Universität Regensburg
Mathematik

Segmentation of three-dimensional images
with parametric active surfaces and
topology changes

Heike Benninghoff and Harald Garcke

Segmentation of Three-dimensional Images with Parametric Active Surfaces and Topology Changes

Heike Benninghoff[*] and Harald Garcke[†]

Abstract

In this paper, we introduce a novel parametric method for segmentation of three-dimensional images. We consider a piecewise constant version of the Mumford-Shah and the Chan-Vese functionals and perform a region-based segmentation of 3D image data. An evolution law is derived from energy minimization problems which push the surfaces to the boundaries of 3D objects in the image. We propose a parametric scheme which describes the evolution of parametric surfaces. An efficient finite element scheme is proposed for a numerical approximation of the evolution equations. Since standard parametric methods cannot handle topology changes automatically, an efficient method is presented to detect, identify and perform changes in the topology of the surfaces. One main focus of this paper are the algorithmic details to handle topology changes like splitting and merging of surfaces and change of the genus of a surface. Different artificial images are studied to demonstrate the ability to detect the different types of topology changes. Finally, the parametric method is applied to segmentation of medical 3D images.

Keywords: Image segmentation, three-dimensional images, active surfaces, parametric method, topology changes, Mumford-Shah, Chan-Vese, finite element approximation

1 Introduction

One major challenge in image processing is the autonomous detection of objects in images and the segmentation of the objects from each other and from their environment.

A very popular approach for image segmentation is the active contour method [20], [17]. In the case of classical two-dimensional images, one or more curves, called contours, evolve in the two-dimensional image domain and stop locally at edges or region boundaries. The motion is described by evolution equations which aim to minimize a certain energy functional. The energies typically contain length terms to control the smoothness of the contours (internal energies) and terms which push the contours to the desired region boundaries or to edges in the image (external energies).

Two main classes of approaches can be distinguished: The first class are edge-based methods where regions are identified by their boundaries where the image intensity function rapidly changes [20], [23], [13]. The second class are region-based methods where the regions are characterized by the mean gray value or mean color, or by the texture or some other grouping [28], [32], [16], [38].

Region-based active contours methods can also be applied on images with so-called weak edges, i.e. edges with only small changes in the image intensity function [16], and on images which contain regions which are groups of smaller objects [3]. Furthermore, the method is less sensitive to noise. If images with high noise have to be segmented, gradient-based approaches may get trapped at locations where the noise is high and the contours may not detect the *real* objects in the image.

In this paper, we study volumetric, i.e. three-dimensional, images given by a scalar or vector-valued image function $u_0 : \Omega \to \mathbb{R}^{(d)}$, where $\Omega \subset \mathbb{R}^3$ is an open and bounded image domain. Real images are often defined on a set of $N_x \times N_y \times N_z$ voxels (=volume pixels), where u_0 is locally constant on each voxel. 3D images may be reconstructed by certain 3D imaging procedures like computed tomography (CT) or magnetic resonance imaging (MRI), cf. [39], [34].

3D image segmentation aims at dividing a given image in connected regions, representing 3D objects in the image or their environment in the image domain Ω. For 3D image segmentation, the boundaries of the regions or objects have to be detected. These boundaries can be represented by a set of two-dimensional surfaces.

The active contours concept [20] can be extended to the three-dimensional case. For 3D images, we can consider time-dependent two-dimensional surfaces (*active*

---

[*]Deutsches Zentrum für Luft- und Raumfahrt (DLR), 82234 Weßling, Germany, email: heike.benninghoff@dlr.de.

[†]Fakultät für Mathematik, Universität Regensburg, 93040 Regensburg, Germany, email: harald.garcke@ur.de.

surfaces) and evolution laws for the surfaces which attract them to region boundaries. In particular, we will study extensions of the Mumford-Shah model [28] and the Chan-Vese model [16] to the three-dimensional case.

The articles [17], [18] belong to the first works, where the active contours model [20] is extended to volumetric image data. There, an analytical framework is introduced for 3D deformable surfaces. For practical computations however, the authors suggest to replace a given 3D image by a sequence of 2D images and to apply the 2D active contour model on each single image, followed by a 3D reconstruction of the surface.

The geodesic active contours model [13] is a popular edge-based method. An extension of the geodesic active contours model to 3D image segmentation is proposed in [14]. The level set method [29] is used to describe the surface implicitly.

The level set method is also used in [40], where 2D and 3D active contour models are presented and applied on medical images. However, practical results are only shown for 2D images. A detailed literature study on 3D brain cortex segmentation is given in [22]. A review on segmentation of medical X-ray computed tomography and magnetic resonance images is presented in [35].

In [27], a combination of edge-based and region-based segmentation methods is proposed. Both explicit (triangulated surfaces) and implicit (level set) methods are implemented. For explicit methods a constant global topology is assumed. For images which require topology changes, only the level set method is applied.

The Chan-Vese model [16] is used for 2D and 3D medical applications in [33] to perform heart segmentation using an iterative version of the Chan-Vese algorithm. The level set method with a finite difference scheme is used to solve the segmentation problem numerically. The level set method is also applied in [1] and [36] for active surfaces. Applications using 3D medical data (i.a. lung and heart segmentation) are considered. In [24], the level-set method is applied for 3D cell membrane segmentation. An approach for tracking cells in 4D images (3D data + time) has been developed in [25].

Also some parametric approaches for surface evolution exist in literature: In [11], a program called "The Surface Evolver" is presented which computes evolving triangulated surfaces, where the evolution is driven by energy minimization problems with possible constraints. The program is able to perform topology changes like splitting if this is instructed by the user.

Another explicit method for evolving surfaces by using triangulated surfaces is proposed in [12]. Techniques are introduced for mesh quality improvement and for topology changes. Splitting is done by a combination of remov-

ing degenerate elements and a certain mesh separation method based on duplication and separation of nodes. Surfaces are thus split if they become locally too thin. Merging is detected by searching for edges which are too close.

Finite element approaches for surface evolutions are pursued in [6]. There, the authors do not consider image segmentation applications, but surface diffusion. Several mesh quality routines like mesh regularization (keeping all angles of simplices at a node of the same size), time step control, refine/coarsening routines and angle width control are proposed.

In this paper, we present a novel parametric approach for 3D image segmentation. We present a scheme for image segmentation describing the evolution of parametric surfaces. For the numerical approximation, the smooth surfaces are replaced by triangulated surfaces. We make use of a parametric finite element scheme based on [8]. There, a scheme is proposed for surface diffusion, (inverse) mean curvature flow and non-linear flows. We use and apply this scheme to image segmentation with multiple phases and regions.

Our method also allows for topology changes which have not been addressed in [8]. We efficiently detect topology changes and perform modifications of the surface triangulations. In [10], we considered segmentation of two-dimensional images, and used and extended a method to detect topology changes [26] to handle a variety of topology changes of curves. Topology changes involving surfaces are more complex compared to topology changes involving curves. For example, if a curve splits up in two subcurves, the discretization has to be modified only at two points, see [10] for details. If a surface is split up in two subsurfaces, many triangles are located in a small volume. The pure detection of such a splitting is quite simple; the idea of an auxiliary background grid [26] as used for curves can be extended to topology changes of surfaces. However, the modifications of the surface triangulation are not as straight-forward as for curves. In case of splitting, we will delete the involved triangles near the splitting point resulting in two surfaces with intermediate holes. Then, we will close the intermediate holes by creating new triangles. Apart from splitting and merging, further topology changes can occur for surfaces: An increase or decrease of the genus of a surface can also occur, for example when a sphere evolves to a torus or vice versa. In summary, the execution of topology changes is the main additional challenge of 3D image segmentation with parametric surfaces. Therefore, one main focus of this paper is the detection, identification and execution of topology changes.

The remaining part of this paper is structured as fol-

lows. In Section 2, we present a region-based active surface model where we extend the Mumford-Shah model [28] and the Chan-Vese model [16] to 3D images. We present an efficient parametric scheme for the evolving surfaces. Also multiple phases can be handled. The main part of this paper is the numerical approximation of our scheme and the handling of topology changes which is described in Section 3. A finite element scheme is presented, a corresponding linear equation is derived and some computational details are given including mesh quality aspects and time step control. The detection, identification and execution of topology changes is described in detail including a description how to modify the triangulations after a topology change has been detected. In Section 4, we present results from segmentation of artificial test images and real medical images. We demonstrate the different topology changes which can occur during the evolution of surfaces. A final conclusion is drawn in Section 5.

2 Segmentation of three-dimensional images

2.1 Region-based Active Surfaces

We perform image segmentation by active surfaces, the surface-analogue to active contours [20], [17]. The idea of active surfaces is to let surfaces $\Gamma(t)$, $t \in [0, T]$, evolve in time such that a certain energy functional is minimized. From the minimization problem, one can derive an evolution law such that the surfaces of $\Gamma(t)$ are attracted to the region boundaries in the given image.

Let $\Omega \subset \mathbb{R}^3$ be open and bounded. We first consider a scalar image function $u_0 : \Omega \to \mathbb{R}$.

In this paper, we restrict on region-based methods for segmentation of 3D images because of advantages of region-based approaches compared to edge-based approaches (cf. Section 1).

The Mumford-Shah [28] method for 3D images aims at finding a set of two-dimensional surfaces $\Gamma = \Gamma_1 \cup \ldots \cup \Gamma_{N_C}$ and a piecewise smooth function $u : \Omega \to \mathbb{R}$ with possible discontinuities across Γ approximating the original image u_0. The energy to be minimized is

$$E^{\mathrm{MS}}(u, \Gamma) = \sigma|\Gamma| + \int_{\Omega \setminus \Gamma} \|\nabla u\|^2 \, \mathrm{d}x + \lambda \int_\Omega (u_0 - u)^2 \, \mathrm{d}x,$$
(1)

where $\sigma, \lambda > 0$ are weighting parameters and $|\Gamma|$ denotes the total area of the surfaces belonging to Γ. (For non-smooth surfaces, we identify $|\Gamma|$ with the two-dimensional Hausdorff measure of $\Gamma \subset \mathbb{R}^3$.)

The first term in (1) penalizes the area of the surfaces, the second term does not allow u to change much in $\Omega \setminus \Gamma$, and the third term requests that u is a good approximation of u_0.

We first consider two-phase image segmentation, we consider one closed, orientable surface Γ separating two disjoint regions Ω_1 and Ω_2 such that $\Omega = \Omega_1 \cup \Gamma \cup \Omega_2$. We assume that Γ is oriented by a unit normal vector field $\vec{\nu}$ pointing from Ω_2 to Ω_1.

Furthermore, we consider a piecewise constant version of the Mumford-Shah functional: We search for a surface Γ and for an approximation $u : \Omega \to \mathbb{R}$ of u_0 which is piecewise constant in each region, i.e. $u_{|\Omega_k} = c_k$, $k = 1, 2$, such that

$$E(\Gamma, c_1, c_2) =$$
$$= \sigma|\Gamma| + \lambda \left(\int_{\Omega_1} (u_0 - c_1)^2 \, \mathrm{d}x + \int_{\Omega_2} (u_0 - c_2)^2 \, \mathrm{d}x \right) \quad (2)$$

is minimized.

