Multi-phase Volume Segmentation with Tetrahedral Mesh

Tuan T. Nguyen
tntr@dtu.dk
Vedrana A. Dahl
vand@dtu.dk
J. Andreas Bærentzen
janba@dtu.dk
Camilla H. Trinderup
ctri@dtu.dk

Department of Applied Mathematics
and Computer Science
Technical University of Denmark
Kgs. Lyngby, DK

Abstract

In life science and material science, it is often desirable to segment a volumetric data set in such a way that multiple materials (phases) are segmented and a tetrahedral mesh representation is obtained for each segment for downstream applications. Unfortunately, obtaining a mesh, typically from CT or MRI scan, is challenging, especially in 3D. This paper proposes a novel approach for volume segmentation using a tetrahedral mesh. Our method employs a deformable model that minimizes the Mumford-Shah energy function. We apply our method to several CT data sets in order to demonstrate its advantages: multi-phase support, robustness to noise, and adaptive resolution outputs. Our method is based on the Deformable Simplicial Complex (DSC) method for tracking deformable interfaces which is designed specifically to deal with topology changes.

1 Introduction

In recent years, it has become an increasingly important concern to build simulation meshes from data acquired using one of the many CT or MRI based scanning modalities. Generally, tetrahedral meshes for finite element or geometric analysis are required, and very often the objects being scanned are heterogeneous, leading to the need for a multi-phase segmentation of the scanned object.

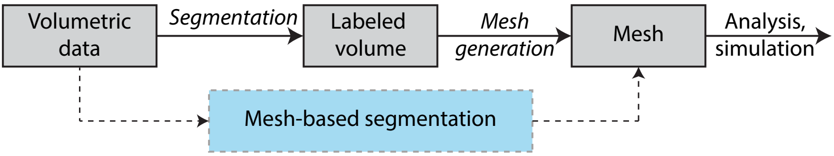


Figure 1: Common work flow in volumetric segmentation for material analysis

An example we use in this paper is concerned with segmenting three phases in the scan of a solid oxide fuel cell. Characterization of the fuel cell, and simulation of its operation, requires a precise description of interfaces between materials, including the triple phase boundary. Another example is segmentation of fibre bundles in composite material which can be used for estimating the material's overall stiffness and strength. The most common workflow for such analysis is illustrated in Fig. 1. The process is split into two parts: a segmentation that operates at the voxel level and mesh generation which produces the desired result from the segmentation.

In this paper, we propose to go directly from the initial volumetric data to a mesh-based representation of the segmentation as illustrated with the blue box in Fig. 1. The pivotal idea is to represent the entire domain as an irregular tetrahedral mesh where each tetrahedron is assigned a label corresponding to the material (or phase) to which it belongs. This simplifies dealing with multiple interfaces since the interface between two materials is simply the set of triangular faces shared by tetrahedra with corresponding labels. Moreover, we surmise that for further processing, typically simulation, it is a significant advantage that the internal representation is also the desired output representation.

Related ideas for two-phase segmentation are well researched in the literature named *deformable model*, a model that is strong against noise and artifacts and suitable for data with homogeneous regions representing materials [43]. First introduced by Terzopoulos *et al.* [17, 36, 37] this has become one of the most successful approaches to image segmentation. In deformable model, we are concerned with the curves/surfaces representation and the force model.

The force models were proposed initially [11, 12, 15, 17, 21, 37, 38, 42]. Despite the variety, all those models use local differential properties of image edges, hence the segmentation often sticks to local minima. User-driven forces may be required to achieve desired segmentation [17]. To overcome the above issue, many authors utilize a global energy function proposed by Mumford and Shah [25], one of the most popular model in deformable model with many applications [3]. This paper shall focus on deformable models that minimize the Mumford-Shah energy functional.

Minimizing the Mumford-Shah functional using an implicit representation (*e.g*. level set method) is popular due to the ease with which topology changes (*e.g*. splitting and merging) are handled. Deformable models using an implicit representation is based on curve evolution [28, 33, 34]. Solving it has been studied in depth with many proposals [1, 4, 5, 6, 7, 8, 18, 22, 32, 35, 39, 40]. Perhaps, the most popular model is the active contour without edge, a two-phase segmentation [8, 9]. For multi-phase, implicit representations have problems in phase overlapping [45]; or are limited to fixed number of phases [40]; or suffer from ill-conditioned equations [20].

In contrast to implicit representations, explicit representations have advantage in representing multi-phase as the interfaces are literally defined. However, the difficulty in handling topological changes is a significant obstacle. Perhaps an effortless approach to overcome this issue is to borrow techniques from explicit interface tracking researches, which leads to many proposals to deal with topological changes. [44] explicitly resolve each intersection, [23, 41] generate new mesh based on an underlying fixed grid, and [10] utilize element deletion technique. Unfortunately, these methods do not support multi-phase yet. For multi-phase, [14] use collision technique, and [31] use Delaunay mesh generation. These two methods only maintain a surface triangle mesh. [24] utilize a tetrahedral mesh and detect topological changes using neighbor information. Among all of these methods, [23, 31] have been applied to image segmentation, but their force models are local. Very few researches accommodate the Mumford-Shah energy function with 3D explicit mesh, and they are limited to segmenting a single region *e.g*. [16].

