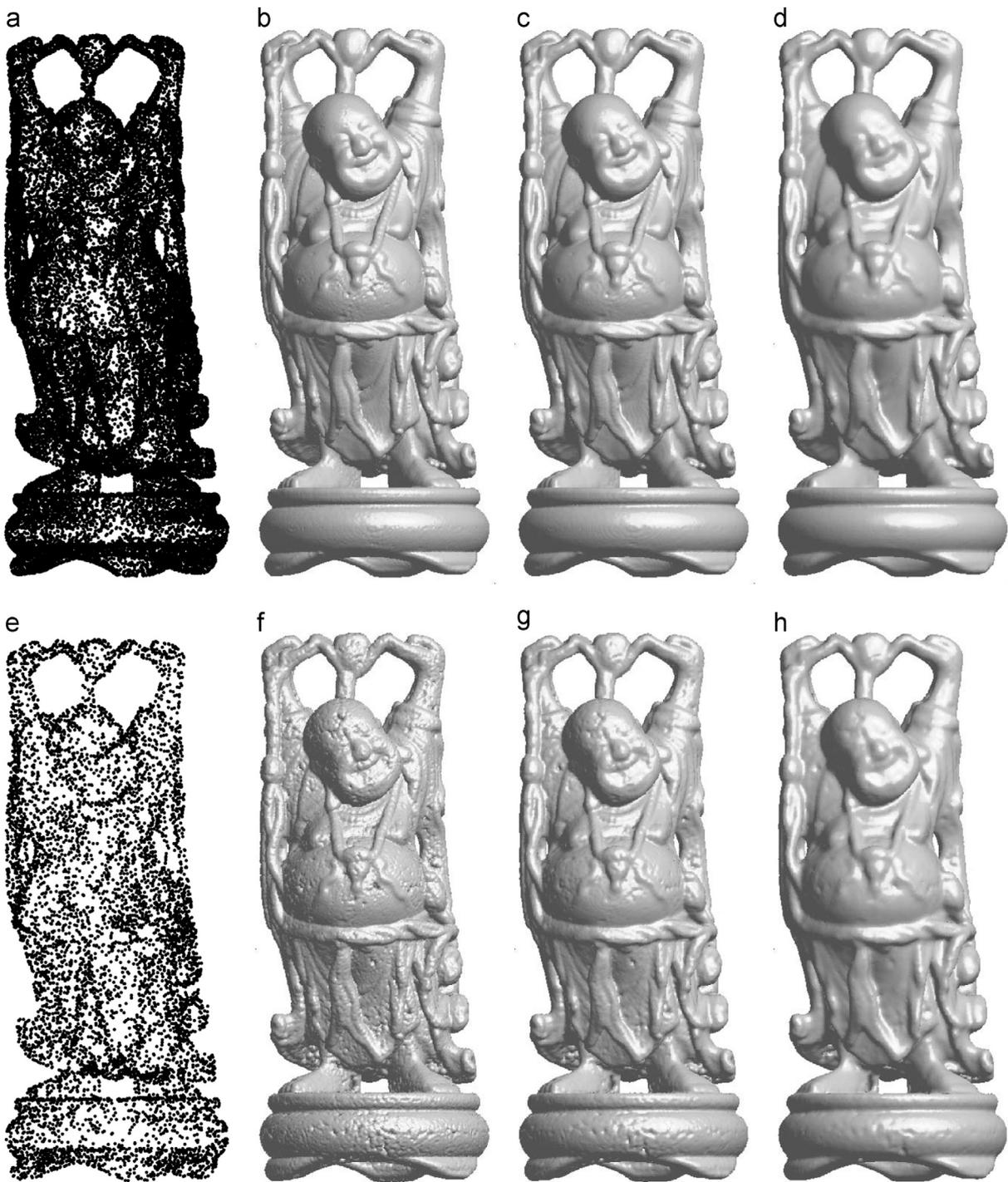
**Fig. 6.** Volume reconstructions for the input surface with random noises. (a) and (b) are isosurfaces of the distance function for 5% and 10% random noises, respectively. (c) and (d) are reconstructed volumes for (a) and (b), respectively.



**Fig. 7.** Volume reconstructions generated by our algorithm with different offset distances. (a)  $l=4h$  and (b)  $l=12h$ . From left to right, these are the outer surface, inner surface, and offset surface in the cross view, respectively. Here, the solid line and circle correspond to the offset surface and input data, respectively.



**Fig. 8.** Volume reconstruction with different values of  $\xi$ . (a)  $\xi_1$ . (b)  $\xi_2$ . (c)  $\xi_4$ . (d) Comparison of the three offset surfaces in the cross view. (e) Closeup view of the boxed region in (d).

