Surface Reconstruction from Discrete Indicator Functions

Fabien Evrard, Fabian Denner, and Berend van Wachem

Abstract—This paper introduces a procedure for the calculation of the vertex positions in Marching-Cubes-like surface reconstruction methods, when the surface to reconstruct is characterised by a discrete indicator function. Linear or higher order methods for the vertex interpolation problem require a smooth input function. Therefore, the interpolation methodology to convert a discontinuous indicator function into a triangulated surface is non-trivial. Analytical formulations for this specific vertex interpolation problem have been derived for the 2D case by Manson et al. [Eurographics (2011) 30, 2] and the straightforward application of their method to a 3D case gives satisfactory visual results. A rigorous extension to 3D, however, requires a least-squares problem to be solved for the discrete values of a symmetric neighbourhood. It thus relies on an extra layer of information, and comes at a significantly higher cost. This paper proposes a novel vertex interpolation method which yields second-order-accurate reconstructed surfaces in the general 3D case, without altering the locality of the method. The associated errors are analysed and comparisons are made with linear vertex interpolation and the analytical formulations of Manson et al. [Eurographics (2011) 30, 2].

Index Terms—Surface reconstruction, volume fractions, discrete indicator function, marching-cubes, vertex interpolation

1 Introduction

In computer graphics [1], [2] and computational physics [3], it is common to represent a closed subset Ω of the d-dimensional space with a discontinuous indicator function $\chi : \mathbb{R}^{\mathrm{d}} \rightarrow \mathbb{R}$ defined as

$$\chi(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x} \in \Omega, \\ 0 & \text{if } \boldsymbol{x} \notin \Omega. \end{cases} \tag{1}$$

If the domain encompassing Ω is discretised into an ensemble $\mathcal{M}$ of small cuboids (or cells), then to each cell K is associated the local fraction of volume occupied by $K \cap \Omega$. That is to say

$$\forall K \in \mathcal{M}, \qquad \gamma_K = \frac{|K \cap \Omega|}{|K|} = \frac{1}{|K|} \int_K \chi(\boldsymbol{x})\, d\boldsymbol{x}. \tag{2}$$

This discrete field γ, illustrated in Fig. 1, is referred to as the *volume fraction* or the *discrete indicator function* (DIF) field. Both denominations are used in this paper.

DIF formulations can be found in various disciplines including computer graphics and image processing (e.g., font rasterization or curve/surface reconstruction), medical and geophysical computed tomography, or interfacial flow modelling. In order to reconstruct an approximation of a free-boundary $\partial\Omega$ from its associated discrete data field, classical polygonisation methods such as the Marching-Cubes method [6] or Bloomenthal's polygoniser [7] require to solve local *vertex interpolation* problems, i.e., compute the position of the intersections between $\partial\Omega$ and the edges of the discrete mesh.

• *The authors are with the Department of Mechanical Engineering, Imperial College London, Exhibition Road, London SW7 2AZ, United Kingdom. E-mail: fa.evrard@gmail.com, {f.denner09, b.van-wachem}@imperial.ac.uk.*

Manuscript received 31 Jan. 2017; revised 16 Feb. 2018; accepted 18 Feb. 2018. Date of publication 20 Mar. 2018; date of current version 29 Jan. 2019. (Corresponding author: Fabien Evrard.) Recommended for acceptance by V. Pascucci. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TVCG.2018.2809751

Most vertex interpolation techniques available in the literature are based on linear or higher order polynomial interpolation [8], and rely on the assumption that the sampled function is smooth along the edges of the discrete mesh. The discontinuous nature of an indicator function makes it badly suited for such methods. Fig. 2 shows a comparison between different vertex interpolation methods and post-reconstruction treatments applied to the DIF surface reconstruction problem. High-frequency error patterns are clearly visible in Fig. 2b where linear interpolation is used directly on the DIF field. Beyond the visual aspect of the reconstruction, some applications (e.g., in computational physics) require the position of the triangulated vertices to be very accurately computed. In the modelling of interfacial flows, for instance, an accurate estimation of the surface curvature is essential in order to prevent the generation of parasitic flow currents [9], [10]. If computed from the reconstructed interface, its accuracy directly depends on the quality of the vertex interpolation process.

This paper proposes a method to accurately position the vertices of a surface reconstructed from DIF values in a general 3D case. It relies on the computation of local volume-fraction-compliant piecewise-linear surface approximations, based on which the positions of the reconstructed surface vertices are iteratively corrected. The proposed method conserves the locality of the interpolation procedure and does not require to solve for a least-squares problem.

2 Related Work

The DIF surface reconstruction problem can be addressed in various ways. A trivial and frequently used solution consists in mollifying the DIF field (e.g., by means of convolution) before reconstructing the surface using a classical linear or higher order vertex interpolation method. This greatly reduces the high-frequency errors visible in Fig. 2b, and provides the reconstructed surface with a better visual aspect. However, it leads to changes in its shape—for instance in regions of high curvature—which may be unacceptable (see Fig. 2c where, prior to the Marching-Cubes reconstruction, the DIF field has been mollified via convolution with a radial basis function which is compact on a sphere of radius $r = 2\Delta x$; Δx being the mesh spacing).

Applying a mesh smoothing operator to the post-reconstruction triangulated surface is also a way to improve the visual aspect of the free-boundary approximation, as shown in Fig. 2d. The iterative use of a discrete smoothing operator defined as

$$\mathcal{S}(\boldsymbol{p}) = \boldsymbol{p} + \frac{\sum_{i \in V} (\boldsymbol{p}_i - \boldsymbol{p})}{\#V}, \tag{3}$$

for example, where V is the local neighbourhood of vertices around the vertex $\boldsymbol{p}$, allows for a reduction of the high-frequency errors at low cost but tends to significantly alter high-curvature features of the triangulated surface. Enhanced mesh smoothing techniques based on the successive use of Laplacian smoothing [4] or on mean curvature flow smoothing [11] can filter small errors and conserve the volume and small scale structures of the surface more accurately. Such methods come at a higher cost and still exhibit errors which can be unsuitable for applications that require a high reconstruction accuracy.