In [26] the authors have utilized the interface tracking technique in [24] to minimize the Mumford-Shah energy function for 2D problem and showed improvement in accuracy compared to other deformable models. The current paper extends the method to 3D and inherits the advantages: multi-phase support; higher accuracy; automatic resolve junction vertices; and output is an adaptive tetrahedral mesh. Beyond generalizing the method to 3D, the contributions in this paper include:

- a scheme for computation of triangle-based forces with Gaussian quadrature formula (Sec. 3.2),
- a generalization that enables the method to work with various input model beside intensity (Sec. 5),
- and a new mesh adaptation scheme, in which parameters are geometric-based and can be invisible to user (Sec. 4).

2 The Deformable Simplicial Complex (DSC) Method

DSC [24] is an explicit interface tracking method. The DSC method uses a tetrahedral mesh and labels each tetrahedron according to its phase (or material). The interface is defined by triangles whose coboundary tetrahedra have different labels (Fig. 3). DSC takes interface vertex displacements as input, deforms the mesh, and resolves topological changes automatically. An implementation of the DSC in C++ is publically available [2].

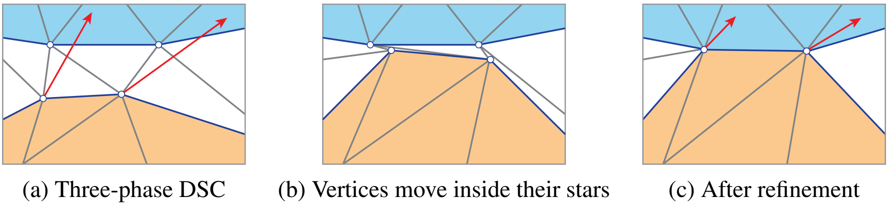

(a) Three-phase DSC (b) Vertices move inside their stars (c) After refinement

Figure 2: The DSC algorithm illustrated in 2D. Red arrows denote desired displacements

The DSC algorithm (Fig. 2) includes two steps. First it moves interface vertices as far as possible inside their stars (neighbor tetrahedra). At this stage, the mesh is still valid (i.e. no self intersection or inverted tetrahedron) but contains downgraded entities. In the second step, the DSC refines the mesh to remove low quality entities. The refinement involves mesh smoothing, edge collapse and edge split. Because the displacement is limited by the star of a vertex, the DSC may requires some iterations to move the interface to the desired destination.

3 Method

Given an image $I : \Omega \to \mathbb{R}$, we want to find a piecewise constant function $u(\mathbf{x}) = c_i$ if $\mathbf{x} \in \Omega_i$ (where Ω_i denotes a disjoint phase of constant intensity c_i) that minimizes the Mumford-Shah functional

$$E(u) = \sum_{i=1}^{N} \int_{\Omega_i} (I - c_i)^2 \mathrm{d}\Omega + \alpha \|\partial u\| \tag{1}$$

where $\|\partial u\|$ denotes the area of the interface, α is the smoothing coefficient, and N is the number of phases.

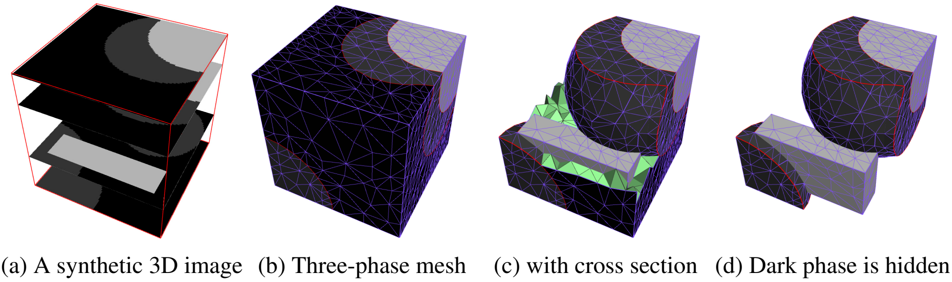

(a) A synthetic 3D image (b) Three-phase mesh (c) with cross section (d) Dark phase is hidden

Figure 3: A piecewise constant function defined on a tetrahedral mesh. Green triangles are internal triangles. Blue/red edges are boundary/junction edges, respectively

We define a piecewise constant function on a tetrahedral mesh by labeling the tetrahedra with a labeling function $\{l_i\} : \{t_i\} \to \mathbb{Z}, l_i \in [1,N]$. Faces that have coboundary tetrahedra with different labels define the interface (Fig. 3). Edges and vertices on the interface are interface edges and interface vertices.