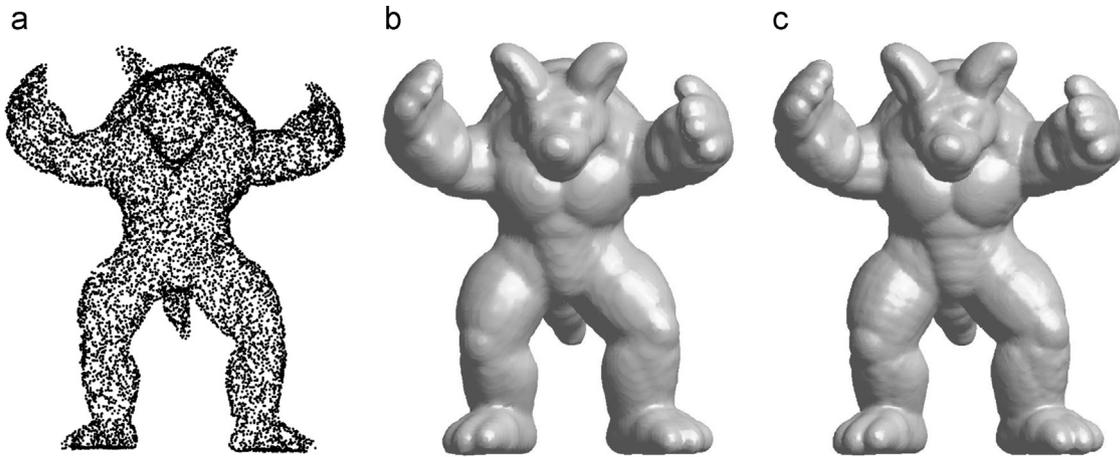


**Fig. 9.** Reconstructed volumes with different values of  $\xi$ . The size of the input data in the bottom row is smaller than the top row by a factor 6. (a) and (e) show the input data. (b) and (f) show the initial reconstruction. (c) and (g) are the final reconstructions with  $\epsilon_6$ . (d) and (h) are the final reconstructions with  $\epsilon_{12}$ .

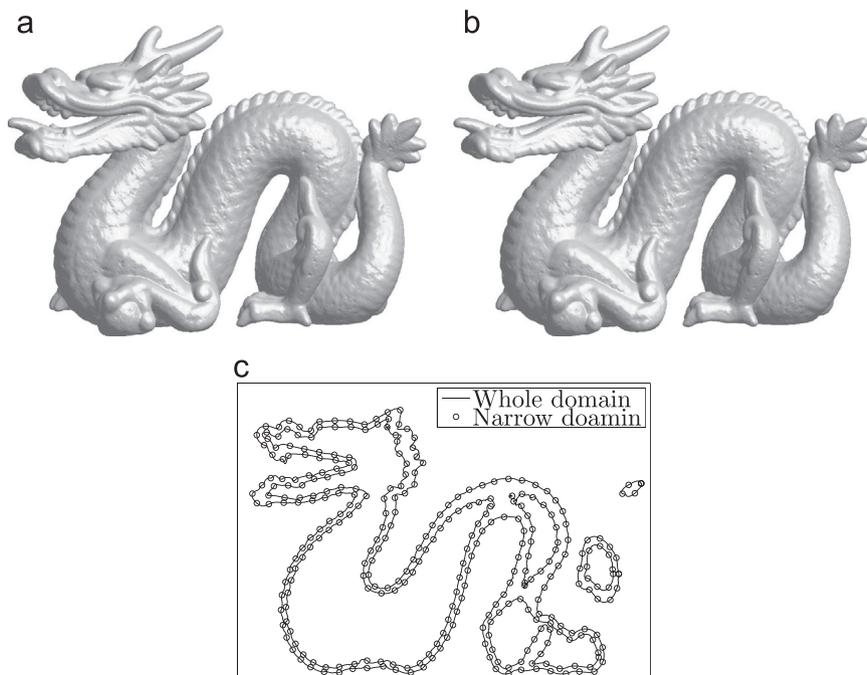
define the error of a grid as the difference between the numerical solution and the exact solution as follows:  $e_m : \mathbf{X}_m^n - \mathbf{X}_m^e$ , where  $\mathbf{X}_m^n$  and  $\mathbf{X}_m^e$  are the numerical and exact surface points. These computations are run up to the steady condition with  $l=0.1$ . The side length of the cube, the diameter of the sphere, the diameter of the cylinder, and the height of the cylinder are set to one. Table 1 shows the maximum and average errors with two different space grid sizes. To compare the current results with the those from a polygonal offsetting method [16], we combine them together. Since the results in [16] are measured in the shrink surface, we also list our results in the shrink surface. We observe that the

numerical results obtained from our proposed method are qualitatively in good agreement with the theoretical values and the results in [16]. Note that this comparison is in some way unfair, because the polygonal offsetting method presented in [16] can achieve a high accuracy in the sharp surface, and on the other hand, our approach can reduce outliers or conflicting points.