Another way to obtain smooth surfaces from volume fractions is to solve a global constrained optimisation problem for the surface smoothness, as done in methods which extract smooth surfaces from binary data [12], [13], [14]. Starting from an initial surface guess, such methods minimise the local curvature under the constraint that the surface must remain consistent with its associated data field. These techniques mainly focus on the smoothness of the reconstructed surface, and they exhibit a relatively large

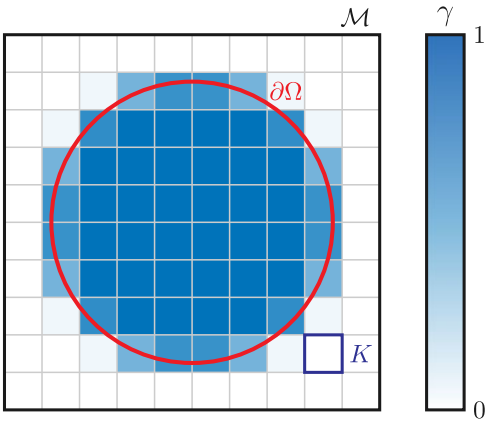
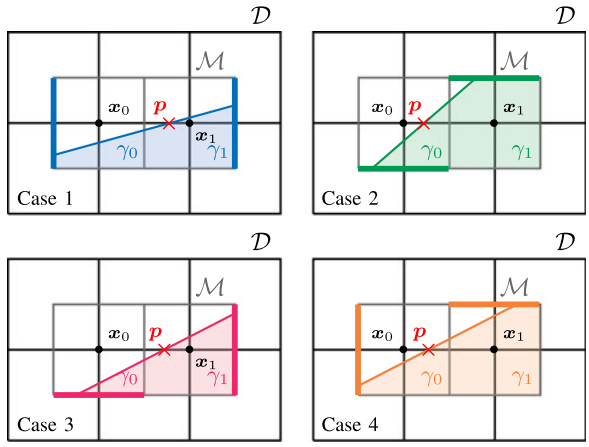


Fig. 1. Example of the *volume fraction* or *discrete indicator function* field γ associated with a circular free-boundary $\partial\Omega$. The domain under consideration is discretised into an ensemble $\mathcal{M}$ of squares K.

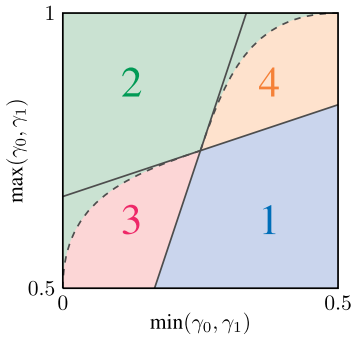
computational cost. Alternatively, Prilepov et al. [15] use a gradient field approximation based on trilinearly blended Coons-patches to generate a smooth volume fraction function, which results in smooth reconstructed surfaces with good volume-conservation properties. Finally, Ledergerber et al. [16] rely on the well-known moving least-squares formalism to reconstruct smooth isosurfaces and compute differential quantities accurately.

In fact, it is possible to avoid using any form of smoothing and still obtain both smooth and accurate results by considering the nature of the information provided by the DIF data, i.e., local occupancies. In order to do this, it is relevant to work with the dual of $\mathcal{M}$, named $\mathcal{D}$, whose vertices correspond to the cell centers of $\mathcal{M}$, as shown in Fig. 3a. By using $\mathcal{D}$ instead of $\mathcal{M}$, a Marching-Cubes or similar reconstruction algorithm can be applied with direct vertex access to the DIF values [17]. With this direct access to the volume fractions, the local vertex interpolation problem (i.e., finding the position of $\boldsymbol{p}$ in Fig. 3a) can be formulated as a function of the two volume fractions γ_0 and γ_1 available at both ends of each edge of $\mathcal{D}$.

For a given edge of $\mathcal{D}$ in 2D, there exists a unique straight line that locally approximates the free-boundary $\partial\Omega$ while complying with the local occupancies γ_0 and γ_1. Manson et al. [5] showed that finding this line can only take the form of four distinct problems for each of which an analytical solution is available. These cases are depicted in Fig. 3a, and Fig. 3b shows which case relates to a given combination (γ_0, γ_1). In formal terms: considering an edge of $\mathcal{D}$ linking two cuboids of $\mathcal{M}$ with centers $\boldsymbol{x_0}$ and $\boldsymbol{x_1}$ and associated volume fractions



(a) The four reference cases encountered when trying to constrain a line with two volume fractions γ_0 and γ_1



(b) Case repartition based on the values of γ_0 and γ_1

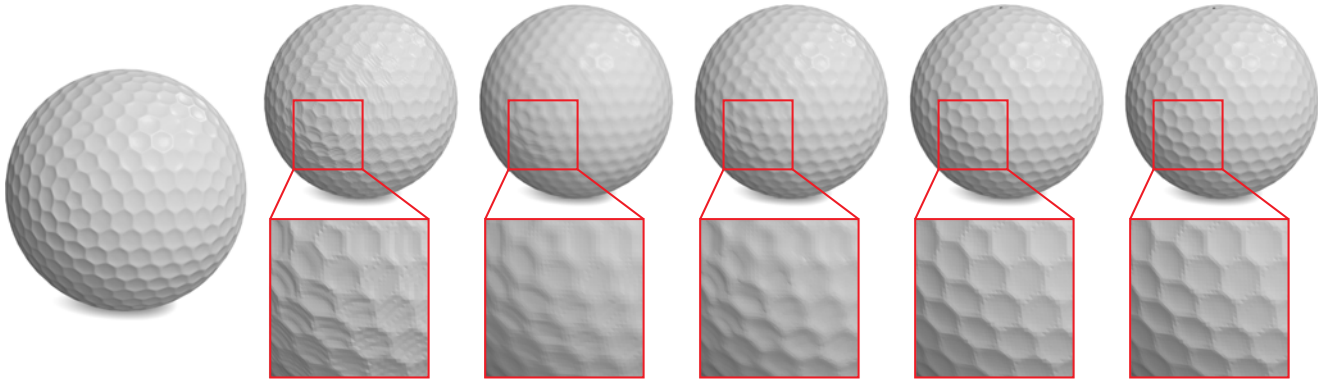
Fig. 3. A unique local linear approximation can be recovered from the volume fractions γ_0 and γ_1 available at both ends of an edge of $\mathcal{D}$.