In our method, the unknowns are the positions of interface vertices $\{\mathbf{p}_i\}$, phase intensities $\{c_i\}$ and the labeling function $\{l_i\}$. The function we want to find is $u = u(\{\mathbf{p}_i\}, \{c_i\}, \{l_i\})$. We treat the minimization problem independently for these unknowns, leading to three minimization problems

$$1) \quad \min_{\{c_i\}} E(u) \qquad 2) \quad \min_{\{\mathbf{p}_i\}} E(u) \qquad 3) \quad \min_{\{l_i\}} E(u) \tag{2}$$

3.1 Minimize E with respect to $\{c_i\}$ and $\{l_i\}$

Phase intensities We find c_i by setting the partial derivative of E with respect to c_i equal to zero

$$\frac{\partial E}{\partial c_i} = \frac{\partial}{\partial c_i} \int_{\Omega_i} (c_i - I)^2 \mathrm{d}\Omega = 2c_i \int_{\Omega_i} d\Omega - 2 \int_{\Omega_i} I d\Omega = 0 \implies c_i = \frac{\int_{\Omega_i} I d\Omega}{\text{Volume}(\Omega_i)} \tag{3}$$

meaning c_i is the mean intensity of the image in phase Ω_i.

Labeling function We find the optimal label of a tetrahedron by choosing the phase that minimizes the energy (Eq. 1) inside the tetrahedron. Note that changing the label also changes the interface, which must be included in computation of the internal energy. For external energy, we approximate the volume integral with Riemann sum and discretize the tetrahedron to a set of sampling points with size of one voxel.

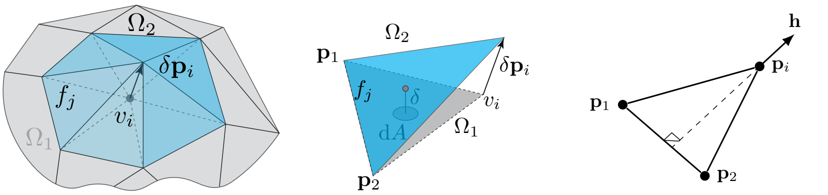
3.2 Minimize E with respect to interface vertices

We move interface vertices in the gradient descent direction. Consider an interface vertex with position $\mathbf{p}_i$, it will be displaced by $\delta \mathbf{p}_i = \nabla_{\mathbf{p}_i} E \mathrm{d}t$, where $\mathrm{d}t$ is the time step. We separate the two terms in Eq. 1 to external energy $E^{\text{ext}} = \sum_{i=1}^{N} \int_{\Omega_i} (I - c_i)^2 \mathrm{d}\Omega$ and internal energy $E^{\text{int}} = \alpha \|\partial u\|$. The gradient of the two energy functions are called the external force $\mathbf{F}^{\text{ext}}$ and the internal force $\mathbf{F}^{\text{int}}$, respectively.

Compute internal force The area of the interface can be computed explicitly as a sum of the areas of all interface triangles. The gradient of the internal energy with respect to a vertex v_i is

$$\mathbf{F}_i^{\text{int}} = \frac{\partial}{\partial \mathbf{p}_i} E^{\text{int}} = \alpha \sum_{f \in \mathcal{N}_i} \frac{\partial}{\partial \mathbf{p}_i} \text{Area}(f) = \alpha \sum_{f \in \mathcal{N}_i} \|\mathbf{p}_1 - \mathbf{p}_2\| \, \mathbf{h} \tag{4}$$

where f denotes an interface triangle; $\mathcal{N}_i$ denotes the neighbor of v_i; $\mathbf{p}_1, \mathbf{p}_2$ are the positions of the two other vertices in f; and $\mathbf{h}$ is the normalized height vector corresponds to $\mathbf{p}_i$ (See Fig. 4c).



(a) Vertex v_i moves by $\delta\mathbf{p}_i$ (b) Triangle integration (c) Triangle area derivative

Figure 4: Illustration for computing forces on v_i. (a) Volume covered by triangles displacement changing intensity from c_2 to c_1 (b) Integration over volume covered by displacement of triangle f_j (c) Derivative of triangle area with respect to a vertex

Compute external force Local displacement of a vertex v_i only changes the piecewise constant function u in the volume covered by the displacement of neighbor interface triangles (See Fig. 4a). Assuming that the displacement $\delta\mathbf{p}_i$ is small, the image intensity on the interface triangles can be considered unchanged, and integration over the volume can be approximated with integration over the surface. The external energy change caused by the displacement of v_i is

$$\Delta E_i^{\text{ext}} = \sum_{f \in \mathcal{N}_i} \int_f \left((I - c_1)^2 - (I - c_2)^2 \right) \delta \text{d}A = \sum_{f \in \mathcal{N}_i} (c_1 - c_2) \int_f (2I - c_1 - c_2) \delta \text{d}A \tag{5}$$

where δ denotes the orthogonal displacement of the small area dA (See Fig. 4b), and δdA represents a small volume.