Similarly, the Chan-Vese functional [16] can be extended to 3D images. The energy to be minimized is

$$E(\Gamma, c_1, c_2) = \sigma|\Gamma| + \mu \int_{\Omega_1} 1 \, \mathrm{d}x + \lambda_1 \int_{\Omega_1} (u_0 - c_1)^2 \, \mathrm{d}x$$
$$+ \lambda_2 \int_{\Omega_2} (u_0 - c_2)^2 \, \mathrm{d}x, \quad (3)$$

where $\sigma, \lambda_1, \lambda_2 > 0$, $\mu \geq 0$ are weighting parameters. For $\mu = 0$ and $\lambda_1 = \lambda_2 = \lambda$, this is the functional (2).

The energy defined in (2) depends on the surface Γ and on the image approximation u given by the coefficients c_1, c_2. For minimizing (2), we perform a two-step approach:

First, we fix the surface Γ and consider variations in the coefficients c_1, c_2. Using the theory of calculus of variations we obtain the mean of the image function in Ω_k for c_k, $k = 1, 2$:

$$c_k = \frac{\int_{\Omega_k} u_0 \, \mathrm{d}x}{\int_{\Omega_k} 1 \, \mathrm{d}x}. \quad (4)$$

Then, we fix c_1 and c_2 and consider small variations of the surface Γ by smooth surfaces $\Gamma(t) \subset \Omega$, $t \in (-\epsilon, \epsilon)$, with $\Gamma(0) = \Gamma$. Let $\Omega_1(t)$ and $\Omega_2(t)$ be the regions separated by $\Gamma(t)$. We define

$$f(\vec{x}, c_1, c_2, t) := \begin{cases} (u_0(\vec{x}) - c_1)^2, & \text{if } \vec{x} \in \Omega_1(t), \\ (u_0(\vec{x}) - c_2)^2, & \text{if } \vec{x} \in \Omega_2(t), \end{cases} \quad (5)$$

which is defined for a.e. $\vec{x} \in \Omega$.

3

By using a transport theorem, we obtain

$$\left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{t=0} E(\Gamma(t), c_1, c_2) =$$

$$= \left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{t=0} \left(\sigma \int_{\Gamma(t)} 1 \, \mathrm{d}A + \lambda \int_{\Omega} f(\vec{x}, c_1, c_2, t) \, \mathrm{d}x \right)$$

$$= -\sigma \int_{\Gamma} \kappa \, V_n \, \mathrm{d}A +$$

$$\quad - \lambda \int_{\Gamma} \left((u_0 - c_1)^2 - (u_0 - c_2)^2 \right) V_n \, \mathrm{d}A$$

$$= - \int_{\Gamma} (\sigma \kappa + F) \, V_n \, \mathrm{d}A$$

where $\mathrm{d}A$ is the area element, V_n is the normal velocity, κ the mean curvature and F is an external force given by

$$F(\vec{x}) = \lambda \left((u_0(\vec{x}) - c_1)^2 - (u_0(\vec{x}) - c_2)^2 \right), \ \vec{x} \in \Gamma. \quad (6)$$

The fastest decrease of the energy is obtained for

$$V_n = \sigma \kappa + F. \quad (7)$$

Also multichannel images with a vector-valued image function $\vec{u}_0 : \Omega \to \mathbb{R}^d$ can be handled. This involves vector-valued coefficients $\vec{c}_k$, $k = 1, 2$, and a modification of the external force to, for example,

$$F(\vec{x}) = \sum_{i=1}^{d} \lambda_i \left[((u_0)_i(\vec{x}) - (c_1)_i)^2 - ((u_0)_i(\vec{x}) - (c_2)_i)^2 \right], \quad (8)$$

where the subscript i denotes the i-th component of a vector, $i = 1, \ldots, d$. For computation of the coefficients, each component of $\vec{c}_k$ is set to the mean of the corresponding component of $\vec{u}_0$ in the region Ω_k, $k = 1, 2$.

In principle, also spaces like the HSV (hue, saturation, value) or CB (chromaticity, brightness) space can be used [4], [15], [37]. In these cases, the image function has values on certain submanifolds of $\mathbb{R}^d$. In [10], we proposed a method to segment 2D images using the color space HSV and CB. The method can be transferred also to the 3D case. In many practical applications however, for example medical 3D image data generated by computed tomography (CT) or magnetic resonance imaging (MRT), the image function is often scalar-valued (cf. for example the lung image database of The Cancer Imaging Archive (TCIA) [31], [2], [30]).

2.2 Parametric and Multiphase Formulation

Equation (7) can be rewritten using a parametric approach to describe the time-dependent surfaces. Further, we now consider a more general setup of multiple surfaces $\Gamma_i(t)$, $t \in [0, T]$, $i = 1, \ldots, N_S$, which separate three-dimensional regions $\Omega_k(t)$, $k = 1, \ldots, N_R$. We assume that the surfaces are compact and oriented by unit normal vector fields $\vec{\nu}_i(\,.\,, t)$ pointing from $\Omega_{k^-(i)}(t)$ to $\Omega_{k^+(i)}(t)$, where $k^\pm(i) \in \{1, \ldots, N_R\}$.

Let $\vec{x}_i(\,.\,, t) : \Upsilon_i \to \mathbb{R}^3$, $i = 1, \ldots, N_S$, be a smooth parameterization of $\Gamma_i(t)$, where Υ_i is a two-dimensional reference manifold, for example the sphere $\Upsilon_i = S^2 \subset \mathbb{R}^3$. The normal velocity of $\Gamma_i(t)$ can be expressed as $(V_n)_i = (\vec{x}_i)_t \cdot \vec{\nu}_i$.

An approximation of the image intensity function u_0 is given by the piecewise constant function $u(\,.\,, t) = \sum_{k=1}^{N_R} c_k(t) \chi_{\Omega_k(t)}$, where $\chi_{\Omega_k(t)}$ is the characteristic function of $\Omega_k(t)$ and $c_k(t)$ is the mean of u_0 in $\Omega_k(t)$.

For each surface, we define the external forcing term

$$F_i(\,.\,, t) = \lambda \left((u_0 - c_{k^+(i)}(t))^2 - (u_0 - c_{k^-(i)}(t))^2 \right) \quad (9)$$

and obtain the following scheme for the surfaces: Find $\vec{x}_i(\,.\,, t) : \Upsilon_i \to \mathbb{R}^3$ and $\kappa_i(\,.\,, t) : \Upsilon_i \to \mathbb{R}$, $i = 1, \ldots, N_S$, satisfying

$$(\vec{x}_i)_t \cdot \vec{\nu}_i = \sigma \kappa_i + F_i, \quad (10\mathrm{a})$$

$$\Delta_\Gamma \vec{x}_i = \kappa_i \vec{\nu}_i. \quad (10\mathrm{b})$$

Equation (10a) is a parametric formulation of (7) for multiple regions. Equation (10b) relates the parametrization $\vec{x}_i$ and the curvature κ_i, see e.g. [19]. The symbol Δ_Γ denotes the Laplace-Beltrami operator. Here, we use a small abuse of notation, i.e. we consider κ_i and $\vec{\nu}_i$ as functions defined on Υ_i, i.e. we identify κ_i with $\kappa_i \circ \vec{x}_i$ and $\vec{\nu}_i$ with $\vec{\nu}_i \circ \vec{x}_i$, $i = 1, \ldots, N_S$.

3 Numerical approximation

3.1 Finite Element Approximation

We introduce a finite element approximation for the scheme (10) which is based on a scheme developed in [8], where geometric flows of two-dimensional surfaces are considered. We extend the ideas to solve schemes like (10) which arise in image segmentation applications.

Let $0 = t_0 < t_1 < \ldots < t_M = T$ be a decomposition of the time interval into possibly variable time steps $\tau_m = t_{m+1} - t_m$ for $m = 0, \ldots, M - 1$.

Let N_S denote the number of surfaces and N_R denote the number of regions. Let the smooth surface $\Gamma_i(t_m)$, $i = 1, \ldots, N_S$, be approximated by a polyhedral surface Γ_i^m of the form

$$\Gamma_i^m = \bigcup_{j=1}^{N_{i,F}} \overline{\sigma_{i,j}^m}, \quad (11)$$

4

where $\sigma_{i,j}^m$, $j = 1, \ldots, N_{i,F}$, are disjoint, open simplices (also called faces) with vertices $\vec{q}_{i,j}^m$, $j = 1, \ldots, N_{i,V}$. Further, let $h := \max_{i=1,\ldots,N_S, j=1,\ldots,N_{i,F}} \operatorname{diam}(\sigma_{i,j}^m)$ be the maximum diameter of a simplex of the triangulated surfaces. The diameter $\operatorname{diam}(\sigma_{i,j}^m)$ is defined as the maximum distance between two points of $\overline{\sigma_{i,j}^m}$.

Let $\vec{X}_i^m$ be a parameterization of Γ_i^m and let Ω_k^m, $k = 1, \ldots, N_R$, denote the open, disjoint subsets of Ω separated by Γ_i^m, $i = 1, \ldots, N_S$. Thus, Ω_k^m is an approximation of $\Omega_k(t_m)$ for $k = 1, \ldots, N_R$.

The new surfaces Γ_i^{m+1} are parameterized over Γ_i^m. Therefore, we define the following finite element spaces

$$W(\Gamma^m) := \{(\eta_1, \ldots, \eta_{N_S}) \in C(\Gamma_1^m, \mathbb{R}) \times \ldots \times$$
$$C(\Gamma_{N_S}^m, \mathbb{R}) : \eta_i|_{\sigma_{i,j}^m} \text{ is linear,}$$
$$\forall i = 1, \ldots, N_S, j = 1, \ldots, N_{i,F}\}, \quad (12a)$$

$$\underline{V}(\Gamma^m) := \{(\vec{\eta}_1, \ldots, \vec{\eta}_{N_S}) \in C(\Gamma_1^m, \mathbb{R}^3) \times \ldots \times$$
$$C(\Gamma_{N_S}^m, \mathbb{R}^3) : \vec{\eta}_i|_{\sigma_{i,j}^m} \text{ is linear,}$$
$$\forall i = 1, \ldots, N_S, j = 1, \ldots, N_{i,F}\}. \quad (12b)$$

The spaces $W(\Gamma^m)$ and $\underline{V}(\Gamma^m)$ thus consist of scalar or vector-valued, piecewise linear functions defined on Γ^m.

A basis of $W(\Gamma^m)$ is given by functions $\chi_{i,j}^m := ((\chi_{i,j}^m)_1, \ldots, (\chi_{i,j}^m)_{N_S}) \in W(\Gamma^m)$, where

$$(\chi_{i,j}^m)_k(\vec{q}_{k,l}^m) = \delta_{ik}\delta_{jl} \quad (13)$$

for $i, k = 1, \ldots, N_S$, $j = 1, \ldots, N_{i,V}$, $l = 1, \ldots, N_{k,V}$.

Note, that depending whether the domain of definition is Γ^{m-1} or Γ^m, we can interpret $\vec{X}^m$ as a different function. In particular we have for $m \geq 1$, $\vec{X}^m \in \underline{V}(\Gamma^{m-1})$, and for $m \geq 0$, $\vec{X}^m \in \underline{V}(\Gamma^m)$ is the identity defined on Γ^m.