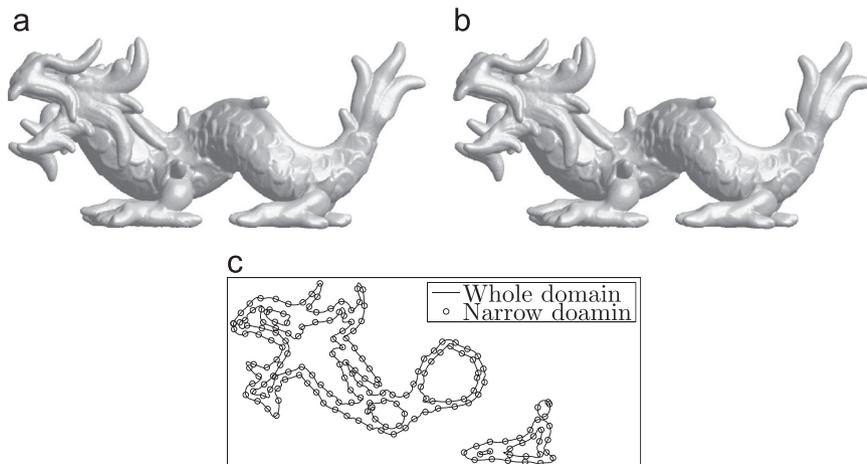
In Fig. 6(a)–(d), we show offset surface reconstructions with 5% and 10% noises. The input surface is a sphere with radius 0.5. The offset distance  $l=0.1$ ,  $\xi=\epsilon_4$ ,  $\epsilon=\epsilon_{12}$ , and  $h=0.01$  are chosen. The average errors are  $2.242E-2$  and  $3.755E-2$  for surface reconstructions with 5% and 10% random noises, respectively. These



**Fig. 10.** Volume reconstructions generated by our algorithm with different offset distances. (a) Initial data, (b) final reconstruction on a coarse grid, and (c) reconstruction on a fine grid.



**Fig. 11.** Comparison between the whole and narrow domains for the Stanford dragon model. (a–c) show offset surfaces reconstructed using the whole domain, using the narrow domain, and a comparison of the two results in the cross view.



**Fig. 12.** Comparison between the whole and narrow domains for the Stanford Asian dragon model. (a–c) show offset surfaces reconstructed using the whole domain, using the narrow domain, and a comparison of the two results in the cross view.



Fig. 13. Volume reconstructions for the Stanford Thai Statue and Stanford Lucy. From left to right, these are the initial data set and offset reconstruction with different views.

results suggest that our proposed method can successfully reconstruct the uniform and smooth offset surface.

Fig. 7(a) and (b) shows the narrow volume generated by our algorithm with the offset distances  $l = 4h$  and  $l = 12h$ , respectively. From left to right, these are the outer surface, the inner surface, and the offset surface in the cross view, respectively. Here, the solid line and the circle correspond to the offset surface and input data, respectively. We put the offset surface and input data together, to show that the uniform cutting depth can be achieved with different offset surface distances.

To show the effect of  $\xi$ , we reconstruct the teacup surface using different values of  $\xi_1$ ,  $\xi_2$  and  $\xi_4$ . In this test, an offset distance of  $l = 10h$  is employed. Fig. 8(a–c) shows the final reconstructions with  $\xi_1$ ,  $\xi_2$ , and  $\xi_4$ , respectively. In order to make comparisons with them in the cross view, we put them together as shown in Fig. 8(d) and (e). As can be seen, when  $\xi$  becomes too small, there will be holes in the reconstructed surface (see the spout of teacup in

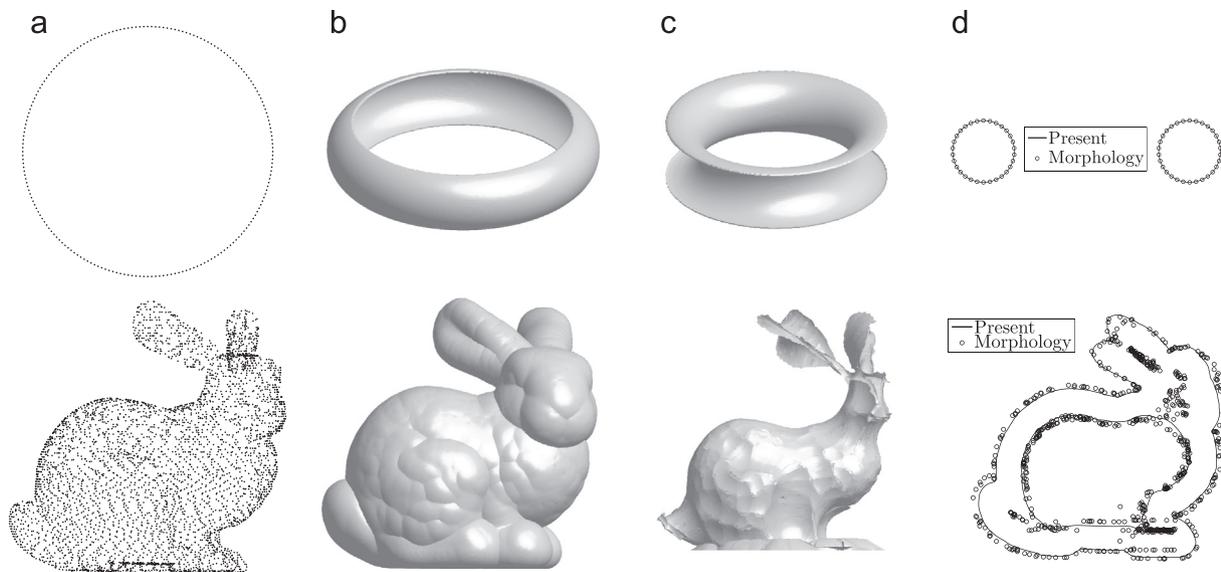
Fig. 8(a)). On the other hand, if  $\xi$  is too large, then the spout of the teacup becomes too fat and is closed as shown in Fig. 8(d) and (e).