We approximate the integration with Gaussian quadrature formulas for triangles [13], which discretizes a triangle to a set sampling points with area coordinate $\{\xi_j, \eta_j, \zeta_j\}$. At a sampling point we have the image intensity I_j, and the orthogonal displacement is $\delta = \xi_j \delta\mathbf{p}_i \cdot \mathbf{n}_f$, where $\delta\mathbf{p}_i \cdot \mathbf{n}_f$) is the dot product that represents the projection of $\delta\mathbf{p}_i$ on the normal direction $\mathbf{n}_f$ of triangle f, and ξ_j is the coordinate correspond to the vertex v_i. We obtain the energy change

$$\Delta E_i^{\text{ext}} = \sum_{f \in \mathcal{N}_i} (c_1 - c_2) A_f \sum_{\text{point} j} \omega_j (2I_j - c_1 - c_2) \xi_j \delta\mathbf{p}_i \cdot \mathbf{n}_f \tag{6}$$

where ω_j is the weight of the sampling point, and A_f is the triangle area.

In Eq. 6 we can see that $\delta\mathbf{p}_i$ only appears in the dot product $\delta\mathbf{p}_i \cdot \mathbf{n}$. Replacing the gradient of a dot product: $\frac{\partial}{\partial \mathbf{p}}(\mathbf{p} \cdot \mathbf{n}) = \mathbf{n}$, we obtain the final external force

$$\mathbf{F}_i^{\text{ext}} = \nabla_{\mathbf{p}_i} E^{\text{ext}} = \sum_{f \in \mathcal{N}_i} \left\{ \mathbf{n}_f A_f (c_1 - c_2) \sum_{\text{point} j} \omega_j \xi_j (2I_j - c_1 - c_2) \right\} \tag{7}$$

3.3 Implementation

In our experiments, the image intensity is scaled to the range $[0:1]$. We transform image coordinate to continuous coordinates by linear interpolation. Our algorithm starts with an uniform mesh with average edge length ε. We utilize Otsu threshold method [29] to initialize the labeling function. In each iteration, we first compute the forces on interface vertices and then deform the mesh using the DSC method. In the second step, we execute adaptive resolution mesh algorithm in Sec. 4. This step can be performed once per 1-5 iterations. Finally, we stop the segmentation if vertex displacements are smaller than 0.01.

To boost the performance, we use sum table to compute $\{c_i\}$ and adaptive time step. Adaptive time step is applied for each interface vertex. We first estimate initial time step for all vertices so that maximum displacement is 0.4ε. In the following step, we scale the time step individually by 1.1 if a vertex moves in the same direction as its previous displacement; and by 0.9 if it moves on the inverse direction. The time step is bounded at $[0.1:3]$. If a vertex is affected by topological events, we reset its time step. We utilize [27] to keep track of modified vertices.

4 Adaptive Resolution Meshes

Adaptivity is important not only for a compact representation but also for memory efficiency and performance in computation. There are two approaches for adaptive mesh: subdivision and coarsening. Perhaps subdivision (start with a sparse mesh then locally subdivide where needed) is more intuitive, but it requires parameters for subdivision criteria. Though these parameters can be useful [26], tuning them can be difficult in 3D. In this paper, we follow the second approach that starts with a dense mesh then locally coarsen where needed.

We utilize edge collapse for mesh coarsening. The criteria for choosing collapsing edge is based on geometric information and in such a way that we deem user tweaking of parameters to be unnecessary. Our algorithm includes internal edge collapse and interface edge collapse described below.

Internal edge collapse does not modify the interface. For each internal vertex v, we choose shortest edge neighboring to v as the potential collapsing edge. We utilize volume-length ratio to measure the tetrahedron quality [30] and only collapse if the qualities of new tetrahedra are larger than 0.3. This 0.3 threshold is independent to input data set.

Interface edge collapse modifies the interface, hence we only collapse an edge on flat surface (mean curvature at removing-vertex smaller than a 0.03). For each interface vertex v, we also pick shortest edge neighboring to v as the potential collapsing edge. We utilize triangle angles to measure triangle quality and only collapse if qualities of new interface triangles are larger than 10°.