For scalar functions $u = (u_1, \ldots, u_{N_S})$, $v = (v_1, \ldots, v_{N_S}) \in L^2(\Gamma_1^m, \mathbb{R}) \times \ldots \times L^2(\Gamma_{N_S}^m, \mathbb{R})$ and for vector-valued functions $u = (u_1, \ldots, u_{N_S})$, $v = (v_1, \ldots, v_{N_S}) \in L^2(\Gamma_1^m, \mathbb{R}^3) \times \ldots \times L^2(\Gamma_{N_S}^m, \mathbb{R}^3)$, we introduce the L^2-inner product over the current polyhedral surface Γ^m as follows:

$$\langle u, v \rangle_m := \int_{\Gamma^m} u \cdot v \, dA = \sum_{i=1}^{N_S} \int_{\Gamma_i^m} u_i \cdot v_i \, dA. \quad (14)$$

If u, v are piecewise continuous with possible jumps across the edges of $\sigma_{i,j}^m$, $i = 1, \ldots, N_S$, $j = 1, \ldots, N_{i,F}$, the mass lumped inner product is defined as

$$\langle u, v \rangle_m^h := \frac{1}{3} \sum_{i=1}^{N_S} \sum_{j=1}^{N_{i,F}} |\sigma_{i,j}^m| \sum_{l=1}^{3} (u \cdot v)((\vec{q}_{i,j_l}^m)^-), \quad (15)$$

where $\vec{q}_{i,j_l}^m$, $l = 1, 2, 3$, are the vertices of $\sigma_{i,j}^m$, $|\sigma_{i,j}^m| = \frac{1}{2}\|(\vec{q}_{i,j_2}^m - \vec{q}_{i,j_1}^m) \times (\vec{q}_{i,j_3}^m - \vec{q}_{i,j_1}^m)\|$ is the area of $\sigma_{i,j}^m$ and $u((\vec{q}_{i,j_l}^m)^-) := \lim_{\vec{p} \to \vec{q}_{i,j_l}^m, \vec{p} \in \sigma_{i,j}^m} u(\vec{p})$.

We assume that the vertices $\{\vec{q}_{i,j_l}^m\}_{l=1}^3$, $j = 1, \ldots, N_{i,F}$, are ordered such that the unit normal $\vec{\nu}_i^m$ at Γ_i^m is given by

$$\vec{\nu}_i^m|_{\sigma_{i,j}^m} := \vec{\nu}_{i,j}^m := \frac{(\vec{q}_{i,j_2}^m - \vec{q}_{i,j_1}^m) \times (\vec{q}_{i,j_3}^m - \vec{q}_{i,j_1}^m)}{\|(\vec{q}_{i,j_2}^m - \vec{q}_{i,j_1}^m) \times (\vec{q}_{i,j_3}^m - \vec{q}_{i,j_1}^m)\|} \quad (16)$$

points from $\Omega_{k^-(i)}^m$ to $\Omega_{k^+(i)}^m$.

We propose the following finite element scheme approximating the scheme (10): Let Γ^0 be a union of polyhedral surfaces approximating $\Gamma(0)$ and let $\vec{X}^0 \in \underline{V}(\Gamma^0)$ be the identity function on Γ^0. Find $\vec{X}^{m+1} \in \underline{V}(\Gamma^m)$ and $\kappa^{m+1} \in W(\Gamma^m)$, $m = 0, 1, \ldots, M-1$, such that

$$\langle \frac{\vec{X}^{m+1} - \vec{X}^m}{\tau_m}, \chi \, \vec{\nu}^m \rangle_m^h - \sigma \langle \kappa^{m+1}, \chi \rangle_m^h = \langle F^m, \chi \rangle_m^h,$$
$$\forall \chi \in W(\Gamma^m), \quad (17a)$$

$$\langle \kappa^{m+1} \, \vec{\nu}^m, \vec{\eta} \rangle_m^h + \langle \nabla_s \vec{X}^{m+1}, \nabla_s \vec{\eta} \rangle_m = 0,$$
$$\forall \vec{\eta} \in \underline{V}(\Gamma^m). \quad (17b)$$

Here, $F^m = (F_1^m, \ldots, F_{N_S}^m)$ is defined by

$$F_i^m(\vec{q}_{i,j}^m) := \lambda \left((u_0(\vec{X}_i^m(\vec{q}_{i,j}^m)) - c_{k^+(i)}^m)^2 + \right.$$
$$\left. - (u_0(\vec{X}_i^m(\vec{q}_{i,j}^m)) - c_{k^-(i)}^m)^2 \right),$$

for $i = 1, \ldots, N_S$ and $j = 1, \ldots, N_{i,V}$ and c_k^m is set to the mean of u_0 in Ω_k^m for $k = 1, \ldots, N_R$.

We further introduce a weighted normal defined at the nodes $\vec{X}_i^m(\vec{q}_{i,j}^m) = \vec{q}_{i,j}^m \in \Gamma_i^m$ by setting

$$\vec{\omega}_i^m(\vec{q}_{i,j}^m) := \vec{\omega}_{i,j}^m := \frac{1}{|\Lambda_{i,j}^m|} \sum_{\sigma_{i,l}^m \in \mathcal{T}_{i,j}^m} |\sigma_{i,l}^m| \, \vec{\nu}_{i,l}^m, \quad (18)$$

where for $i = 1, \ldots, N_S$ and $j = 1, \ldots, N_{i,V}$, $\mathcal{T}_{i,j}^m := \{\sigma_{i,l}^m : \vec{q}_{i,j}^m \in \overline{\sigma_{i,l}^m}\}$ and $\Lambda_{i,j}^m := \bigcup_{\sigma_{i,l}^m \in \mathcal{T}_{i,j}^m} \overline{\sigma_{i,l}^m}$.

Further, we set $\vec{v}_i^m(\vec{q}_{i,j}^m) := \vec{v}_{i,j}^m := \vec{\omega}_{i,j}^m / \|\vec{\omega}_{i,j}^m\|$ and $\vec{\omega}^m = (\vec{\omega}_1^m, \ldots, \vec{\omega}_{N_S}^m)$ and $\vec{v}^m = (\vec{v}_1^m, \ldots, \vec{v}_{N_S}^m)$.

As in [7], [8], we make a very mild assumption on the triangulations:

$(\mathcal{A})$ For $m = 0, \ldots, M$, we assume that $|\sigma_{i,j}^m| > 0$ for all $i = 1, \ldots, N_S$ and $j = 1, \ldots, N_{i,F}$ and for $m = 0, \ldots, M-1$, we assume that $\dim \operatorname{span} \{\vec{\omega}_{i,j}^m\}_{j=1}^{N_{i,V}} = 3$.

The assumption ($\mathcal{A}$) is only violated in very rare cases. For closed surfaces without self intersections, it always holds.

Theorem 1. *Let the assumption ($\mathcal{A}$) hold. Then there exists a unique solution $\left\{\vec{X}^{m+1}, \kappa^{m+1}\right\} \in \underline{V}(\Gamma^m) \times W(\Gamma^m)$ to the system* (17).

Proof. (See also [8].) Since the system is linear, it is sufficient to show uniqueness. Therefore, we consider the following scheme: Find $\vec{X} \in \underline{V}(\Gamma^m)$ and $\kappa \in W(\Gamma^m)$ such that

$$-\frac{1}{\tau_m}\langle \vec{X}, \chi \, \vec{\nu}^m \rangle_m^h + \sigma \langle \kappa, \chi \rangle_m^h = 0, \quad \forall \chi \in W(\Gamma^m), \tag{19a}$$

$$\langle \kappa \, \vec{\nu}^m, \vec{\eta} \rangle_m^h + \langle \nabla_s \vec{X}, \nabla_s \vec{\eta} \rangle_m = 0, \quad \forall \vec{\eta} \in \underline{V}(\Gamma^m), \tag{19b}$$

holds. Testing (19a) with $\chi = \kappa$ and (19b) with $\vec{\eta} = \vec{X}$ leads to

$$\sigma \tau_m \langle \kappa, \kappa \rangle_m^h + \langle \nabla_s \vec{X}, \nabla_s \vec{X} \rangle_m = 0. \tag{20}$$

It follows that $\kappa_{i,j} = 0$ and $\vec{X}_{i,j} = \vec{C}_i \in \mathbb{R}^3$ for $i = 1, \ldots, N_S$, $j = 1, \ldots, N_{i,V}$. Inserting $\kappa = 0$ and $\vec{X} = \vec{C} = (\vec{C}_1, \ldots, \vec{C}_{N_S})$ in (19a) results in

$$\langle \vec{C}, \chi \, \vec{\nu}^m \rangle_m^h = 0, \quad \forall \chi \in W(\Gamma^m). \tag{21}$$

Choosing $\chi = \chi_{i,j}^m$ (the standard basis) and using (18), the definition of $\vec{\omega}_{i,j}^m$, results in

$$\vec{C}_i \cdot \vec{\omega}_{i,j}^m = 0, \quad \forall i = 1, \ldots, N_S, j = 1, \ldots, N_{i,V}. \tag{22}$$

Finally, using the assumption ($\mathcal{A}$), it follows that $\vec{C}_i = 0$ for each $i = 1, \ldots, N_S$. $\qquad \square \qquad \square$

3.2 Solution of the Discrete System

We define $\delta \vec{X}^{m+1} := \vec{X}^{m+1} - \vec{X}^m$. As $\delta \vec{X}^{m+1}$ and κ^{m+1} are uniquely given by their values at the nodes $\vec{q}_{i,j}^m$, we consider them as elements in $(\mathbb{R}^3)^N$ and $\mathbb{R}^N$, respectively, where $N = \sum_{i=1}^{N_S} N_{i,V}$. We introduce the matrices $M_m \in \mathbb{R}^{N \times N}$, $\vec{N}_m \in (\mathbb{R}^3)^{N \times N}$ and $\vec{A}_m \in (\mathbb{R}^{3 \times 3})^{N \times N}$ by

$$M_m := \begin{pmatrix} M_m^1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & M_m^{N_S} \end{pmatrix},$$

$$\vec{N}_m := \begin{pmatrix} \vec{N}_m^1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \vec{N}_m^{N_S} \end{pmatrix},$$

$$\vec{A}_m := \begin{pmatrix} \vec{A}_m^1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \vec{A}_m^{N_S} \end{pmatrix},$$

where $M_m^i \in \mathbb{R}^{N_{i,V} \times N_{i,V}}$, $\vec{N}_m^i \in (\mathbb{R}^3)^{N_{i,V} \times N_{i,V}}$, $\vec{A}_m^i \in (\mathbb{R}^{3 \times 3})^{N_{i,V} \times N_{i,V}}$, $i = 1, \ldots, N_S$. Their entries are defined by

$$[M_m^i]_{kl} := \langle \chi_{i,k}^m, \chi_{i,l}^m \rangle_m^h,$$
$$[\vec{N}_m^i]_{kl} := \langle \chi_{i,k}^m, \chi_{i,l}^m \vec{\nu}^m \rangle_m^h, \tag{23}$$
$$[\vec{A}_m^i]_{kl} := \langle \nabla_s \chi_{i,k}^m, \nabla_s \chi_{i,l}^m \rangle_m \, \vec{\mathrm{Id}}_3,$$

with $i = 1, \ldots, N_S$, $k, l = 1, \ldots, N_{i,V}$. Here, $\vec{\mathrm{Id}}_3$ denotes the identity matrix in $\mathbb{R}^{3 \times 3}$. Further, we introduce $b_m = (b_m^1, \ldots, b_m^{N_S}) \in \mathbb{R}^N$ defined by

$$[b_m^i]_k := \langle F_i^m, \chi_{i,k}^m \rangle_m^h, \quad i = 1, \ldots, N_S, \, k = 1, \ldots, N_{i,V}. \tag{24}$$