γ_0 and γ_1, then if $\text{sign}(\gamma_0 - 0.5) = -\text{sign}(\gamma_1 - 0.5)$, the free-boundary intersects the edge, and the position of the intersection $\boldsymbol{p}$ writes

$$\boldsymbol{p} = \mu\, \boldsymbol{x_0} + (1 - \mu)\, \boldsymbol{x_1}. \tag{4}$$

Assuming $\gamma_0 < \gamma_1$ and with the notation $\bar{\gamma} = 1 - \gamma$, the parameter μ is given by

$$\text{Case 1}: \qquad \mu = \frac{0.5 - \gamma_0}{\gamma_1 - \gamma_0} \quad \text{(linear interpolation)} \tag{5a}$$



(a) Reference surface (b) Linear vertex interpolation on the sharp DIF (c) Linear vertex interpolation on the mollified DIF (d) Mesh smoothing on the triangulated surface using Taubin [4] (e) Vertex interpolation method of Manson et al. [5] (f) Proposed vertex interpolation method

Fig. 2. Reconstruction of a *Golf Ball* (obtained from the Princeton University Suggestive Contour Gallery) with Marching-Cubes using different vertex interpolation methods and/or post-reconstruction treatments: (a) Reference model from which the DIF is computed; (b) Reconstructed surface using a linear vertex interpolation method; (c) Reconstructed surface using a linear vertex interpolation method on a mollified DIF; (d) Reconstructed surface using linear vertex interpolation and after 15 post-reconstruction Taubin [4] mesh smoothing iterations; (e) Reconstructed surface using the vertex interpolation method of Manson et al. [5]; (f) Reconstructed surface using the vertex interpolation method proposed in this paper.

$$\text{Case 2}: \quad \mu = \frac{3}{2} - \gamma_0 - \gamma_1 \qquad \text{(column height)} \tag{5b}$$

$$\text{Case 3}: \quad \mu = 1 - \frac{2\gamma_1 - 1}{8\gamma_0 + 4\gamma_1 - 8\sqrt{\gamma_0(\gamma_0 + \gamma_1)}} \tag{5c}$$

$$\text{Case 4}: \quad \mu = 1 - \frac{2\bar{\gamma}_1 - 1}{8\bar{\gamma}_0 + 4\bar{\gamma}_1 - 8\sqrt{\bar{\gamma}_0(\bar{\gamma}_0 + \bar{\gamma}_1)}}. \tag{5d}$$

From a computational point of view, a sequence of simple tests allows to determine which case is associated with a given couple (γ_0, γ_1). If $\gamma_0 > \gamma_1$, then $(\gamma_0, \boldsymbol{x_0})$ and $(\gamma_1, \boldsymbol{x_1})$ are swapped and the vertex interpolation follows the same process.

The same reasoning cannot be applied to the general 3D case without losing the locality of the method: that is because two values of the DIF are not enough to constrain a plane—doing so would require to solve a minimisation problem for the errors in volume fractions of the neighbouring cells (in a way which is similar to what the so-called (E)LVIRA methods [18] do for the computation of surface normals from volume fractions). In practice, applying the Formula (5) 'as is' for the 3D case provides satisfactory visual results [5]. This, however, is equivalent to constraining the surface normal to have a zero-component in one of the main grid directions perpendicular to the considered edge, which can lead to relatively large errors when, in reality, the normal component in that direction is large.

3 Proposed Method

The present section proposes an iterative vertex positioning procedure which allows to consider the local normal orientation of the surface during the vertex interpolation process. This novel approach couples piecewise-linear volume-fraction-compliant surface approximations, widely used in the field of interfacial flow modelling, with the Marching-Cubes surface reconstruction problem.

3.1 Piecewise-Linear Surface Approximations

In the proposed approach, prior to iteratively positioning the vertices, a linear approximation of the surface that complies with the local volume fraction is computed in each interfacial cell[1] of $\mathcal{M}$, using the analytical solutions of Scardovelli and Zaleski [19]. This requires the surface normal orientation to be estimated in the interfacial cells.

3.1.1 Estimation of the Surface Normals from the DIF

The surface normals are estimated directly from the volume fractions. Making use of the Cartesian nature of the grid $\mathcal{M}$, a height-function (HF) method [20] is employed to do so. The HF method applies central-differences on the 'heights of material' in a local $3 \times 3 \times 9$ stencil of cells. This stencil is oriented based on an estimation of the local dominant normal direction, and the heights of material $H_{i,j}$ are obtained by summing the volume fractions in each column, following:

$$H_{i,j} = \sum_{k=-4}^{4} \gamma_{i,j,k}\, \Delta x\ , \quad (i,j) \in \{-1,0,1\}^2, \tag{6}$$

where $\gamma_{i,j,k}$ is the volume fraction in the (i,j,k)th cell of the HF stencil. Assuming, for instance, that the dominant normal direction is along the z-direction, the surface normal for the stencil under consideration is given by

$$\boldsymbol{n} = \frac{1}{\sqrt{H_x^2 + H_y^2 + 1}} \begin{bmatrix} -H_x \\ -H_y \\ 1 \end{bmatrix}, \tag{7}$$

1. An 'interfacial' cell has a volume fraction that satisfies $0 < \gamma < 1$.

with

$$H_x = \frac{H_{1,0} - H_{-1,0}}{2\Delta x}, \tag{8}$$

$$H_y = \frac{H_{0,1} - H_{0,-1}}{2\Delta x}. \tag{9}$$

The formulas for x-dominant or y-dominant normal directions are straightforward to obtain. The HF method is known to provide erroneous results when heights are not 'consistent', that is when one of the following conditions is not satisfied:

- At least one cell in each column is full (i.e., there is at least one cell for which $\gamma = 1$)
- At least one cell in each column is empty (i.e., there is at least one cell for which $\gamma = 0$)
- Each column crosses the interface exactly once (i.e., all cells with $0 < \gamma < 1$ are adjacent to each other)

For this reason, the HF method is here coupled with a mixed-Youngs-centred (MYC) method [21]. The MYC method estimates the local surface normal on a $3 \times 3 \times 3$ stencil, using central-differences on the mollified DIF field or on the local heights of material, regardless of their consistency. It provides the initial normal guess used to determine the dominant normal direction in the HF method, as well as an alternative to it when consistent heights are not found on the $3 \times 3 \times 9$ stencil.