5 Generalization

In many cases, intensity alone cannot distinguish features in an image. Methods like dictionary, filtering, *etc.* [19] are efficient approaches that output probability maps $\{\mathcal{P}_i : \Omega \to \mathbb{R}, i = 1:N\}$ of voxels belong to phases. We apply our method for probability input by modifying the energy function to

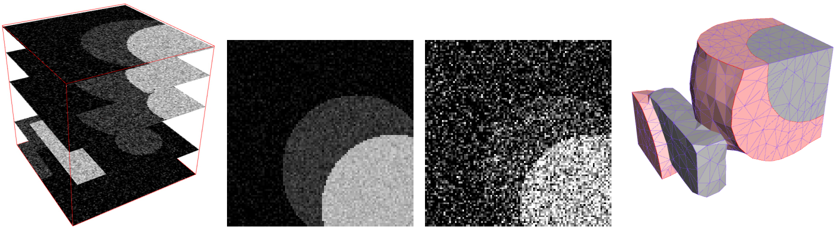
$$E^{\text{prob}}(u) = \sum_{i=1}^{N} \int_{\Omega_i} (1 - \mathcal{P}_i)^2 \mathrm{d}\Omega + \alpha \|\partial u\| \tag{8}$$

Minimizing E^{prob} is similar except that the external force in Eq. 7 now becomes

$$\mathbf{F}_i^{\text{ext}} = \sum_{f\in\mathcal{N}_i} \left\{ \mathbf{n}_f A_f \sum_{\text{point}\, j} \omega_j \xi_j (2-\mathcal{P}_1-\mathcal{P}_2)(\mathcal{P}_1-\mathcal{P}_2) \right\} \tag{9}$$

6 Result and Discussion

Accuracy Our method, and deformable models in general, is strong against noise. We tested our method on a three-phase synthetic data set with a variable level of Gaussian noise (See Fig. 5). In Tab. 1, one can see that our results are consistent while the noise level increases. In another example with CT data in Fig. 6, the result is visually accurate compared to a photograph of the real object.



(a) 3D slices (b) Noise 0.003 (c) Noise 0.1 (d) Segmentation

Figure 5: Segmentation of synthetic data. Volume size 100 × 100 × 100 voxels. (d) Segmentation result with noise variance 0.1

Table 1: Experiment with noisy synthetic data. (a) Variance of the Gaussian noise (using `imnoise` in Matlab). (b) Mean squared difference between noisy volume and the ground truth. (c) Voxel-based mean squared difference between our result and the ground truth (d) Correct segmented voxels.

(a)	0.003	0.01	0.05	0.1
(b)	0.002	0.007	0.03	0.06
(c)	0.0019	0.0018	0.0024	0.003
(d)	99%	99%	98.1%	97%

Multi-phase support is the most important property of our method. We demonstrate the segmentation of two (Fig. 6), three (Fig. 7) and five (Fig. 8) phases, but the number of phases can be arbitrary. It is also noteworthy to mention that the shared interfaces between phases are defined explicitly without fuzziness.

Segmentation with probability input Fig. 8a shows a scan of composite material, where carbon fibers are grouped into bundles characterized by different orientations. By applying an orientation filter [19], we obtain five probability maps (Fig. 8b) of voxels belong to four bundle orientations or the resin. For this task our method takes probability input. The result of five-phase segmentation is shown in Fig. 8.

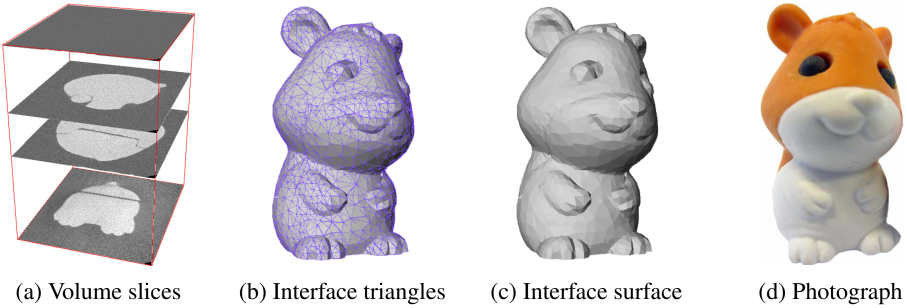

(a) Volume slices (b) Interface triangles (c) Interface surface (d) Photograph

Figure 6: Segmentation of micro CT scan of a plastic figure

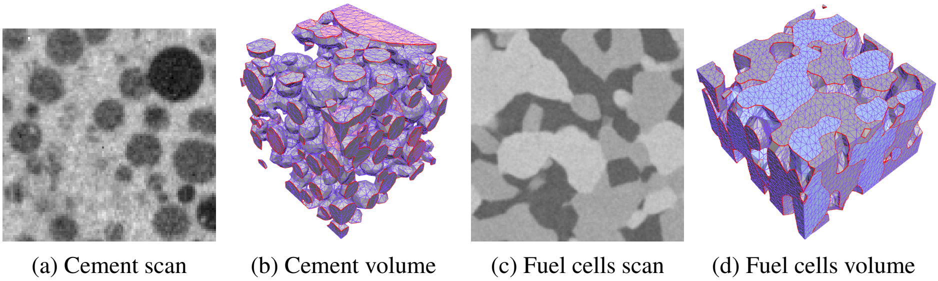

(a) Cement scan (b) Cement volume (c) Fuel cells scan (d) Fuel cells volume

Figure 7: Segmentation of cement scan and fuel cells scan. The third phase (the air) is hidden