The scheme (17) can be rewritten to the following problem: Let Γ^0 be a polyhedral approximation of $\Gamma(0)$ and let $\vec{X}^0 = (\vec{X}_1^0, \ldots, \vec{X}_{N_S}^0) \in (\mathbb{R}^3)^N$ with $\vec{X}_i^0 = (\vec{X}_{i,1}^0, \ldots, \vec{X}_{i,N_{i,V}}^0)$ such that $\vec{X}_{i,j}^0$ are the coordinates of the vertices of Γ_i^0 for $i = 1, \ldots, N_S$, $j = 1, \ldots, N_{i,V}$. For $m = 0, \ldots, M - 1$ find $\delta \vec{X}^{m+1} \in (\mathbb{R}^3)^N$ and $\kappa^{m+1} \in \mathbb{R}^N$ such that

$$\begin{pmatrix} \sigma \tau_m M_m & -\vec{N}_m^T \\ \vec{N}_m & \vec{A}_m \end{pmatrix} \begin{pmatrix} \kappa^{m+1} \\ \delta \vec{X}^{m+1} \end{pmatrix} = \begin{pmatrix} -\tau_m b_m \\ -\vec{A}_m \vec{X}^m \end{pmatrix}. \tag{25}$$

Applying a Schur complement approach, we can transform this system to

$$\kappa^{m+1} = \frac{1}{\sigma} M_m^{-1} \left(\frac{1}{\tau_m} \vec{N}_m^T \delta \vec{X}^{m+1} - b_m \right), \tag{26a}$$

$$\left(\frac{1}{\sigma \tau_m} \vec{N}_m M_m^{-1} \vec{N}_m^T + \vec{A}_m \right) \delta \vec{X}^{m+1} =$$
$$= -\vec{A}_m \vec{X}^m + \frac{1}{\sigma} \vec{N}_m M_m^{-1} b_m. \tag{26b}$$

Since the system matrix in (26b) is symmetric and positive definite under the assumption ($\mathcal{A}$), there exists a unique solution.

The linear system (26b) can be solved with an iterative solver, for example, with the method of conjugate gradients with possible preconditioning, or with a direct solver for sparse matrices. For the experiments and examples presented in Section 4 of this paper, we use a MATLAB built-in routine, a direct solver for sparse systems. Even for two-dimensional problems, which result from the evolution of two-dimensional surfaces, the sparse direct solver is very efficient from a computational view.

3.3 Topology Changes

Parametric methods cannot handle topology changes automatically in contrast to other numerical methods like the level set method. During the evolution of surfaces singularities can occur like a pinch-off, see [6], [8]. In order to proceed after a pinch-off, the surface has to be split in two single surfaces. Other possible topology changes, that we will consider here, are merging of two surfaces and change of the genus (occurs for example during an evolution of a torus to a sphere or vice versa).

In [26], an algorithm is proposed to efficiently detect splitting and merging of evolving curves in $\mathbb{R}^2$, see also [5]. In [10], we used and extended this algorithm to detect topology changes in 2D images. In this paper, we want to adapt this approach to the 3D case. We aim at detecting topology changes which could occur during the evolution of surfaces. Having found the location where a topology change occurs, we propose a method how to modify the triangulations.

3.3.1 Detection of a Topology Change

In [26], a virtual, auxiliary 2D background grid is constructed which covers a two-dimensional domain, and topology changes of curves are detected if node points from different curves or different parts of one curve are located in one array of the background grid.

Motivated by this method for evolving curves, we propose the following method to detect topology changes of evolving surfaces. The basic idea is the use of a uniform 3D grid of cubes. A topology change may occur, if a large number of nodes or if nodes of different surfaces or different parts of one surface (with opposite normal vector) are located in one cube.

In detail, to detect a change in topology, we construct a uniform 3D background grid which covers the image domain Ω. In the following, we assume that Ω is a cuboid. If the 3D image u_0 is not given on a cuboid volume, we consider a cuboid which contains Ω.

Let $\Omega = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \times [z_{\min}, z_{\max}] \subset \mathbb{R}^3$ be the image domain containing in particular the surfaces Γ_i^m, $i = 1, \ldots, N_S$. We consider a grid dividing Ω in a set of many small cubes of edge width $a \in \mathbb{R}$. Let the grid consist of $N_x \times N_y \times N_z$ cubes, where $N_x = \mathrm{ceil}((x_{\max} - x_{\min})/a)$, $N_y = \mathrm{ceil}((y_{\max} - y_{\min})/a)$ and $N_z = \mathrm{ceil}((z_{\max} - z_{\min})/a)$.

We now perform one loop over all surfaces and nodes $\vec{X}_{i,j}^m$, $i = 1, \ldots, N_S$, $j = 1, \ldots, N_{i,V}$. If a node $\vec{X}_{i,j}^m$ is the first node which is detected to lie in a certain cube, we create a new list for that cube, where we store the index pair (i, j). If another node has already been identified to be located inside that cube, we add the index pair (i, j) to the existing list.

If there is a large number of nodes located in a cube, i.e. more than N_{detect} nodes, a topology change likely occurs, and the cube is stored in a list for possible topology changes. It is also possible that the node density is only locally very high at this location, but no topology change happens.

If there are less than N_{detect} nodes in the current list, we compare the surface index and the direction of the weighted normal vector of the current node with those of the nodes already stored in the list. If two different surface indices i_1 and i_2 occur or if two nodes with (nearly) opposite weighted normal vector are located in one cube, a topology change likely happens. The corresponding cube is accordingly stored in a list for topology changes.

After having considered all nodes, the cubes marked for topology changes are considered one by one. If a topology change is identified, the surface triangulation is accordingly changed. Thus, by successively considering all marked cubes, more than one topology change can be executed in one time step. The list can be optionally sorted such that the cube with the largest number of nodes is considered first.

The detection of topology changes is very efficient from a computational view. The effort is of order $\mathcal{O}(N)$, where N is the total number of node points. For comparison, a simple approach, where all possible pairs of two nodes are considered and where the distance between two nodes is computed to detect a topology change, would result in a computational effort of order $\mathcal{O}(N^2)$.

Similar as described in [10], the grid size a can be adaptively set, for example dependent on the speed of the evolving surfaces. Therefore the method to detect topology changes is both efficient and robust.

3.3.2 Identification of the Topology Change

For identifying which kind of topology change occurs, the nodes of the affected cube, i.e. the current cube of the sorted list, and the nodes of up to 26 neighbor cubes (in total up to 27 cubes, i.e. $3 \times 3 \times 3$ cubes) are considered. Let $S = \{j_1, \ldots, j_{n_c}\}$ denote the index set of the nodes and let $\vec{X}_j$, $j \in S$, denote the coordinates of the nodes located in the cubes. Further, let $\vec{\omega}_j$, $j \in S$, denote the corresponding weighted normal vectors at $\vec{X}_j$, recall their definition in (18). For the ease of illustration, we omit the time dependency (time index m) in the notation.

The different topology changes are distinguished by considering the weighted normal vectors $\vec{\omega}_j$. The idea is that in case of merging and in case of increasing genus, their are two main group of nodes which can be found

by considering their normal vectors. For splitting and decrease of genus, there are more than two main directions.

The node with index $j = j_1$ is set as representative of the first group. We choose thresholds $thr1 < thr2$, for example $thr1 = 20°$, $thr2 = 160°$. For $j = j_2, \ldots, j_{n_c}$ we consider the angle α between $\vec{\omega}_{j_1}$ and $\vec{\omega}_j$. If $\alpha < thr1$, the node j belongs to the first group. If $\alpha > thr2$ and if the second group is empty, the node j becomes the representative node of the second group. If the second group is not empty, we consider the angle β between $\vec{\omega}_j$ and $\vec{\omega}_{k_0}$, where $k_0 \in \{j_2, \ldots, j_{n_c}\}$ is the representative of the second group. If $\beta < thr1$, j is added to the second group.

For the next search we replace $\vec{\omega}_{j_1}$ and $\vec{\omega}_{k_0}$ by the average normal vectors $\vec{n}_1$ and $\vec{n}_2$ of group 1 and group 2, respectively, and re-consider the nodes which could not have been assigned to one group in the first step. If the angle between $\vec{\omega}_j$ and $\vec{n}_1$ or $\vec{\omega}_j$ and $\vec{n}_2$ is smaller than $thr1$, the node $\vec{X}_j$ is added to the corresponding group.

If group 2 is empty, or if one of the groups consists of only a small number of nodes (e.g. $< 5\%$ of n_c), we start again by using another node as representative for the first group (e.g. $j = j_2$), since the node $j = j_1$ could be an outlier. If no start node can be found, such that there exist two groups of nodes as described above, no topology change takes place. This can happen, if all weighted normals point in nearly the same direction.

The method such provides that a topology change like splitting is not wrongly detected at locations where several nodes are just close to each other but their normal vectors point in the same direction. From a mesh quality point of view, such meshes should be avoided; the nodes should be distributed equally over the surface. However, the detection of topology changes should be robust enough not to wrongly detect a splitting at locations where there is just a high density of nodes.

If both groups have a sufficient number of nodes, we proceed by considering the remaining normal vectors which could have not been assigned to one of the two groups. We again consider the angle between $\vec{\omega}_j$ and $\vec{n}_1$ and $\vec{\omega}_j$ and $\vec{n}_2$. If both angles are $> thr3$ (e.g. $thr3 = 40°$), the normal $\vec{\omega}_j$ points in a complete different direction compared to $\vec{n}_1$ and $\vec{n}_2$. Let N_0 be the number of such points. If N_0 exceed a predefined number (e.g. $1/3\, n_c$), then there are more than two main groups of directions. In this case, there is a splitting or decrease of genus. Splitting and decrease of genus are handled similarly (see below for details how to modify the triangulations). Triangles close to the detected cube are deleted. If the remaining triangulation of the former surface consists of two connected components, a split-

ting occurs. If the remaining triangulation is connected, a decrease of genus occurs.

If N_0 is zero or if it does not exceed the predefined number, possible remaining normal vectors are only single outliers and there are only two main groups of normal vectors with nearly opposed normal vector. In this case, there is a merging or increase of genus. If there are nodes belonging to two different surfaces, a merging occurs. Otherwise, an increase of genus occurs.

An illustration of the algorithm to detect and identify topology changes is given in Figure 1.

3.3.3 Algorithm for Splitting and Decreasing of Genus

We propose the following algorithm for a possible modification of the surface triangulation after the detection of a splitting or decrease of genus.