3.1.2 Computation of the Piecewise-Linear Approximations

Once the surface normals $\boldsymbol{n}$ have been evaluated, a plane

$$\boldsymbol{n} \cdot \boldsymbol{x} = \alpha, \tag{10}$$

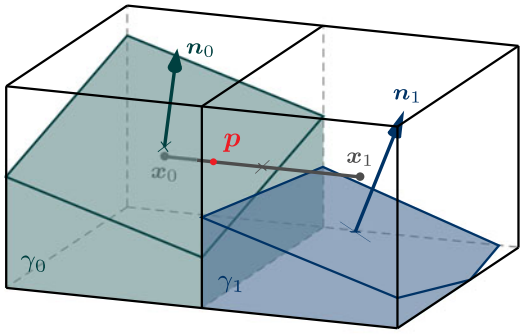
is defined in each interfacial cell. Without loss of generality and using simple coordinate transformations, the cell can be oriented such that $0 \le n_1 \le n_2 \le n_3$. The local volume fraction γ can then be expressed as a function of α following the relation [22]

$$\begin{aligned} \gamma = \frac{\alpha^3}{6|K|n_1 n_2 n_3} \Bigg\{ 1 - \sum_{i=1}^{3} H(\alpha - n_i c_i)(1 - n_i c_i/\alpha)^3 \\ + \sum_{i=1}^{3} H(\alpha - \alpha_{\max} + n_i c_i)(1 - \alpha_{\max}/\alpha + n_i c_i/\alpha) \Bigg\}, \end{aligned} \tag{11}$$

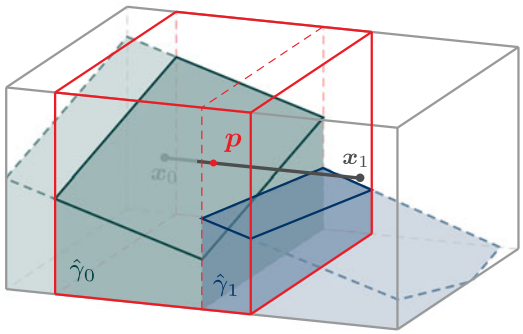
where $\boldsymbol{c} = (c_1, c_2, c_3)^T$ contains the dimensions of the cuboids of $\mathcal{M}$, the parameter $\alpha_{\max}$ corresponds to the smallest α for which $\gamma = 1$ (hence $\alpha_{\max} = \boldsymbol{n} \cdot \boldsymbol{c}$), and H is the Heaviside function. The coefficient α which complies with a given local volume fraction γ is calculated using the analytical formulations of Scardovelli and Zaleski [19] for the inverse problem $\alpha = f(\boldsymbol{n}, \gamma)$. An example of piecewise-linear surface approximations obtained from given normals and volume fractions is shown in Fig. 4a.

3.2 Iterative Vertex Positioning

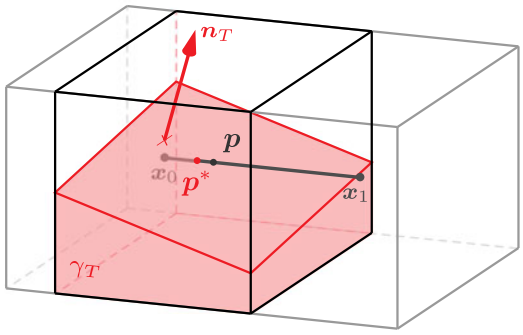
A given vertex $\boldsymbol{p}$ of the triangulated surface belongs, by construction, to an edge $[\boldsymbol{x}_0, \boldsymbol{x}_1]$ of the dual grid $\mathcal{D}$, with $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ the centres of two cells K_0 and K_1 of $\mathcal{M}$ (as illustrated in Fig. 4a). During the Marching-Cubes surface reconstruction process, the vertex $\boldsymbol{p}$ is positioned on the edge (for instance: it is arbitrarily placed at the middle of the edge or, alternatively, is positioned using linear interpolation or the vertex interpolation method of Manson et al. [5]). As part of the method proposed in this paper, a temporary cuboid T with similar dimensions as K_0 and K_1 is then centred around the vertex $\boldsymbol{p}$, as shown in Fig. 4b. The piecewise-linear approximations in K_0 and K_1 are used to compute the volume fractions $\hat{\gamma}_0$ and $\hat{\gamma}_1$ associated with $K_0 \cap T$ and $K_1 \cap T$, from which the volume fraction associated with T, named γ_T, follows as



(a) Piecewise-linear volume-fraction-compliant approximations in the cells K_0 and K_1



(b) Fractions of volume $\hat{\gamma}_0$ and $\hat{\gamma}_1$ contained in the cuboid centred around $\boldsymbol{p}$



(c) Piecewise-linear approximation in the temporary cell and vertex $\boldsymbol{p}^*$ after projection

Fig. 4. Iterative vertex positioning procedure: (a) Piecewise-linear approximations of the surface are computed in the two cuboids of $\mathcal{M}$ associated with a given edge of $\mathcal{D}$; (b) The fractions of volume $\hat{\gamma}_{(0,1)}$ in a temporary cuboid centred around the vertex $\boldsymbol{p}$ are calculated; (c) Using the calculated volume fraction γ_T and the vertex normal $\boldsymbol{n}_T$, a piecewise-linear approximation of the surface is computed in the temporary cuboid. The corrected vertex $\boldsymbol{p}^*$ is the projection of $\boldsymbol{p}$ on this plane.

$$\gamma_T = \frac{\hat{\gamma}_0 |K_0 \cap T| + \hat{\gamma}_1 |K_1 \cap T|}{|T|}. \tag{12}$$

The plane coefficient α_T in the temporary cell T is then calculated using the volume fraction γ_T and the normal $\boldsymbol{n}_T$ given by

$$\boldsymbol{n}_T = \frac{(1-\mu)\boldsymbol{n}_0 + \mu\,\boldsymbol{n}_1}{\|(1-\mu)\boldsymbol{n}_0 + \mu\,\boldsymbol{n}_1\|}. \tag{13}$$

The variable μ, associated with the position of p on the edge $[\boldsymbol{x}_0, \boldsymbol{x}_1]$, is introduced in Equation (4). Finally, the vertex $\boldsymbol{p}$ is moved on the edge $[\boldsymbol{x}_0, \boldsymbol{x}_1]$ until reaching the plane computed in the temporary cell T, or one of the extremities of the edge. Restricting the vertex to stay on the edge yields a vertex interpolation that remains consistent with the Marching-Cubes formalism. The resulting surface is thus guaranteed not to self-intersect. The corrected vertex is shown as $\boldsymbol{p}^*$ in Fig. 4c.