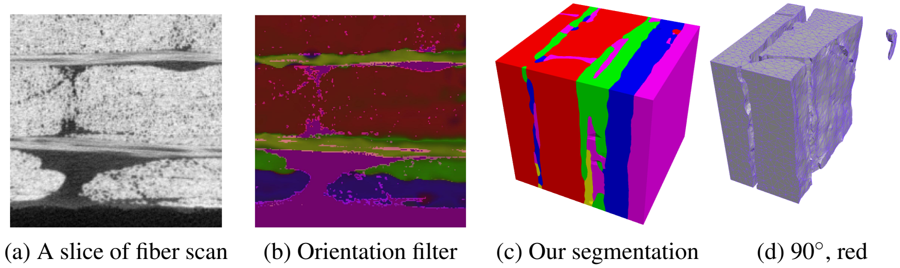
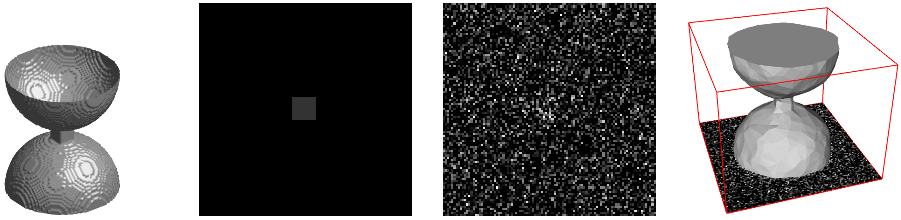

(a) A slice of fiber scan (b) Orientation filter (c) Our segmentation (d) 90°, red

Figure 8: Fiber bundle segmentation, utilizing fiber orientation filter. Color codes: red is 90° bundle, green is 45°, blue is 0°, yellow is $-45°$, and pink is the resin. We demonstrate the mesh of phase 1 (90° orientation) in (d)

Comparison with 2D slice segmentation Using 3D information can improve the segmentation accuracy. In Fig. 9, it is difficult to segment the center slice unless we start with a good initialization that requires manual input. On the other hand, our method can still automatically segment that small region because we use support from other slices.

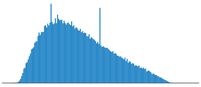
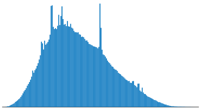
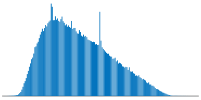
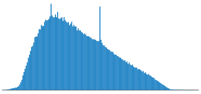


(a) Test data surface (b) Center slice (c) with noise (d) Our segmentation

Figure 9: Segmentation contain slices with small features. Noise is Gaussian with variance 0.1. Volume size 100 × 100 × 80.

Output is an adaptive mesh is an advantage of our method over implicit representation. Adaptivity is an integral part of the algorithm, and it helps reduce the mesh size from initial mesh to final output up to 70% while still maintains a good quality mesh (mean dihedral angle is around 45° and smallest dihedral angle is acceptable in Tab. 2). This quality is sufficient for visualization and measurement. For procedures that require higher quality, we can tweak the threshold in the adaptivity algorithm and the DSC, which also leads to a less adaptive output.

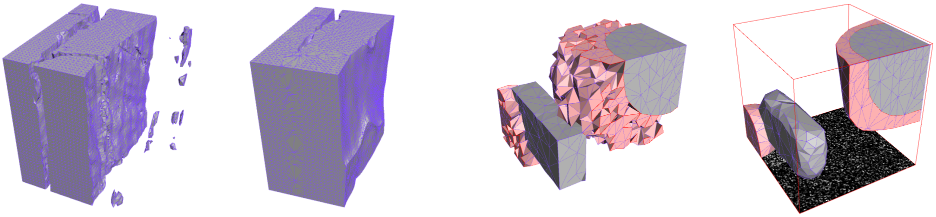
Table 2: Mesh statistic. All numbers in (c,d) are in thousands. (a) min/max dihedral angle (b) Dihedral angle histogram [0° : 180°] (high peaks at 45° and 90° are tetrahedra at the boundary of the domain) (c) # init tetrahedra / # final tetrahedra / # final interface traingles (d) # voxels

	Fig. 6	Fig. 7a	Fig. 7c	Fig. 8
(a)	4.9° - 176°	3.4° - 179°	1.8° - 179°	2.8° - 178°
(b)				
(c)	317 / 92 / 9.4	257 / 220 / 61	260 / 140 / 27	421 / 185 / 31
(d)	5584	1000	2250	8000

Two parameters are important for our method: the edge length ε at initialization and the smoothing coefficient α. Choosing ε is straightforward as it represents the size of the smallest feature we want to segment. Choosing α requires some trial and error. Fig. 10 shows the effect of α: If α is too small, we segment noise while a large α tends to over-smooth. In our experience, the best α is often in the range from 0.01 to 0.5 (Tab. 3).