- **Preparation and deletion of simplices:** In case of splitting or decreasing genus, we consider the set of affected nodes $\vec{X}_j$, $j \in S$, which are located in the cube or in a neighbor cube, where the topology change has been detected. Let $\vec{p}_E$ be the mean of the points $\{\vec{X}_j : j \in S\}$. We delete all simplices with at least one vertex belonging to the set $\{\vec{X}_j : j \in S\}$. When deleting one simplex, we change the neighbor information of neighbor simplices at the corresponding edges to -1 (free edges). Deletion creates two temporary holes in the surface(s). Simplices with two or three free edges are deleted as well, see Figure 2. As a result we either have two sets of connected simplices ($\to$ splitting) or one set of connected simplices ($\to$ decrease of genus).

- **Set surface index (splitting only):** The remaining simplices form two connected sets. For one set, we need to re-set the surface index. Let i be the surface index of the original surface. By splitting the total number of surfaces is increased to $N_S + 1$. Starting with one simplex with a free edge, we re-set its surface index from i to $N_S + 1$. Then, we consider its neighbor simplices and assign them to surface $N_S + 1$ also. By this procedure, the simplices of one of the two connected components are assigned one by one to the surface $N_S + 1$ by heritage, i.e. by use of neighbor information.

- **Generate new simplices:** We close each of the two intermediate holes (where simplices have been deleted) by constructing new simplices at edges of simplices where the neighbor information has been set to -1. First we create two new points, each with
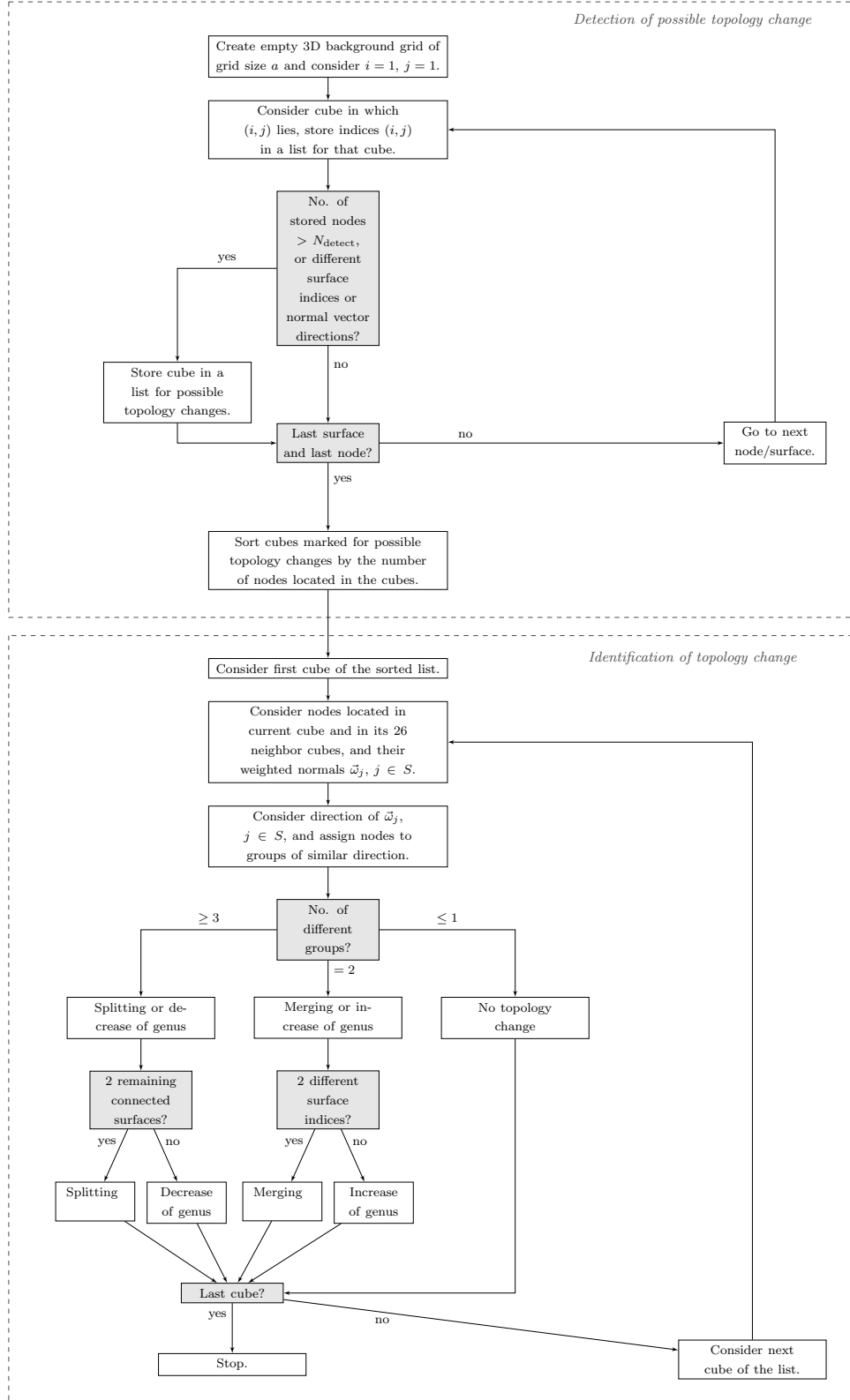
Create empty 3D background grid of grid size a and consider $i = 1$, $j = 1$.

Consider cube in which (i, j) lies, store indices (i, j) in a list for that cube.

No. of stored nodes $> N_{\text{detect}}$, or different surface indices or normal vector directions?

yes

Store cube in a list for possible topology changes.

no

Last surface and last node?

no

Go to next node/surface.

yes

Sort cubes marked for possible topology changes by the number of nodes located in the cubes.

Consider first cube of the sorted list.

Consider nodes located in current cube and in its 26 neighbor cubes, and their weighted normals $\vec{\omega}_j$, $j \in S$.

Consider direction of $\vec{\omega}_j$, $j \in S$, and assign nodes to groups of similar direction.

No. of different groups?

≥ 3

≤ 1

$= 2$

Splitting or decrease of genus

Merging or increase of genus

No topology change

2 remaining connected surfaces?

2 different surface indices?

yes   no

yes   no

Splitting

Decrease of genus

Merging

Increase of genus

Last cube?

yes

no

Stop.

Consider next cube of the list.

Figure 1: Illustration of the detection and identification of topology changes of surfaces.

9
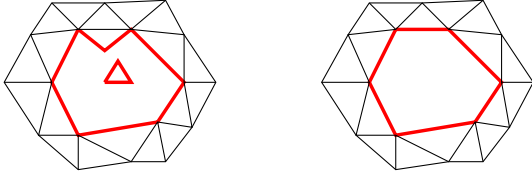
Figure 2: Left: Part of the surface near a temporary hole. Free edges are drawn with red color. Right: Surface near the hole after deletion of simplices with more than one free edge.
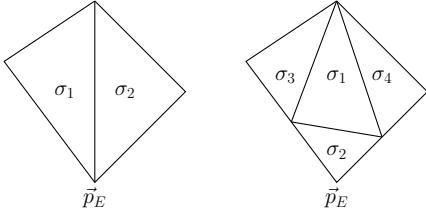
Figure 3: Improving mesh quality after a splitting or decrease of genus - Decrease the number of edges at the node $\vec{p}_E$.

coordinates $\vec{p}_E$, one for each intermediate hole. In the next time steps the two nodes can move away from each other. If σ is a simplex with a free edge given by $\vec{X}_{\sigma,j_1}$ and $\vec{X}_{\sigma,j_2}$ with no neighbor simplex, a new simplex is generated given by the vertices $\vec{X}_{\sigma,j_1}$, $\vec{X}_{\sigma,j_2}$ and one of the two new nodes at $\vec{p}_E$. The new simplex inherits the surface index from σ.

- **Improve mesh quality:** By simply connecting all free edges with one of the two new vertices at $\vec{p}_E$, the two vertices can belong to a big number of simplices. Let σ_1 and σ_2 be two of the newly generated simplices with one common edge. We construct 4 new simplices from σ_1 and σ_2 such that the new vertex belongs to only one of the 4 new simplices, see Figure 3. Therefore the number of elements to which the vertices at $\vec{p}_E$ belong to is approximately halved. We repeat this procedure until each of the newly created nodes at $\vec{p}_E$ belongs to ≤ 8 elements.

3.3.4 Algorithm for Merging and Increasing of Genus

We propose the following algorithm for a possible modification of the surface triangulation after the detection of a merging or increase of genus.

- **Preparation and deletion of simplices:** In case of merging or increasing genus, we consider the set of affected nodes $\vec{X}_j$, $j \in S$, which are located in the cube or in a neighbor cube, where the topology change has been detected. We delete all simplices with at least one vertex belonging to the set $\{\vec{X}_j : j \in S\}$. This generates temporary free edges, i.e. simplices exist which do not have a neighbor simplex at that edge. The neighbor index corresponding to the free edge is set to -1. This creates two intermediate holes. Simplices with two or three free edges are deleted as well, see Figure 2.

- **Matching free nodes/edges:** There exist now two sets of connected free edges. Let $I_{free,k}$, $k = 1, 2$, be the set of nodes corresponding to the free edges (end points of the edges). We try to match the nodes of $I_{free,1}$ with the nodes of $I_{free,2}$ using the Hungarian method [21] which is a combinatorial optimization algorithm. The Euclidean distance is used as cost criterion for matching two nodes. Since the number of nodes of the two sets need not be equal, there can be nodes which could not have been matched in the first step, see Figure 4 (top). Therefore, new nodes are created by bisection of simplices at a free edge. Finally, each node can be matched, see Figure 4 (bottom).

- **Point/Edge merging:** Since two matched nodes can have slightly different coordinates, they are replaced by one node in the middle of the line connecting the two nodes. The free edges of the two open holes are merged by identifying the matched nodes. The simplices, nodes and edges administration needs to be adapted. The nodes in $I_{free,1}$ are updated; each point is replaced by the mid point between it and its matching partner. The nodes belonging to $I_{free,2}$ are deleted. Half of the free edges are deleted. The edge, node and neighbor information of the simplices at the former holes need to be adapted. In case of merging, the surface index of all simplices belonging to the second surface is set to the surface index of the first surface.

Note, that local refinement after a merging is typically necessary. This is automatically done, by a refinement method described in Section 3.4. If the surface grows locally near the former merging part, the simplices will become greater compared to the average simplex of the surface. In this case, the large simplices will be refined.

The idea of creating two intermediate open holes and merging the two surfaces there is based on [12]. There however, each hole is restricted to consist of exactly four
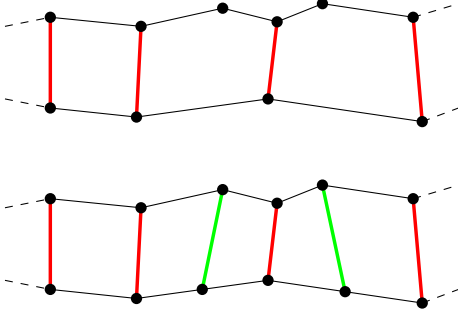
Figure 4: A subset of the free nodes. Top: Intermediate matching (red lines mark matching pairs). Bottom: Complete matching after inserting new nodes (red and green lines mark matching pairs).

free edges. New triangles between the free edges are created instead of merging the edges.