Other than the volume fractions γ_0 and γ_1 and the local normals $\boldsymbol{n}_0$ and $\boldsymbol{n}_1$, no information is required from the neighbouring cells in order to execute this procedure. This simplifies the implementation of the method and keeps the vertex interpolation process entirely local. Moreover, the proposed method does not require to solve a non-linear minimisation problem for each cell, but instead simply requires to execute a limited number of geometric operations, which keeps its computational cost low.

The process is repeated until the projection leaves μ unchanged (within the tolerance of a small positive constant $\delta \ll 1$). Algorithm 1 provides a summary of the proposed surface reconstruction method.

Algorithm 1. Proposed Surface Reconstruction Method

Parameters:
$k_{\max}$ ← Maximum number of iterations
δ ← Small positive constant
1. Marching-Cubes surface reconstruction on $\mathcal{D}$
(e.g., using linear vertex interpolation)
2. Piecewise-linear surface reconstruction on $\mathcal{M}$
for each cell $K \in \mathcal{M}$ **do**
  if $0 < \gamma_K < 1$ **then**
    $\boldsymbol{n}_K \leftarrow$ Compute MYC normal from DIF
    if heights are consistent **then**
      $\boldsymbol{n}_K \leftarrow$ Compute HF normal from DIF
    $\alpha_K \leftarrow$ Compute plane coeff. from $(\gamma_K, \boldsymbol{n}_K)$
3. Iterative vertex positioning
for each vertex $\boldsymbol{p}$ on an edge $[\boldsymbol{x}_0\boldsymbol{x}_1]$ of $\mathcal{D}$ **do**
  a. Initialisation
  $\mu^{(0)} \leftarrow -1$
  $\mu^{(1)} \leftarrow$ Compute μ from position of $\boldsymbol{p}$ on $[\boldsymbol{x}_0\boldsymbol{x}_1]$
  $k \leftarrow 0$
  b. Iterative vertex projection
  while $|\mu^{(k+1)} - \mu^{(k)}| > \delta$ and $k < k_{\max}$ **do**
    $T \leftarrow$ Temporary cuboïd centred around $\boldsymbol{p}$
    $\gamma_T \leftarrow$ Compute volume fraction in T
    $\boldsymbol{n}_T \leftarrow$ Compute normal from $(\mu^{(k+1)}, \boldsymbol{n}_0, \boldsymbol{n}_1)$
    $\alpha_T \leftarrow$ Compute plane coeff. from $(\gamma_T, \boldsymbol{n}_T)$
    $\boldsymbol{p} \leftarrow$ Project $\boldsymbol{p}$ on plane $\boldsymbol{n}_T \cdot \alpha_T = 0$
    $\mu^{(k)} \leftarrow \mu^{(k+1)}$
    $\mu^{(k+1)} \leftarrow$ Update μ from new position of $\boldsymbol{p}$
    $k \leftarrow k+1$

4 Error Analysis

The proposed approach is compared with linear vertex interpolation and the method of Manson et al. [5], on implicit surfaces and a 3D model. For the first test-case, a *Sphere* given by the expression

$$F(\boldsymbol{x}) = x^2 + y^2 + z^2 - R^2, \tag{14}$$

is considered. The second test-case considers a fifth-order polynomial, referred to as *Genus*, given by the expression

$$\begin{aligned} F(\boldsymbol{x}) = {} & 2y(y^2 - 3x^2)(1 - z^2) \\ & + (x^2 + y^2)^2 - (9z^2 - 1)(1 - z^2). \end{aligned} \tag{15}$$

Third, the so-called *Stanford Bunny* model issued from the Stanford 3D Model Repository is considered.

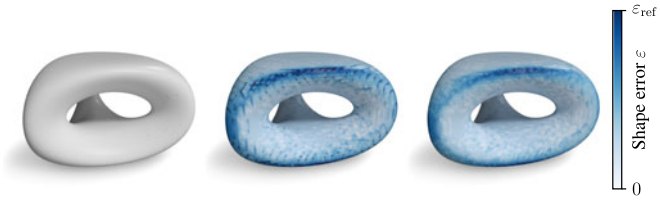
(a) Reference surface (b) Manson et al. (c) Proposed method

Method	$\|\varepsilon\|_2/R$	$\|\varepsilon\|_\infty/R$
Linear interpolation	3.06×10^{-4}	6.73×10^{-4}
Manson et al.	1.95×10^{-5} ($\times 0.064$)	7.99×10^{-5} ($\times 0.119$)
Proposed method	1.23×10^{-5} ($\times 0.040$)	1.80×10^{-5} ($\times 0.027$)

Fig. 5. Shape error ε for the reconstruction of a *Sphere* with $R/\Delta x = 64$ using different vertex interpolation methods. The reference error used for visualisation is $\varepsilon_{\text{ref}} = 2.5 \times 10^{-5}\, R$.