Table 3: Experiment detail. The computation time is separated to forces computation and meshing

Data set	Volume size	Edge length	α	# iters	Computation time [sec]	Energy (int/ext)
Fig. 6	170 × 150 × 219	5	0.03	50	64 - 114	11000 / 1000
Fig. 7a	100 × 100 × 100	3	0.01	60	91 - 205	7600 / 3900
Fig. 7c	150 × 150 × 100	4	0.03	28	59 - 74	3040 / 4983
Fig. 8	200 × 200 × 200	5	0.1	70	175 - 595	340000 / 25900



(a) 0.01 (b) 0.5 (c) 0.01 (d) 0.8

Figure 10: Smoothing effect. (a)(b) fiber bundle phase 1, reference to Fig. 8d. (c)(d) synthetic data, reference to Fig. 5b

In conclusion, this paper proposes a volume segmentation method using a deformable model based on a tetrahedral mesh. Our method minimizes the Mumford-Shah functional, which is a global, noise resilient energy function.

Our method is based on DSC which is an explicit (or Lagrangian) method for interface tracking that is still volumetric in the sense that the entire domain is partitioned into labeled tetrahedra. It is this approach which allows us to easily handle multiple materials while also resolving topological changes during segmentation.

7 Acknowledgment

The authors acknowledge the financial support from CINEMA: the allianCe for ImagiNg of Energy MAterials, DSF-grant no. 1305-00032B under the Danish Council for Strategic Research. We thank Luis Pedro Esteves formerly at DTU Civil Engineering and GN ReSound for allowing us to test our method on their data.

References

[1] Luigi Ambrosio and Vincenzo Maria Tortorelli. Approximation of functional depending on jumps by elliptic functional via t-convergence. *Communications on Pure and Applied Mathematics*, 43(8):999–1036, 1990.

[2] J. Andreas Bærentzen. Deformable Simplicial Complex (DSC) method, 2013. URL https://github.com/janba/DSC.

[3] Leah Bar, Tony F. Chan, Ginmo Chung, Miyoun Jung, Nahum Kiryati, Rami Mohieddine, Nir Sochen, and Luminita A. Vese. Mumford and Shah Model and its applications to image segmentation and image restoration. In *Handbook of Mathematical Methods in Imaging*, pages 1097–1154. 2011.

[4] Andrew Blake and Andrew Zisserman. *Visual reconstruction*. MIT press, 1987.

[5] Blaise Bourdin. Image segmentation with a finite element method. *Mathematical Modelling And Numerical Analysis*, 33:229–244, 1999.

[6] Antonin Chambolle. Finite-differences discretizations of the Mumford-Shah functional. *Mathematical Modelling and Numerical Analysis*, 33(2):261–288, 1999.

[7] Antonin Chambolle and Gianni Dal Maso. Discrete approximation of the Mumford-Shah functional in dimension two. *Mathematical Modelling and Numerical Analysis*, 33(4):651–672, 1999.

[8] Tony F. Chan and Luminita A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[9] Tony F. Chan, Selim Esedoglu, and Mila Nikolova. Algorithms for Finding Global Minimizers of Image Segmentation and Denoising Models. *SIAM Journal on Applied Mathematics*, 66(5):1632–1648, jan 2006.

[10] Nuttapong Chentanez, Matthias Müller, Miles Macklin, and Tae-yong Kim. Fast grid-free surface tracking. *ACM Transactions on Graphics*, 34(4):148:1–148:11, jul 2015.

[11] Laurent D Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 53(2):211–218, 1989.

[12] Laurent D. Cohen and Isaac Cohen. Finite-element methods for active contour models and balloons for 2-D and 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9212248):1131–1147, 1993.

[13] G. R. Cowper. Gaussian quadrature formulas for triangles. *International Journal for Numerical Methods in Engineering*, 7(3):405–408, 1972.

[14] Fang Da, Christopher Batty, and Eitan Grinspun. Multimaterial mesh-based surface tracking. *ACM Transactions on Graphics*, 33(4):1–11, jul 2014.

[15] H. Delingette. Simplex meshes: a general representation for 3D shape reconstruction. *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 856–859, 1994.

[16] Alexandre Dufour, Nicole Vincent, and Auguste Genovesio. 3D Mumford-Shah based active mesh. *Progress in Pattern Recognition, Image Analysis and Applications*, 4225: 208–217, 2006.

[17] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes : Active Contour Models. *International Journal of Computer Vision*, 331(4):321–331, 1988.