In our method, the seeking for close points (and therefore close edges/simplices) is very efficient, since we make use of a background grid motivated by the method presented in [26]. We extended this method originally intended for curves in the plane to topology changes of surfaces. We allow for intermediate holes with an arbitrary number of free edges. The hole size is of the magnitude of the grid size.

3.4 Additional Computational Aspects

3.4.1 Computations of regions and coefficients

The computation of regions Ω_k^m and coefficients c_k^m, the mean of u_0 in Ω_k^m, is done as follows: We assign each voxel of the three-dimensional image domain to a phase Ω_k^m. If a voxel is truncated by a surface, it is assigned to the phase to which the largest part belongs or to any of the two regions in case of two equal parts. Let S_k^m be the set of n_k^m voxels belonging to Ω_k^m. Then the approximation c_k^m is set to

$$c_k^m := \frac{C_k^m}{n_k^m}, \qquad C_k^m := \sum_{vox \in S_k^m} u_0|_{vox}. \qquad (27)$$

The entire image domain needs to be considered only for $m = 0$. For $m > 0$, we only locally update the regions and re-compute the coefficients on this basis. For that, we consider a small band/tube of voxels around the current surfaces and look for changes of the region assignment.

As the normal $\vec{\nu}_i^m$ points from $\Omega_{k^-(i)}^m$ to $\Omega_{k^+(i)}^m$, the voxels close to the surface Γ_i^m can be assigned to the phase $k^+(i)$ or $k^-(i)$, respectively.

In the update step, we first set $n_k^m = n_k^{m-1}$ and $C_k^m = C_k^{m-1}$ for $k = 1, \ldots, N_R$. For $i = 1, \ldots, N_S$, all voxels in an environment of Γ_i^m are subsequently considered. Let a voxel vox be assigned to phase $k \in \{k^+(i), k^-(i)\}$ and let $l \neq k$ be the former phase index of the voxel. Then, we set

$$n_k^m = n_k^m + 1, \quad n_l^m = n_l^m - 1,$$
$$C_k^m = C_k^m + u_0|_{vox}, \quad C_l^m = C_l^m - u_0|_{vox}. \qquad (28)$$

After having considered all voxels close to the surfaces, the coefficients are set to $c_k^m = C_k^m/n_k^m$ for $k = 1, \ldots, N_R$.

3.4.2 Time Step Control

We use a certain adaptive time step setting to control the speed of the evolution of the surface(s). Let $\Delta t = \tau_m$ denote the (possibly variable) time step size. The time step size is controlled as follows: Let $\delta X_n^{\min} > 0$, $\delta X_n^{\max} > 0$ with $\delta X_n^{\min} < \delta X_n^{\max}$ be user-defined tolerances for the absolute value of the position difference in normal direction. Let $\Delta t > 0$ be an initial time step size for $m = 0$ or the time step size of the previous time step for $m > 0$.

We propose the following time step size control: Choose a factor $\lambda_t \in \mathbb{N}$ (for example $\lambda_t = 2$ or $\lambda_t = 10$).

1. Solve equation (26b) and set δX_n^{m+1} to the maximum of $|\delta \vec{X}_{i,j}^{m+1} \cdot \vec{\omega}_{i,j}^m|$ for $i = 1, \ldots, N_S$ and $j = 1, \ldots, N_{i,V}$.

2. If $\delta X_n^{m+1} > \delta X_n^{\max}$, set Δt to $\frac{1}{\lambda_t} \Delta t$ and repeat step (i).

3. Otherwise, if $\delta X_n^{m+1} < \delta X_n^{\min}$, set Δt to $\lambda_t \Delta t$ and repeat step (i).

4. Otherwise, proceed by checking for topology changes (see above) and go to the next time step, i.e. set m to $m + 1$.

The effect of this time step size control is simple: If there are too high changes in the position of the nodes in normal direction (i.e. if the normal velocity is too high), the time step size will be decreased. This occurs if the sum of weighted curvature and external term is high. If the change in the position in normal direction is too small, the time step size will be increased to speed up the image segmentation process.

3.4.3 Mesh Quality Aspects

During the evolution of surface, it may be necessary to control the mesh quality. For example, if a surface continuously grows, the simplices become larger and should

be refined if their area exceeds a certain threshold. Similarly, too small simplices should be deleted.

For computing the matrix entries, cf. (23), we already need to compute the area of each simplex of the triangulation of the surface Γ_i^m, $i \in \{1, \ldots, N_S\}$. Let $A_{\text{desired}} > 0$ be a predefined desired area for one simplex. Let $a > 0$ be a given factor (e.g. $a = 2$ or $a = 10$). If the area of a simplex exceeds aA_{desired}, it will be refined by bisection of its largest edge. Its neighbor simplex across the refinement edge will be also refined such that no hanging nodes remain.

Local refinement is necessary to avoid too large simplices. Further, a mesh can also be continuously refined, for example when one starts with a small surface which globally grows. The triangles of the growing surface are refined one by one. Furthermore, the triangles which have one very large angle, i.e. an angle larger than a given threshold (e.g. $\geq 160°$), are also refined.

If a simplex area is smaller than a certain percentage of the desired area A_{desired}, for example smaller than 1%, the simplex is deleted. Further, it is also deleted if one of its three inner angles is smaller than a given threshold, for example smaller than 2°. When a simplex is marked for deletion, one or more neighbor simplices are also deleted.

Mesh operations like deletion of triangles are rarely necessary. These operations are usually performed only a few times, for example close before or after topology changes.

Figure 5 illustrates examples how the triangulation is adapted close to simplices which are marked for refinement or deletion.

Our numerical method for surface evolution for image segmentation tasks is based on a numerical method developed in [8]. This method provides a good mesh quality in many cases. However, if, for example, a pinch-off occurs, the mesh can get distorted and some routine for keeping a good mesh quality is needed.

The idea of a mesh regularization method proposed in [9] is to induce or reduce the tangential motion of nodes along a surface. We use this method to control the tangential motion of nodes of surfaces during 3D image segmentation. For details, we refer to [9]. The system (17) is replaced by a scheme which controls also the tangential motion of the nodes.

3.5 Summary of the Image Segmentation Algorithm

In summing up, we propose the following algorithm for segmentation of 3D images. Given a set of triangulated surfaces $\Gamma^0 = (\Gamma_1^0, \ldots, \Gamma_{N_S}^0)$ and nodes $\vec{X}_{i,j}^0$,
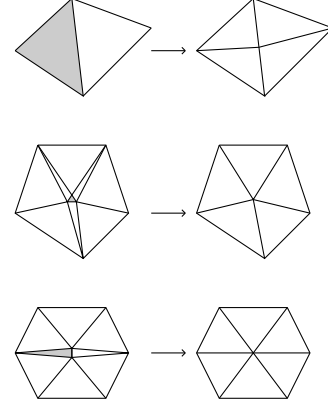

Figure 5: Top: Refinement of a simplex (marked in gray). The neighbor simplex is also refined to avoid hanging nodes. Center: Deletion of a simplex with a too small area (marked in gray). Three neighbor simplices are also deleted. Bottom: Deletion of a simplex with a too small angle (marked in gray). One neighbor simplex is also deleted.

$i = 1, \ldots, N_S$, $j = 1, \ldots, N_{i,V}$, perform the following steps for $m = 0, 1, \ldots, M - 1$:

1. Compute the regions Ω_k^m and the coefficients c_k^m, $k = 1, \ldots, N_R$, as described in Section 3.4.1.

2. Compute b^m as defined in (24) by using the coefficients c_k^m of step 1. Compute $\vec{X}^{m+1} = \vec{X}^m + \delta\vec{X}^{m+1}$ by solving the linear equation (26b).

3. Check whether the time step size needs to be increased or decreased, see Section 3.4.2. If the time step size needs to be changed, repeat step 2 with the new time step size.

4. Check whether topology changes occur and execute the topology change, see Section 3.3.

5. If necessary, refine too large simplices or delete too small simplices of the triangulation as described in Section 3.4.3.

4   Results

4.1   Artificial Test Images

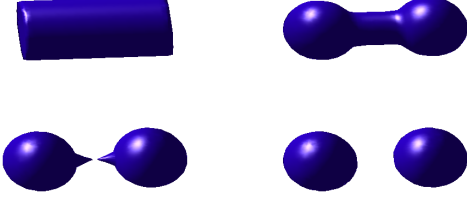In this section we demonstrate the developed method for segmentation of 3D images with parametric active sur-

Figure 6: Splitting of a surface during 3D image segmentation. Surface(s) at step $m = 0, 100, 215, 300$ at time $t_m = 0, 0.01, 0.0215, 0.0298$.

Figure 7: Merging demonstration. Surface(s) at step $m = 0, 50, 80, 200$ at time $t_m = 0, 0.005, 0.008, 0.02$.

faces. We first study four examples of artificial test images to demonstrate the ability of the method to detect different topology changes (splitting, merging, increase and decrease of genus).

In the first experiment, we demonstrate how a surface is split in two surfaces. We consider an artificial image defined on an image domain given given by the cuboid $\Omega = [-2.5, 2.5] \times [-1.5, 1.5] \times [-1.5, 1.5]$. The image intensity function is defined by

$$u_0 : \Omega \to \mathbb{R}, \quad u_0(\vec{x}) = \begin{cases} 0 & \text{if } \|\vec{x} - (-1.2, 0, 0)^T\| \leq 0.8 \\ & \lor \|\vec{x} - (1.2, 0, 0)^T\| \leq 0.8, \\ 1 & \text{else.} \end{cases}$$

The three-dimensional image contains two balls centered at $(\pm 1.2, 0, 0)^T \in \mathbb{R}^3$ with radius 0.8. The segmentation process is started using a cylinder-like surface as initial surface placed in the center of the cuboid. Figure 6 shows the surface at different time steps.

For weighting the curvature term and the forcing term for the image segmentation, the parameters $\sigma = 1$ and $\lambda = 100$ are used. At time step $m = 214$ a splitting of the evolving surface occurs. In the subsequent iterations steps, the two new surfaces each evolve to a ball. To detect the topology change, we use an auxiliary background grid with grid size $a = 0.025$ as described in Section 3.3.1. A cube of the grid is considered for possible topology changes if more than $N_{\text{detect}} = 10$ nodes are located inside the cube. We further use the parameters $thr1 = 30°$, $thr2 = 150°$ and $thr3 = 40°$.

Furthermore, we also perform a time step control (cf. Section 3.4.2) using the thresholds $\delta X_n^{\min} = 0.003$ and $\delta X_n^{\max} = 0.05$. For almost each time step, a time step size of $\Delta t = 10^{-4}$ was used. Only immediately after the splitting, for two time steps, Δt was reduced to 10^{-5} to avoid a too fast retraction of the newly generated surfaces close to the former splitting point.