The DIF fields associated with the implicit surfaces are generated by integrating the volume fraction in each cell using the `Vofi` library [23], whereas the DIF field for the *Stanford Bunny* is generated by sampling, using the original 3D model. For the *Sphere*, the errors are computed by calculating the exact distance between the reconstructed vertices and their projection on the implicit surface. For the *Genus*, the errors are obtained by computing the distance between the reconstructed vertices and a high resolution reference model generated using a 512^3 grid. The errors for the *Stanford Bunny* are obtained by computing the distance between the reconstructed vertices and the original 3D model. Finally, the following parameters are chosen for the iterative vertex positioning procedure: $k_{\max} = 100$, and $\delta = 10^{-12}$. Three configurations are considered for the initial Marching-Cubes surface reconstruction (step 1 of Algorithm 1): *Mid-edge*—the vertices are initially positioned at the middle of their associated edge of $\mathcal{D}$, regardless of the local volume fraction values; *Linear*—the vertices are initially positioned using linear vertex interpolation; *Manson*—the vertices are initially positioned using the method of Manson et al. [5]. For all tests conducted in this paper, all three types of initial surface guesses lead to identical reconstructed surfaces after convergence of the proposed iterative vertex positioning procedure (step 3 of Algorithm 1). A fixed-point solution thus exists regardless of the initial surface guess obtained from the Marching-Cubes reconstruction.

Figs. 5, 6, and 7 show the shape errors for the reconstruction of the *Sphere*, *Genus*, and *Stanford Bunny*, respectively (the error reduction relative to the linear vertex interpolation case is given in brackets). In all cases, the root mean square ($\|\cdot\|_2$) and maximum ($\|\cdot\|_\infty$) values of the shape errors are maximal with linear vertex interpolation and minimal using the proposed method. Overall, the proposed method performs significantly better than the other two methods when the free-boundary is not locally almost normal to the considered grid edge (which mostly happens in the cases 1, 3, and 4 of Fig. 3). When the free-boundary is locally almost normal to the grid edge (e.g., case 2 of Fig. 3), then both Manson et al. [5] and the proposed method give results of similar accuracy. This can be clearly observed in Fig. 5 as well as in Figs. 6 and 7. From a visual point of view, if there is an obvious improvement between linear vertex interpolation and both Manson et al. [5] and the proposed method, there is no noticeable difference between these last two methods. When extracting information from the reconstructed surface, however, there can be a significant improvement with the proposed method compared to Manson et al. [5]. To quantify this gain, the impact of the vertex interpolation method on the local mean curvature of the reconstructed *Sphere* is studied. Curvature is evaluated at each vertex of the reconstructed surface by fitting a parabolic patch onto a local neighbourhood of vertices,[2] using the fitting approach of Taubin [24]. Tests are conducted over a range of grid resolutions, and 100 simulations are performed with random center positions for each resolution. The root mean square (RMS) errors are averaged over the 100 simulations, and the maximum errors correspond to the maximum encountered during the 100 simulations. The results, plotted in Fig. 8, show that all vertex interpolation methods tested (i.e., linear interpolation —□—, Manson et al. —△—, and the proposed method —○—) yield shape errors that converge with mesh refinement. Over the range of resolutions tested, the proposed method consistently performs 25 to 50 percents better than Manson et al. [5] with regards to the RMS shape error. The maximum shape errors associated with these two methods present notable differences: as the resolution $R/\Delta x$ increases, the method of Manson et al. [5] exhibits a convergence rate somewhat between first- and second-order, whereas the method proposed in this paper converges with second-order accuracy. Indeed, use of the method of Manson et al. [5] in 3D cannot yield second-order accurate results since it is not able to consider arbitrary surface normal orientations, while the method proposed in this paper relies on second-order accurate piecewise-linear local surface approximations. This improvement translates into significant differences when considering the curvature errors. For $\Delta x < R/10$,



(a) Reference surface (b) Manson et al. (c) Proposed method

Method	$\|\varepsilon\|_2/D$	$\|\varepsilon\|_\infty/D$
Linear interpolation	5.88×10^{-4}	1.37×10^{-3}
Manson et al.	1.99×10^{-4} ($\times 0.34$)	1.13×10^{-3} ($\times 0.82$)
Proposed method	1.60×10^{-4} ($\times 0.27$)	5.62×10^{-4} ($\times 0.41$)

Fig. 6. Shape error ε for the reconstruction of a *Genus* with $D/\Delta x = 75$ (D being the diagonal of the surface's bounding box) using different vertex interpolation methods. The reference error used for visualisation is $\varepsilon_{\text{ref}} = 4.2 \times 10^{-4}\, D$.

(a) Reference model (b) Manson et al. (c) Proposed method

Method	$\|\varepsilon\|_2/D$	$\|\varepsilon\|_\infty/D$
Linear interpolation	4.88×10^{-4}	2.52×10^{-3}
Manson et al.	2.87×10^{-4} ($\times 0.59$)	2.52×10^{-3} ($\times 1.00$)
Proposed method	2.56×10^{-4} ($\times 0.53$)	2.13×10^{-3} ($\times 0.85$)

Fig. 7. Shape error ε for the reconstruction of the *Bunny* with $D/\Delta x = 100$ (D being the diagonal of the surface's bounding box) using different vertex interpolation methods. The reference error used for visualisation is $\varepsilon_{\text{ref}} = 5.2 \times 10^{-4}\, D$.

2. The local neighbourhood of a vertex is here defined as the two consecutive layers of neighbours surrounding it.

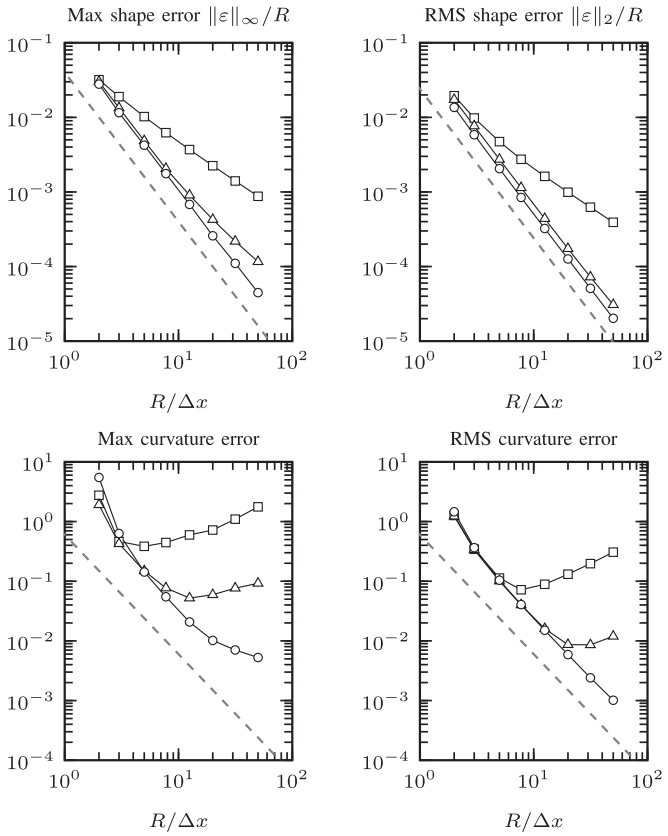
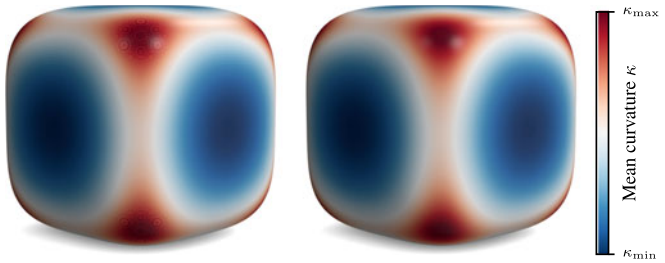