Fig. 9 shows our reconstruction results for the happy buddha model with different input data points and values of  $\epsilon$ . The size of the input data in the bottom row is smaller than in the top row by a factor 6. In Fig. 9, the first, second, third, and fourth columns are the plots of the input data, initial reconstruction, final reconstruction with  $\epsilon_3$ , and final reconstruction with  $\epsilon_6$ , respectively. If the size of input data is large, using a small  $\epsilon$  is a good choice for capturing finer detail in the reconstructed volume (refer to the first row of Fig. 9(c) and (d)). On the other hand, if the number of input data points is small, then the initial reconstruction becomes wrinkled. When  $\epsilon$  is large, the noises can be removed effectively, as shown in Fig. 9(d).

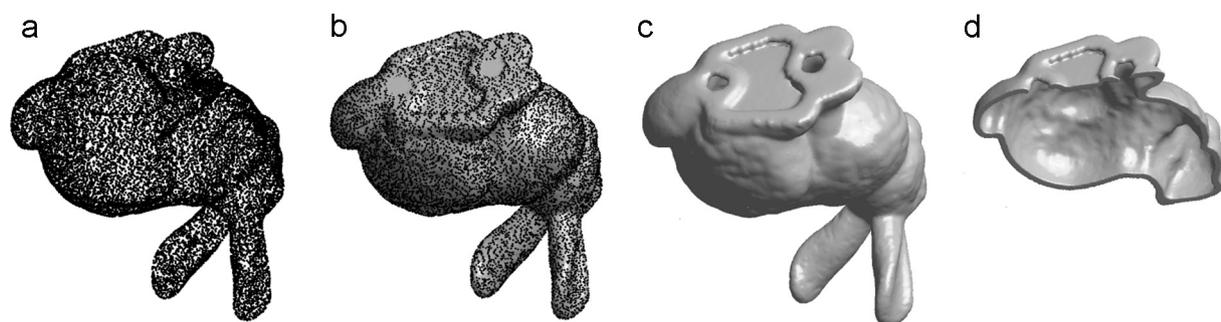
Fig. 10 presents our reconstruction results for the armadillo model with coarse and fine grids. We observe that as the mesh grid is increased, the scales of the armadillo become more sharply pronounced.

**Table 2**  
List of data information, iterations, and CPU times (second). Here, 'Pt size' represents the number of data points. 'CPU(ini)' is the time taken for the initial reconstruction (the simplification of the input data, computation of the distance function, and location of the connections of grid points). 'CPU(pro)' is the time taken to process the offset surface reconstruction. 'W' and 'N' represent the cases using the whole and narrow domains, respectively. 'N/W' is the ratio between 'N' and 'W'.

Case	Pt size	Grid size		Iteration		CPU (ini)		CPU (pro)	
		W	N/W	W	N	W	N	W	N
Fig. 5	2964	100 × 100 × 100	0.363	7	7	0.17	0.24	3.93	1.60
Fig. 7(a)	17,974	254 × 251 × 201	0.162	8	8	1.20	1.36	54.83	11.21
Fig. 7(b)	17,974	270 × 267 × 216	0.178	6	6	7.41	9.48	43.66	8.67
Fig. 8(a)	26,103	276 × 182 × 146	0.127	7	7	1.29	1.59	18.00	2.92
Fig. 8(b)	26,103	278 × 184 × 148	0.179	6	6	1.78	2.26	21.28	5.44
Fig. 8(c)	26,103	282 × 188 × 152	0.279	7	7	3.15	4.12	33.87	11.62
Fig. 9(c)	541,449	194 × 453 × 195	0.214	4	5	13.78	18.29	25.87	9.07
Fig. 9(d)	541,449	194 × 453 × 195	0.221	9	10	13.78	18.30	73.60	21.34
Fig. 9(g)	90,241	195 × 454 × 195	0.219	8	8	2.73	3.49	48.66	14.96
Fig. 9(h)	90,241	195 × 454 × 195	0.219	12	12	2.73	3.48	91.92	26.69
Fig. 10(b)	43,244	107 × 123 × 98	0.205	5	6	1.08	1.42	2.23	0.78
Fig. 10(c)	43,244	303 × 353 × 278	0.209	5	5	21.31	27.64	111.92	27.13
Fig. 11(a)	218,823	462 × 330 × 215	0.158	4	4	6.82	8.54	44.55	10.17
Fig. 11(b)	451,182	460 × 214 × 311	0.087	5	5	11.78	15.47	49.65	6.49
Fig. 13(a)	1,249,999	501 × 833 × 434	0.068	5	6	35.09	41.91	287.28	36.22