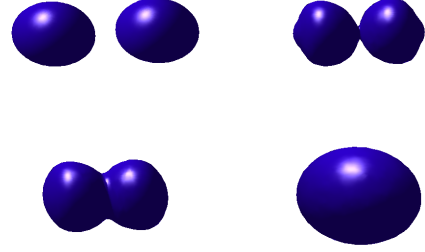The reversed topology change of splitting is a merging

of two surfaces to one single surface. The initial surfaces in the next example are two balls. The image domain is given by $\Omega = [-1.2, 1.2] \times [-0.8, 0.8] \times [-0.8, 0.8]$ and the image intensity function is defined by

$$u_0 : \Omega \to \mathbb{R}, \quad u_0(\vec{x}) = \begin{cases} 0 & \text{if } \|\vec{x}\| \leq 0.6, \\ 1 & \text{else.} \end{cases}$$

As weighting parameters $\sigma = 2$ and $\lambda = 60$ are used. Time step control is performed applying the thresholds $\delta X_n^{\min} = 0.001$ and $\delta X_n^{\max} = 0.02$. No change of the time step size is necessary in this example; the time step size $\Delta t = 10^{-4}$ need not be changed throughout the evolution. To detect the merging, $a = 0.03$, $N_{\text{detect}} = 10$ and $thr1 = 20°$, $thr2 = 150°$ and $thr3 = 40°$ are used. The resulting surfaces of this experiment at different time steps are shown in Figure 7.

Since the surface grows continuously, some simplices have to be refined as described in Section 3.4.3. The desired area for one simplex is $A_{\text{desired}} = 0.001$; a simplex is refined by bisection of its largest angle if its area is larger than a certain factor of A_{desired}. A simplex is also bisected if one angle is larger than 170°. A simplex is deleted if one angle is smaller than 2° or if its area is smaller than 1% of the desired area of A_{desired}.

In the next examples, we demonstrate another kind of topology changes: increase and decrease of the genus of a surface. Therefore, we consider an image segmentation example where a sphere should evolve to a torus. The image intensity function is given by

$$u_0(\vec{x}) = \begin{cases} 0 & \text{if } (\sqrt{x_1^2 + x_2^2} - R)^2 + x_3^2 \leq r^2, \\ 1 & \text{else,} \end{cases} \quad (29)$$

where $R = 1.2$ and $r = 0.4$ are used here.

Figure 8 shows the surface, its mesh (cross-section of the mesh) at different time steps. For this example we
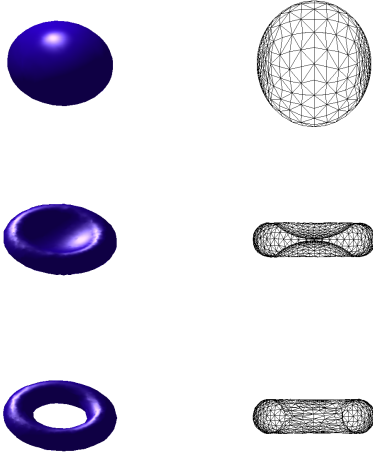
13

Figure 8: Demonstration of an increase of the genus of a surface. Surface, mesh and cross-section at step $m = 0, 325, 500$ (row-wise) at time $t_m = 0, 0.325, 0.5$. Column 1-2: surface and mesh (cross-section).
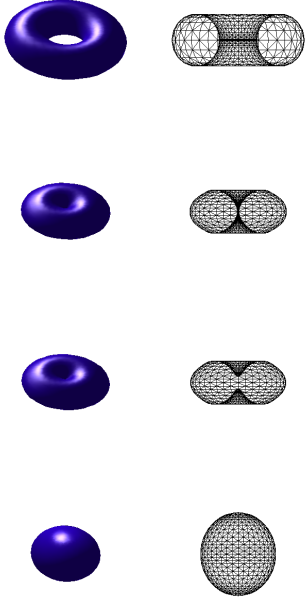

Figure 9: 3D image segmentation example where a torus evolves to a ball. Surface at step $m = 0, 425, 500, 1000$ (row-wise) at time $t_m = 0, 0.0425, 0.04406, 0.27316$. Column 1-2: surface and mesh (cross-section).
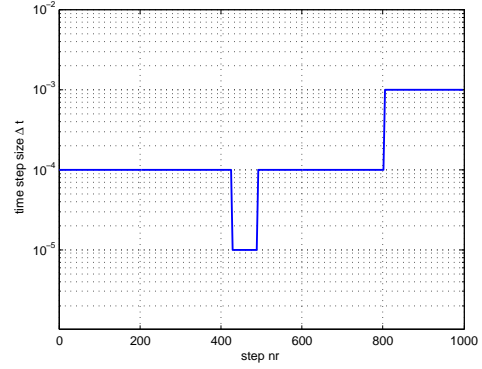

Figure 10: Time step sizes during the evolution of the torus to a ball.

apply $\sigma = 1$, $\lambda = 60$ (weighting parameters). The topology change is detected using $a = 0.0565$, $N_{\text{detect}} = 8$ and $thr1 = 20°$, $thr2 = 150°$ and $thr3 = 40°$. As parameters to control the refinement, the desired triangle area is set to $A_{\text{desired}} = 0.005$, and the angles $170°$ and $2°$ are used for bisection or deletion of a triangle, respectively.

Further, $\delta X_n^{\min} = 0.01$ and $\delta X_n^{\max} = 0.1$ are applied as thresholds for the time step size control. Throughout the evolution, there was no need to change the initial time-step size of $\Delta t = 10^{-3}$.

At time step $m = 325$, two different parts (top and bottom) of the surface with nearly opposite normal vector nearly touch. A topology change is detected and a small hole occurs. The genus of the surface is increased from $g = 0$ to $g = 1$. At time step $m = 500$, the 3D object, a torus, is detected; its boundary is represented by the surface.

Finally, we present an example where a torus is used as initial surface and a sphere should be detected. The image intensity function is given by

$$u_0 : \Omega \to \mathbb{R}, \quad u_0(\vec{x}) = \left\{ \begin{array}{ll} 0 & \text{if } \|\vec{x}\| \leq 0.8, \\ 1 & \text{else.} \end{array} \right.$$

Figure 9 shows the surface at several time steps. As weighting parameters $\sigma = 1$ and $\lambda = 20$ are applied. For the detection of the decrease of genus, the parameters $a = 0.025$, $N_{\text{detect}} = 20$ and $thr1 = 20°$, $thr2 = 150°$ and $thr3 = 40°$ are used.

The time step size is controlled using the thresholds $\delta X_n^{\min} = 0.0005$ and $\delta X_n^{\max} = 0.01$. Figure 10 shows the time step sizes during the image segmentation process. After the topology change the time step size is decreased from 10^{-4} to 10^{-5}. Later it is increased to speed up the segmentation.
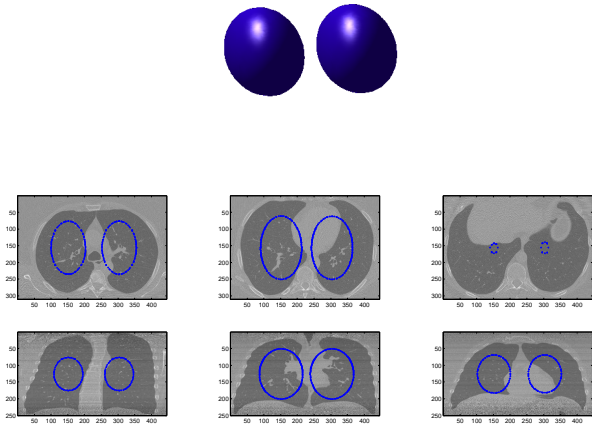
14

Figure 11: Lung segmentation: Surfaces (row 1) and cross-sections (row 2: $z = 80, 150, 200$, row 3: $y = 80, 150, 200$) at $m = 0$ at time $t = 0$. The original images are from the Lung Image Database Consortium image collection (LIDC-IDRI) of The Cancer Imaging Archive (TCIA), see [31], [2], [30]..
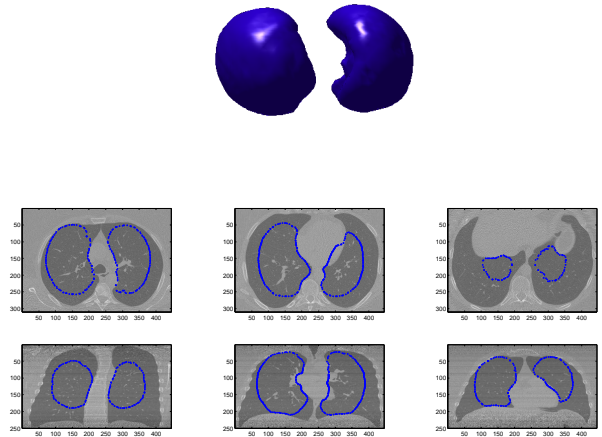

Figure 12: Lung segmentation: Surfaces (row 1) and cross-sections (row 2: $z = 80, 150, 200$, row 3: $y = 80, 150, 200$) at $m = 100$ at time $t = 10$. The original images are from the Lung Image Database Consortium image collection (LIDC-IDRI) of The Cancer Imaging Archive (TCIA), see [31], [2], [30].
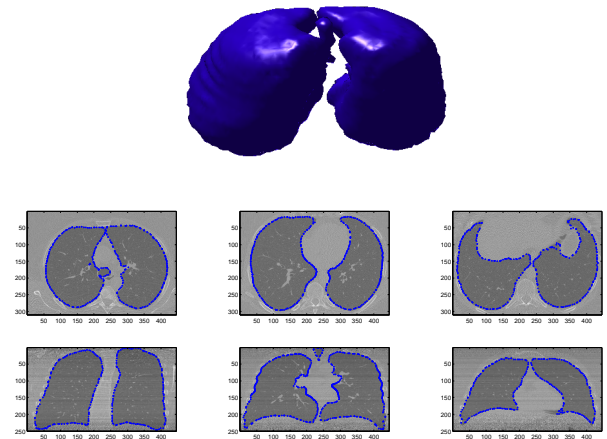
4.2 Segmentation of Medical 3D Images

In this section, we apply the segmentation method for three-dimensional images to medical image data. Segmentation of medical images is a challenging task due to possible high noise and image artifacts, see [35].

3D image data often consists of a set of 2D slice images generated by radiology scans, for example computed tomography (CT) and magnetic resonance (MR) scans. With a 3D image segmentation technique, one can segment organs (heart, lung, abdomen, liver, etc.) or tumors from their environment. The output, i.e. the resulting surface, serves as a reconstruction and visualization of the medical object and could be used for further medical analysis and diagnostic purposes: After the segmentation, one can compute the area of the triangulated surfaces and the volume of the enclosed regions. The area of the surfaces and the volume of the regions could be used for example to analyze if a tumor has been growing in the time between two radiological examinations.

First, we consider a sample 3D image of the Lung Image Database Consortium image collection (LIDC-IDRI) of The Cancer Imaging Archive (TCIA) (https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI), see [31], [2], [30])[1]. The

Figure 13: Lung segmentation: Surfaces (row 1) and cross-sections (row 2: $z = 80, 150, 200$, row 3: $y = 80, 150, 200$) at $m = 600$ at time $t = 60$. The original images are from the Lung Image Database Consortium image collection (LIDC-IDRI) of The Cancer Imaging Archive (TCIA), see [31], [2], [30].

---

[1]The author acknowledges the National Cancer Institute and the Foundation for the National Institutes of Health, and their

data set consists of diagnostic CT scans. The original data set consists of 2D slice images stored as DICOM files. The files are first preprocessed to cuboid 3D images with $N_x \times N_y \times N_z$ voxels, here: $N_x = 445$, $N_y = 310$ and $N_z = 250$.