Fig. 8. Shape and curvature errors for the reconstruction of a *Sphere* with different mesh resolutions. Squares —□— represent linear vertex interpolation, triangles —△— represent the vertex interpolation method of Manson et al. [5], and empty circles —○— represent the proposed method. The slope of the dashed line corresponds to second-order convergence.

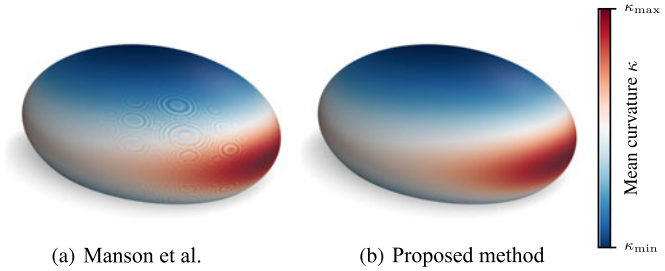


(a) Manson et al. (b) Proposed method

Fig. 9. Mean curvature of a *Smooth Box* evaluated by fitting a parabolic patch on the local neighbourhood of each vertex. Low curvatures are represented in blue and high curvatures in red. (a) The surface is reconstructed with Marching-Cubes using the vertex interpolation method of Manson et al. [5]; (b) The method proposed in this paper is employed.

the maximum curvature error is consistently larger with the method of Manson et al. [5] than with the method proposed in this paper. When reaching $\Delta x = R/50$, the average curvature error with the method of Manson et al. [5] is more than one order of magnitude larger than with the proposed method. The reduction of the curvature errors associated with the proposed method is illustrated in Figs. 9 and 10 which show the mean curvature fields of a *Smooth Box* and an *Ellipsoid*. With the method of Manson et al. [5], visible error patterns can be observed on the mean curvature field, in locations where the reconstruction errors are the largest. Using the proposed method, these errors are greatly reduced, as shown in Figs. 9b and 10b.

Finally, execution times for the reconstruction of the *Sphere* with $R/\Delta x = 50$ are given in Table 1. They are averaged over 100 simulation runs, and the three initial configurations *Mid-edge*, *Linear*, and *Manson* are considered. It is worth mentioning that the majority of cells do not contain any surface element, yet they are explored during the Marching-Cubes reconstruction process. The cost associated with these cells is considered negligible as a simple test on the volume fraction values at the corners of each cell allow to determine whether to skip them or not. Additionally, Table 2 shows the average number of iterations (in step 3 of Algorithm 1) necessary to reach convergence to the fixed-point surface solution, for the cases presented in Figs. 5, 6, and 7. Here also, the three initial configurations *Mid-edge*, *Linear*, and *Manson*, are considered.



(a) Manson et al. (b) Proposed method

Fig. 10. Mean curvature of an *Ellipsoid* evaluated by fitting a parabolic patch on the local neighbourhood of each vertex. Low curvatures are represented in blue and high curvatures in red. (a) The surface is reconstructed with Marching-Cubes using the vertex interpolation method of Manson et al. [5]; (b) The method proposed in this paper is employed.

TABLE 1
Average Execution Times (Given in *s*) for the reconstruction of the *Sphere* with $R/\Delta x = 50$ Using the Method Proposed in this Paper, and for Different Initial Surface Guesses

Initial guess	Mid-edge	Linear	Manson
Marching-Cubes reconstruction	0.278	0.308 (×1.11)	0.326 (×1.17)
Normals evaluation	0.996	0.995 (×0.99)	0.996 (×1.00)
Iterative vertex projection	0.786	0.521 (×0.66)	0.440 (×0.56)
Total	2.060	1.824 (×0.89)	1.762 (×0.86)

The time increase relative to the "mid-edge" initial guess case is given in brackets.

TABLE 2
Average Number of Iterations to Reach the Fixed-Point Solution for the Cases Presented in Figs. 5, 6, and 7, Using the Method Proposed in this Paper, and for Different Initial Surface Guesses

	Sphere	Genus	Bunny
Mid-edge initial guess	4.93	6.75	8.68
Linear initial guess	4.42	6.25	8.10
Manson et al. initial guess	3.49	4.06	7.22

Several conclusions can be drawn out of these tables. Table 1 logically shows that the more elaborate the initial guess strategy is, the more expensive the Marching-Cubes reconstruction process becomes. This is counter-balanced by the fact that the more elaborate and accurate the initial guess strategy is, the less expensive the iterative vertex positioning process becomes. This concurs with Table 2 which shows that the complexity and accuracy of the initial guess is inversely proportional to the number of iterations necessary to reach convergence to the fixed-point solution. Overall, it appears from Table 1 that using the method of Manson et al. [5] for the initial surface reconstruction (i.e., the most expensive and accurate initial guess option) allows for the fastest execution of Algorithm 1. The iterative vertex positioning is indeed associated with a higher cost than the Marching-Cubes reconstruction, due to the resolution of the inverse problems $\alpha = f(\boldsymbol{n}, \gamma)$, which require to

solve cubic polynomial equations. For all cases tested, the additional costs associated with the estimation of the surface normals and the iterative vertex positioning process induce a total execution time that is 5 to 10 times larger than the Marching-Cubes reconstruction itself.