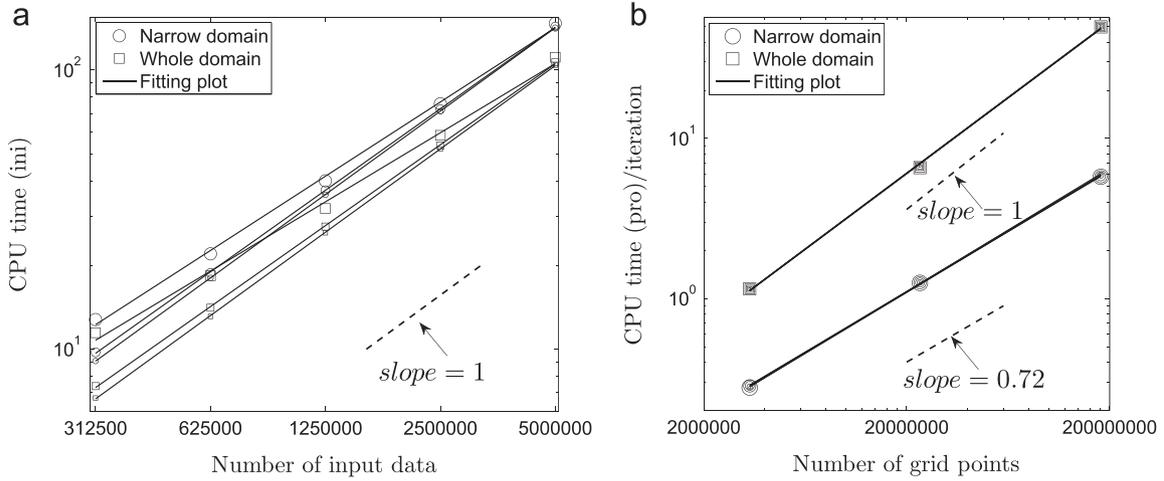
**Fig. 14.** Offset surfaces generated by morphological operations. (a) shows the input data points. (b–d) show the outer surface, the inner surface obtained by morphological operations, and a comparison between the offset surface obtained by morphological operations and our proposed method in the cross view, respectively. Note that the input data in the top is  $S = \{(0.5 + 0.4 \cos(m\pi/1000), 0.5 + 0.4 \sin(m\pi/1000), 0.5) | m = 1, \dots, 2000\}$ .



**Fig. 15.** Following of a bunny model. (a) shows the input data points, where two holes are added. For a better visualization, we show the input surface together with data points in (b). (c) and (d) show the reconstructed surface in whole view and cut view, respectively.

In this experiment, we compare the results on the whole and narrow domains to demonstrate the efficiency of our proposed method. The reconstructed offset surfaces for the Stanford dragon models obtained with the whole and narrow domains are shown

in Figs. 11 and 12. Comparisons between the two domains in the cross view are shown in the third rows of Figs. 11 and 12. As can be observed, the agreement between the results computed on the whole and narrow domains is good.



**Fig. 16.** (a) Logarithm plots of CPU times for the initial reconstruction versus the number of input data points. From small sizes of the circle symbol to large ones, these represent the results obtained using  $501 \times 833 \times 434$ ,  $253 \times 416 \times 221$  and  $134 \times 215 \times 117$  grid points, respectively. Square symbols are defined in a similar manner. (b) Logarithm plots of the CPU time per iteration versus the number of grid points. Circle symbols from small to large represent the results obtained using 5, 2.5, 1.25, 0.625, and 0.3125 million input points. Square symbols are similarly defined.

To compare the quality of the narrow band method against the global method, we perform a test using a quality measure. We define the error of a grid as the discrete  $l_2$ -norm of the difference between the numerical solution obtained by the narrow band method ( $\phi$ ) and the global method ( $\bar{\phi}$ ) as  $\sqrt{\sum_{s=1}^L (\phi_s - \bar{\phi}_{ijk})^2} / L$ . Note that for some  $s$  and  $\phi_s$  in narrow band domain, there exists some  $ijk$  and  $\bar{\phi}_{ijk}$  in the global domain corresponding to  $\phi_s$ . The errors obtained using these definitions are  $2.955E-4$  and  $2.376E-4$  for the two cases above, respectively. We can observe that the agreement between them is strong.