[18] Georges Koepfler, Christian Lopez, and Jean-Michel Morel. A multiscale algorithm for image segmentation by variational method. *SIAM journal on numerical analysis*, 31(1):282–299, 1994.

[19] M. Krause, J. M. Hausherr, B. Burgeth, C. Herrmann, and W. Krenkel. Determination of the fibre orientation in composites using the structure tensor and local X-ray transform. *Journal of Materials Science*, 45(4):888–896, 2010. ISSN 00222461.

[20] Johan Lie, Marius Lysaker, and Xue-Cheng Tai. A variant of the level set method and applications to image segmentation. *Mathematics of Computation*, 75(255):1155–1175, 2006.

[21] David MacDonald, David Avis, and Alan C. Evans. Multiple surface identification and matching in magnetic resonance images. In Richard A. Robb, editor, *Proc. SPIE 2359, Visualization in Biomedical Computing*, number September 1994, pages 160–169, sep 1994.

[22] G. Maso, J. M. Morel, and S. Solimini. A variational method in image segmentation: Existence and approximation results. *Acta Mathematica*, 168(1):89–151, 1992.

[23] T McInerney and D Terzopoulos. Topology adaptive deformable surfaces for medical image volume segmentation. *IEEE transactions on medical imaging*, 18(10):840–850, 1999.

[24] Marek Krzysztof Misztal and Jakob Andreas Bærentzen. Topology-adaptive interface tracking using the deformable simplicial complex. *ACM Transactions on Graphics*, 31 (3):1–12, may 2012.

[25] David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989.

[26] Tuan Nguyen, Vedrana Dahl, and Jacob Bærentzent. Multi-phase image segmentation with the adaptive deformable mesh. *in submission*, 2017.

[27] Tuan T. Nguyen, Vedrana A. Dahl, and J. Andreas Bærentzen. Cache-mesh, a Dynamics Data Structure for Performance Optimization. *Procedia Engineering*, 203:193–205, 2017.

[28] Stanley J. Osher. Fronts Propagating with Curvature Dependent Speed. *Computational Physics*, 79(1):1–5, 1988.

[29] Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[30] V N Parthasarathy, C M Graichen, and A F Hathaway. A comparison of tetrahedron quality measures. *Finite Elements in Analysis and Design*, 15(3):255–261, 1994.

[31] Jean Philippe Pons and Jean Daniel Boissonnat. Delaunay Deformable Models: Topology-Adaptive Meshes Based on the Restricted Delaunay Triangulation. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 13:384–394, 2007.

[32] Christophe Samson, Laure Blanc-Féraud, Gilles Aubert, Josiane Zerubia, Christophe Samson, Laure Blanc-Féraud, Gilles Aubert, Josiane Zerubia, and Multiphase Evolution. Multiphase Evolution and Variational Image Classification. Technical Report RR-3662, INRIA, apr 1999.

[33] J. A. Sethian. Curvature and the evolution of fronts. *Communications in Mathematical Physics*, 101(4):487–499, dec 1985.

[34] JA Sethian. A review of recent numerical algorithms for hypersurfaces moving with curvature dependent speed. *J. Differential Geometry*, 31:131–161, 1989.

[35] J. Shah. A common framework for curve evolution, segmentation and anisotropic diffusion. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, number I, pages 136–142. IEEE, 1996.

[36] Demetri Terzopoulos. On Matching Deformable Models to Images. *Topical Meeting on Machine Vision*, 12:160–167, 1986.

[37] Demetri Terzopoulos and Kurt Fleischer. *Deformable models*. 1988.

[38] Demetri Terzopoulos, Andrew Witkin, and Michael Kass. Constraints on deformable models: Recovering 3d shape and nonrigid motion. *Artificial intelligence*, 36(1):91–123, 1988.

[39] Andy Tsai, Anthony Yezzi, and A.S. Willsky. Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Transactions on Image Processing*, 10(8):1169–1186, 2001.

[40] Luminita A. Vese and Tony F. Chan. A multiphase level-set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50(3)(3):271–279, 2002.

[41] Chris Wojtan, Nils Thürey, Markus Gross, and Greg Turk. Deforming meshes that split and merge. *ACM Transactions on Graphics*, 28(3):1, jul 2009.

[42] Chenyang Xu and Jerry L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998.

[43] Chenyang Xu, Jerry L Prince, and Dzung L. Pham. Image Segmentation Using Deformable Models. In *Handbook of Medical Imaging*, pages 129–174. 2004.

[44] A Zaharescu, E Boyer, and R Horaud. Topology-Adaptive Mesh Deformation for Surface Evolution, Morphing and Multiview Reconstruction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PP(99):1, 2010.

[45] Hong-Kai Zhao, T. Chan, B. Merriman, and S. Osher. A Variational Level Set Approach to Multiphase Motion. *Journal of Computational Physics*, 127:179–195, 1996.