5 Conclusion

A procedure to accurately position the vertices of a surface reconstructed from a discrete indicator function (or volume fraction) field has been presented. The proposed method, which relies on the computation of local volume-fraction-compliant piecewise-linear approximations of the surface, conserves the locality of the vertex interpolation process and does not require to solve local least-squares problems. A range of tests on algebraic surfaces and a 3D model, show that the method performs better than other techniques thanks to its ability to consider arbitrary normals for the local linear approximations used in the vertex interpolation problem. The associated reduction of reconstruction errors is hardly visible on the triangulation itself, but leads to significant improvements when extracting information from the triangulation, such as the local curvature components. This, for example, improves both the accuracy and stability of surface tension dominated interfacial flow simulations, where an accurate evaluation of the interface curvature is essential.

Acknowledgments

The *Stanford Bunny* is issued from the Stanford 3D Model Repository,[3] and the *Golf Ball* model has been obtained from the Princeton University Suggestive Contour Gallery.[4] We are grateful to Petrobras and the EPSRC (grant EP/M021556/1) for their financial support. We would like to acknowledge the anonymous reviewers of this manuscript for their helpful comments and suggestions.

References

[1] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. Eurographics Symp. Geom. Process.*, 2006, pp. 61–70.
[2] J. Manson, G. Petrova, and S. Schaefer, "Streaming surface reconstruction using wavelets," *Comput. Graph. Forum*, vol. 27, no. 5, pp. 1411–1420, 2008.
[3] C. Hirt and B. Nichols, "Volume of fluid (VOF) method for the dynamics of free boundaries," *J. Comput. Physics*, vol. 39, no. 1, pp. 201–225, 1981.
[4] G. Taubin, "A signal processing approach to fair surface design," in *Proc. 22nd Annu. Conf. Comput. Graph. Interactive Techn.*, 1995, pp. 351–358.
[5] J. Manson, J. Smith, and S. Schaefer, "Contouring discrete indicator functions," *Comput. Graph. Forum*, vol. 30, no. 2, pp. 385–393, 2011.
[6] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *Comput. Graph.*, vol. 21, no. 4, pp. 163–169, 1987.
[7] J. Bloomenthal, "An implicit surface polygoniser," in *Graphics Gems IV*, Heckbert, Ed. New York, NY, USA: Academic, 1994, pp. 324–349.
[8] S. Fuhrmann, M. Kazhdan, and M. Goesele, "Accurate Isosurface interpolation with Hermite data," in *Proc. Int. Conf. 3D Vis.*, 2015, pp. 256–263.
[9] T. Abadie, J. Aubin, and D. Legendre, "On the combined effects of surface tension force calculation and interface advection on spurious currents within volume of fluid and level set frameworks," *J. Comput. Physics*, vol. 297, pp. 611–636, 2015.
[10] F. Evrard, F. Denner, and B. van Wachem, " Estimation of curvature from volume fractions using parabolic reconstruction on two-dimensional unstructured meshes," *J. Comput. Physics*, vol. 351, pp. 271–294, 2017.
[11] Y. Ohtake, A. Belyaev, and I. Bogaevski, "Mesh regularization and adaptive smoothing," *Comput.-Aided Des.*, vol. 33, pp. 789–800, 2001.
[12] S. F. F. Gibson, "Constrained elastic surface nets: Generating smooth surfaces from binary segmented data," in *Proc. 1st Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, 1998, pp. 888–898.
[13] A. Chica, J. Williams, C. Andujar, P. Brunet, I. Navazo, J. Rossignac, and A. Vinacua, "Pressing: Smooth Isosurfaces with flats from binary grids," *Comput. Graph. Forum*, vol. 27, no. 1, pp. 36–46, 2008.
[14] V. Lempitsky, "Surface extraction from binary volumes with higher-order smoothness," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1197–1204.
[15] I. Prilepov, H. Obermaier, E. Deines, C. Garth, and K. I. Joy, "Cubic gradient-based material interfaces," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 10, pp. 1687–1699, Oct. 2013.
[16] C. Ledergerber, G. Guennebaud, M. Meyer, M. Bächer, and H. Pfister, "Volume MLS ray casting," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 6, pp. 1372–1379, Nov. 2008.
[17] K. S. Bonnell, D. R. Schikore, K. I. Joy, M. Duchaineau, and T. H. E. T.-M. Case, "Constructing material interfaces from data sets with volume-fraction information," in *Proc. Conf. Vis.*, 2000, pp. 367–372.
[18] J. E. Pilliod and E. G. Puckett, "Second-order accurate volume-of-fluid algorithms for tracking material interfaces," *J. Comput. Physics*, vol. 199, no. 2, pp. 465–502, 2004.
[19] R. Scardovelli and S. Zaleski, "Analytical relations connecting linear interfaces and volume fractions in rectangular grids," *J. Comput. Physics*, vol. 164, no. 1, pp. 228–237, 2000.
[20] S. Cummins, M. Francois, and D. Kothe, "Estimating curvature from volume fractions," *Comput. Struct.*, vol. 83, no. 6/7, pp. 425–434, Feb. 2005.
[21] E. Aulisa, S. Manservisi, R. Scardovelli, and S. Zaleski, "Interface reconstruction with least-squares fit and split advection in three-dimensional Cartesian geometry," *J. Comput. Physics*, vol. 225, no. 2, pp. 2301–2319, Aug. 2007.
[22] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, and S. Zaleski, "Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows," *J. Comput. Physics*, vol. 152, pp. 423–456, 1999.
[23] S. Bnà, S. Manservisi, P. Yecko, and S. Zaleski, "VOFI—A library to initialize the volume fraction scalar field," *Comput. Physics Commun.*, vol. 200, pp. 291–299, 2016.
[24] G. Taubin, "Estimation of planar curves, surfaces and nonplanar space curves," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 11, pp. 1115–1138, Nov. 1991.

3. http://graphics.stanford.edu/data/3Dscanrep/
4. http://gfx.cs.princeton.edu/proj/sugcon/models/