Fig. 13(a) and (b) shows the reconstruction of the Stanford Thai Statue and Stanford Lucy, respectively. From left to right, these are the input data set and final reconstruction with different views. We observe that our proposed method can handle the complex morphology, and the reconstructed offset surface is smooth.

Next, Table 2 presents the information on the number of data points, the iteration numbers, and the CPU times. The CPU times (seconds) of our calculations, which are performed using C, are measured on 3.4 GHz with 8 GB of RAM. For each case, we perform the computations on the whole and narrow domains. Here, ‘Pt size’ represents the number of data points. ‘CPU(ini)’ is the time taken for the initial reconstruction (the simplification of the input data, computation of the distance function, and location of the connections of grid points). ‘CPU(pro)’ is the time taken to process the offset surface reconstruction. ‘W’ and ‘N’ denote the cases using the whole and narrow domains, respectively. ‘N/W’ is the ratio between ‘N’ and ‘W’. As can be seen from Table 2, our proposed method in both the whole and narrow domains achieves fast convergence after a few iterations, as expected from the unconditionally stable discrete scheme. Furthermore, observing the grid sizes used for the whole and narrow domains, we can confirm that our method can significantly reduce the storage memory. As the grid size in the whole domain is increased, the difference between the two cases becomes greater (see the last three lines in Table 2). In comparing the elapsed time, we see that the new method is much faster than the preceding whole domain method.

Our proposed method can be generalized further to morphology operations such as erosion and dilation. The erosion of data points  $S$  by  $B_l$ , which denotes a ball with the center located on the origin of the data points  $S$  with a radius of  $l$ , can be understood as the locus of points reached by the center of the data points  $S$  when the data points within  $B_l$  move inside  $S$  [35,36]. We denote the surface obtained in this way as  $S^-$ . Thus, in the interior of the offset surface,  $S^-$  can be defined as  $S^- = \{\mathbf{X}_m - l\mathbf{n}_m | m = 1, \dots, M\}$ . Here,  $\mathbf{n}$  denotes

the outward unit normal vector to the solid. The dilation of data points  $S$  can be understood as the locus of the points covered by  $B_l$ . Thus, inside the offset surface,  $S^+$  can be defined as  $S^+ = \{\mathbf{X}_m + l\mathbf{n}_m | m = 1, \dots, M\}$ . Next, we perform two numerical tests. Fig. 14, from left to right, presents the input data, outer surface, inner surface obtained by morphological operations, and a comparison between the offset surfaces obtained by morphological operations and our proposed method in the cross view. We can confirm from this that our proposed method can effectively be further generalized to morphological erosion and dilation. In addition, with morphological operators only, the generated surfaces are topologically faithful and self-intersecting (see Fig. 14(c)). However, by coupling with morphological operators and other representations, uniform offsetting can be obtained. For example, Chen and Wang [16] proposed a simple and efficient polygonal model, based on directly computing the offset boundary using morphological operations, converting the boundary into structurally sampled points, and accordingly filtering the sampling points to reconstruct the offset contour.

In 3D printing, occasionally, to save large amounts of material, a suitable method is to design a hollow model [37]. A straightforward method is to uniformly hollow out 3D models by opening some holes and leaving out the material from the inside of the model. Our approach can be further generalized to the hollowing of models. Fig. 15 shows the reconstructed volume for a bunny. Fig. 15(a) shows the input data points, where two holes are added. To achieve a better visualization, we show the input surface and data points together in Fig. 15(b). As can be seen in Fig. 15(c) and (d), our proposed method can be further generalized successfully to the hollowing of a model in 3D printing.

Finally, we will show how the proposed method behaves in terms of the number of input surface points and the number of grid points. The narrow domain and whole domain are considered with the Stanford Thai Statue, which has 5 million points [38]. This test case is also down-sampled with factors of two, in order to obtain point sets with 5, 2.5, 1.25, 0.625, and 0.3125 million points. Furthermore, the computational times are given with three different numbers of grid points:  $501 \times 833 \times 434$ ,  $253 \times 416 \times 221$ , and  $134 \times 215 \times 117$ .

Fig. 16(a) shows logarithm plots of the CPU time for the initial reconstruction versus the number of input data points. Note that from the small size of the circle symbol to the large one, these are the results obtained using  $501 \times 833 \times 434$ ,  $253 \times 416 \times 221$ , and  $134 \times 215 \times 117$  grid points. Square symbols are defined in a similar manner. We find that the CPU time using our proposed

method is slightly larger than for the global method, as we should find the connection of grid points in the narrow domain. From Fig. 16(a), we also observe that the cost of the initial reconstruction is linear with respect to the input data points, as we expected from the computing algorithm for the distance function. Logarithm plots of the CPU time per iteration versus the number of grid points are presented in Fig. 16(b), and these confirm that our proposed method is more efficient than the global method. Here, the circle symbols from small to large denote the results obtained using 5, 2.5, 1.25, 0.625, and 0.3125 million input points. Square symbols are similarly defined.

## 5. Conclusion

In this study, we proposed a novel fast and efficient numerical method for volume reconstruction from unorganized points in three dimensions. The present study constitutes an extension of previous research [12]. The main idea in the present work was to perform the computations only on a narrow band around a given surface data set. In this way, we were able to significantly reduce the storage memory and CPU time. The proposed numerical method, based on operator splitting techniques, can use a large time step size and exhibits unconditional stability. Fast numerical techniques for the Allen–Cahn equation also allow for larger data sets to be processed, greatly aiding the speed of computation. Various numerical experiments were presented to demonstrate the robustness and efficiency of the proposed method. The quality of the reconstructed surfaces with our method certainly is dependent on the number of input data points and the mesh grids used. When coarse grids are used, the number of input data points can be decreased, in order to reduce the CPU time. However, the quality of the reconstructed surfaces is greatly limited by the mesh grid. To obtain a high-quality surface, both a large number of input data and sufficient grid points are required. The proposed numerical method, based on operator splitting techniques, is fast, as expected from the unconditionally stable discrete schemes. Its optimal complexity is only related to the number of grids in the narrow domain. However, the cost of computing a distance function in the current method is high, which has an optimal complexity of  $O(M(m+l/h)^3)$  operations for  $M$  input data points. Zhao [39] presented a simple and fast sweeping method for computing distance functions, having an optimal complexity of  $O(N_x N_y N_z)$  for  $N_x N_y N_z$  grid points. Ye et al. [3] used a fast sweeping method on the entire domain and obtained the narrow band by thresholding the unsigned distance function at some value. But in our method, the narrow domain can be obtained before computing the unsigned distance function. And we do not need to perform the computation on the whole domain, which significantly reduces the storage memory. In future work, to speed up the initial computation, we will investigate a fast numerical method for computing the distance function in the narrow domain.

## Conflict of interest

None declared.

## Acknowledgments

Y.B. Li was supported by the Fundamental Research Funds for the Central Universities, China (No. XJJ2015068) and supported by China Postdoctoral Science Foundation (No. 2015M572541). The corresponding author (J.S. Kim) was supported by the National

Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2014R1A2A2A01003683). The authors are grateful to the reviewers whose valuable suggestions and comments significantly improved the quality of this paper.

## References

- [1] N. Amenta, M. Bern, M. Kamvysselis, A new Voronoi-based surface reconstruction algorithm, in: Proceedings of SIGGRAPH 1998, pp. 415–421.
- [2] S. Liu, C.C.L. Wang, Orienting unorganized points for surface reconstruction, *Comput. Graph.* 34 (2010) 209–218.
- [3] J. Ye, I. Yanovsky, B. Dong, R. Gandini, A. Brandt, S. Osher, Multigrid narrow band surface reconstruction via level set functions, *Advances in Visual Computing* (2012) 61–70.
- [4] L. Jian, F. Park, H. Zhao, Robust and efficient implicit surface reconstruction for point clouds based on convexified image segmentation, *J. Sci. Comput.* 54 (2013) 577–602.
- [5] J.D. Boissonnat, Geometric structures for three-dimensional shape representation, *ACM Trans. Graph.* 3 (1984) 266–286.
- [6] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle, Surface reconstruction from unorganized points, *Comput. Graph.* 26 (1992) 71–78.
- [7] H.K. Zhao, S. Osher, R. Fedkiw, Fast surface reconstruction using the level set method, in: Proceedings of IEEE Workshop on Variational and Level Set Methods in Computer Vision, 2001.
- [8] H.K. Zhao, S. Osher, B. Merriman, M. Kang, Implicit, non-parametric shape reconstruction from unorganized points using a variational level set method, *Comp. Vis. Image Underst.* 80 (2000) 295–314.
- [9] S. Osher, R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Springer-Verlag, New York, NY, 2003.
- [10] S.M. Allen, J.W. Cahn, A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening, *Acta Metall.* 27 (1979) 1085–1095.
- [11] 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.
- [12] Y. Li, D. Lee, C. Lee, J. Lee, S. Lee, J. Kim, S. Ahn, J. Kim, Surface embedding narrow volume reconstruction from unorganized points, *Comput. Vis. Image Underst.* 121 (2014) 100–107.
- [13] S. Liu, C.C.L. Wang, Duplex fitting of zero-level and offset surfaces, *Comput. Aided Des.* 41 (2009) 268–281.
- [14] S. Liu, C.C.L. Wang, Fast intersection-free offset surface generation from freeform models with triangular meshes, *IEEE Trans. Autom. Sci. Eng.* 8 (2011) 347–360.
- [15] C.C.L. Wang, Y. Chen, Thickening freeform surfaces for solid fabrication, *Rapid Prototyp.* J. 19 (2013) 395–406.
- [16] Y. Chen, C.C.L. Wang, Uniform offsetting of polygonal model based on layered depth-normal images, *Comput. Aided Des.* 43 (2011) 31–46.
- [17] J.M. Lien, Covering Minkowski sum boundary using points with applications, *Comput. Aided Geom. Des.* 25 (2008) 652–666.
- [18] G. Varadhan, D. Manocha, Accurate Minkowski sum approximation of polyhedral models, in: Proceedings of 12th Pacific Conference on Computer Graphics and Applications; IEEE Comput. Sci. (2004) 392–401.
- [19] B. Curless, M. Levoy, Better optical triangulation through spacetime analysis, *Proc. IEEE* (1995) 987–994.
- [20] B. Curless, M. Levoy, A volumetric method for building complex models from range images, in: Proceedings of SIGGRAPH, vol. 96, 1996, pp. 303–312.
- [21] D. Pavić, L. Kobbelt, High-resolution volumetric computation of offset surfaces with feature preservation, *Comput. Graph. Forum* 27 (2) (2008) 165–174.
- [22] G. Varadhan, S. Krishnan, Y.J. Kim, D. Manocha, Feature-sensitive subdivision and isosurface reconstruction, In: Proceedings of Visualization vol. 03, Seattle, 2003, pp. 99–106.
- [23] G. Varadhan, S. Krishnan, D. Manocha, Topology preserving surface extraction using adaptive subdivision, in: Proceedings of the Symposium Geometry Processing, 2004, pp. 235–244.
- [24] D.E. Breen, S. Mauch, Generating shaded offset surfaces with distance, closest-point and color volumes, in: Proceedings of the International Workshop on Volume Graphics, 1999, pp. 307–320.
- [25] D.E. Breen, S. Mauch AUCH, R.T. Whitaker, 3d scan conversion of csg models into distance volumes, in: Proceedings of the IEEE Symposium on Volume Visualization, New York, 1998, pp. 7–14.
- [26] R.T. Farouki, The approximation of non-degenerate offset surfaces, *Comput. Aided Geometric Des.* 3 (1986) 15–43.
- [27] S.J. Kim, M.Y. Yang, Triangular mesh offset for generalized cutter, *Comput. Aided Des.* 37 (2005) 999–1014.
- [28] X. Qu, B. Stycker, A 3d surface offset method for stl-format models, *Rapid Prototyp.* J. 9 (2003) 133–141.
- [29] C.C.L. Wang, D. Manocha, GPU-based offset surface computation using point samples, *Comput. Aided Des.* 45 (2013) 321–330.
- [30] B. Yang, W.-K. Wong, C. Liu, P. Poignet, 3D soft-tissue tracking using spatial-color joint probability distribution and thin-plate spline model, *Pattern Recognit.* 47 (2014) 2962–2973.
- [31] J. Kim, Phase-field models for multi-component fluid flows, *Commun. Comput. Phys.* 12 (2012) 613–661.

- [32] Y. Li, J. Kim, An unconditionally stable hybrid method for image segmentation, *Appl. Numer. Math.* 82 (2014) 32–43.
- [33] Y. Li, J. Kim, Multiphase image segmentation using a phase-field model, *Comput. Math. Appl.* 62 (2011) 737–745.
- [34] Y. Li, J. Kim, A fast and accurate numerical method for medical image segmentation, *J. KSIAM* 14 (2010) 201–210.
- [35] X. Jiang, S. Lou, Paul.J. Scott, Morphological operations in surface metrology and dimensional metrology based on the alpha shape, *Meas. Sci. Technol.* 23 (2012) 015003.
- [36] M. Couprie, G. Bertrand, Topology preserving alternating sequential filter for smoothing two-dimensional and three-dimensional objects, *J. Electron. Imaging* 13 (2004) 720–730.
- [37] W. Wang, T.Y. Wang, Z. Yang, L. Liu, X. Tong, W. Tong, J. Deng, F. Chen, X. Liu, Cost-effective printing of 3D objects with Skin-Frame structures, *ACM Trans. Graph.* 32 (5) (2013) 1–10.
- [38] Available Online: (<http://graphics.stanford.edu/data/3Dscanrep>).
- [39] H.K. Zhao, A fast sweeping method for Eikonal equations, *Math Comput.* 74 (2005) 603–627.

**Yibao Li** received B.S. degree from the Department of Mathematics, Xi'an University of Technology, China in 2007. He also received the M.S. and Ph.D. degrees in Applied Mathematics from Korea University, Korea, in 2011 and 2013, respectively. Before he joined the School of Mathematics and Statistics, Xi'an Jiaotong University, China in 2014, he held a research position in Department of Computational Science and Engineering, Yonsei University, Korea. His research interests include image processing, phase field model, computational fluid dynamics, and scientific computing.

**Junseok Kim** received his Ph.D. in Applied Mathematics from the University of Minnesota, USA 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 computational fluid dynamics.