Figures 11-13 show the evolving 3D surfaces and six representative 2D cross-sections at different time steps $m = 0, 100, 600$. In the subfigures showing 2D cross-sections, the image cross-sections for constant z (in detail $z = 80, 150, 200$) and constant y (in detail $y = 80, 150, 200$) are drawn as well as the intersection points of the surfaces' edges with the cross-section planes.

For the image segmentation the weight of the curvature term is set to $\sigma = 10$, the weight of the external forcing term to $\lambda = 1000$. As parameters for the time step size control, $\delta X_n^{\min} = 0.05$ and $\delta X_n^{\max} = 2$ are used. The time step size $\Delta t = 0.1$ need not be changed during the segmentation.

In a second experiment, we consider an experiment where a topology change occurs. We perform again a lung segmentation starting now with one initial surface which is split into two surfaces. Figure 14-18 show the surface(s) at time step $m = 0, 50, 100, 500, 900$ as well as cross-sections of the image and of the surface(s). For the cross-sections, we consider the planes given by $z = 80, 120, 160$ and $y = 50, 64, 80$.

The splitting occurs at time step $m = 50$. To detect the topology change, we use an auxiliary background grid with grid size $a = 2$. A cube of the grid is considered for possible topology changes if more than $N_{\text{detect}} = 8$ nodes are located inside the cube. Further, we use the parameters $thr1 = 30°$, $thr2 = 150°$ and $thr3 = 40°$, recall Section 3.3.2. After the splitting, the two surfaces grow and new triangles are created by bisection of too large triangles. For the segmentation we use the parameters $\sigma = 1$ and $\lambda = 20$. The time step size is set to $\Delta t = 0.2$ with time step control using $\delta X_n^{\max} = 2$, $\delta X_n^{\min} = 0.1$. However, no increase or decrease of the time step size is necessary.

As postprocessing step, we compute the volume of the two enclosed regions and the area of the region boundaries. The right lung of the patient, i.e. the left surface in the Figure 18, has an area of $A_1 = 3.309 \cdot 10^4$ and a volume of $V_1 = 2.691 \cdot 10^5$ (CT images are mirror images). The left lung of the patient (right surface in the figure) has an area of $A_2 = 2.801 \cdot 10^4$ and a volume of $V_2 = 1.923 \cdot 10^5$. Thus, as expected, the volume of the right lung is larger compared to the left lung. Note, that we handle a voxel as a cube with side length 1, resulting in values of magnitude 10^4 for the area and 10^5 for the
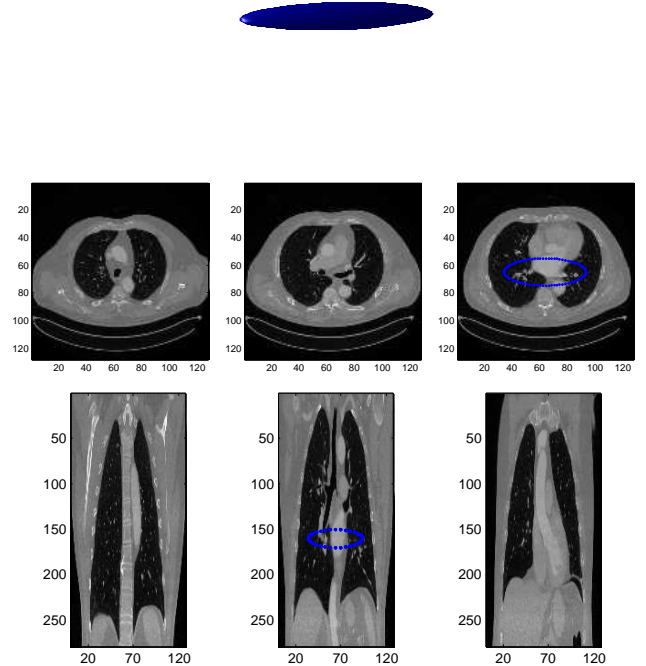

Figure 14: Lung segmentation with splitting: Surface at different viewing angles (row 1) and cross-sections (row 2: $z = 80, 120, 160$, row 3: $y = 50, 64, 80$) at $m = 0$ at time $t = 0$. Credits (original CT images): C. Stroszczynski, Radiology, University Hospital Regensburg.

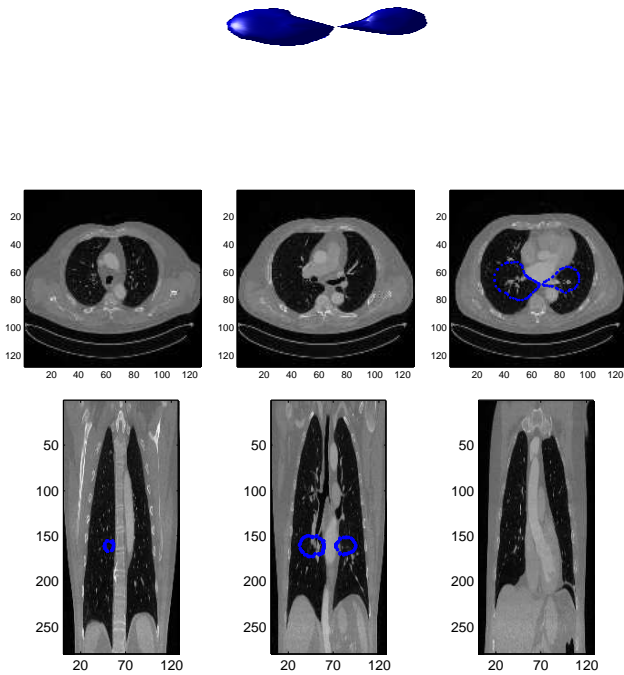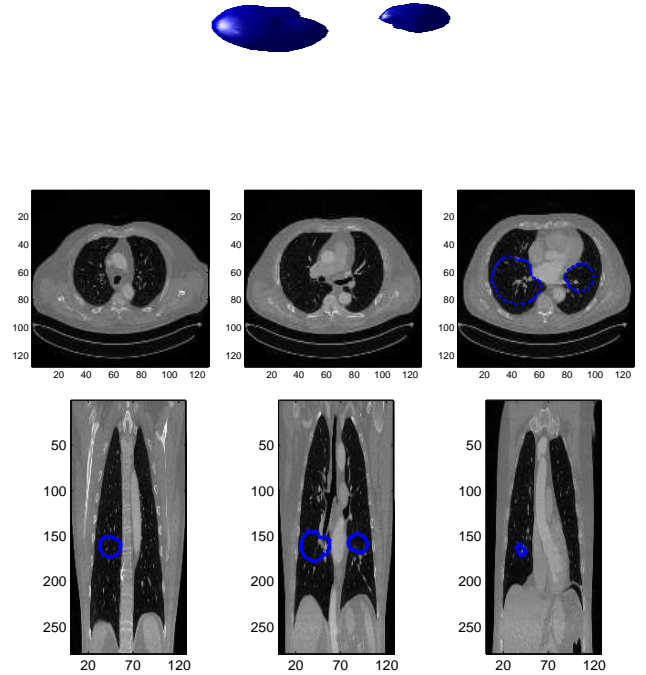critical role in the creation of the free publicly available LIDC/IDRI Database.
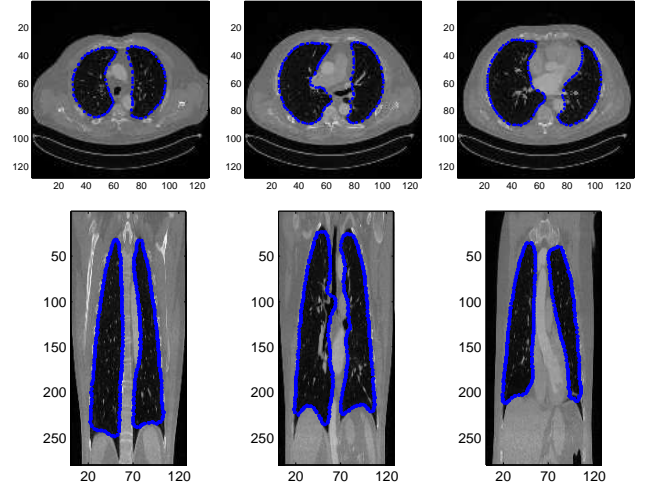
Figure 15: Lung segmentation with splitting: Surfaces (row 1) and cross-sections (row 2: $z = 80, 120, 160$, row 3: $y = 50, 64, 80$) at $m = 50$ at time $t = 10$. Credits (original CT images): C. Stroszczynski, Radiology, University Hospital Regensburg.

Figure 16: Lung segmentation with splitting: Surfaces (row 1) and cross-sections (row 2: $z = 80, 120, 160$, row 3: $y = 50, 64, 80$) at $m = 100$ at time $t = 20$. Credits (original CT images): C. Stroszczynski, Radiology, University Hospital Regensburg.
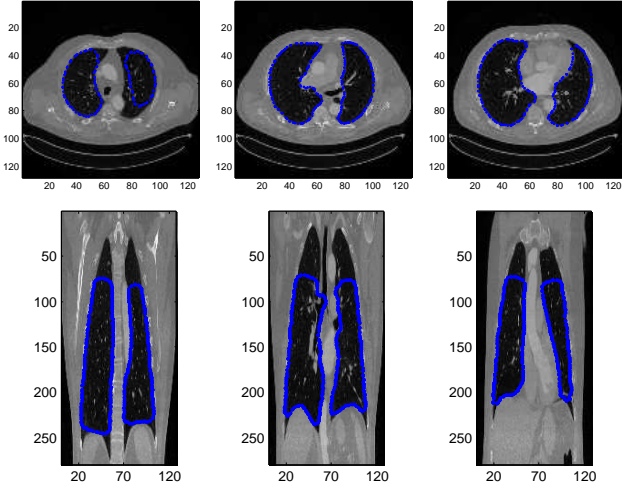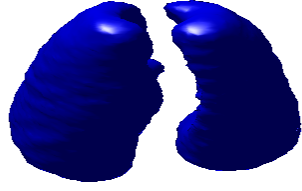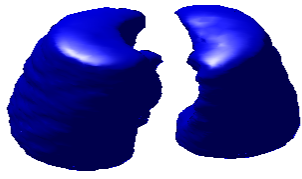
Figure 17: Lung segmentation with splitting: Surfaces (row 1) and cross-sections (row 2: $z = 80, 120, 160$, row 3: $y = 50, 64, 80$) at $m = 500$ at time $t = 100$. Credits (original CT images): C. Stroszczynski, Radiology, University Hospital Regensburg.

Figure 18: Lung segmentation with splitting: Surfaces (row 1) and cross-sections (row 2: $z = 80, 120, 160$, row 3: $y = 50, 64, 80$) at $m = 900$ at time $t = 180$. Credits (original CT images): C. Stroszczynski, Radiology, University Hospital Regensburg.

volume. If details on the acquisition system of the CT images are known (like the slice thickness, and the height and width of one pixel of a slice image), the area and the volume can be computed precisely and can be expressed in the metric system for practical interpretation of the values.

5   Conclusion

We presented a new parametric method for segmentation of 3D images. We considered extensions of the Mumford-Shah and Chan-Vese functional for 3D image segmentation by active surface. For the time-dependent surfaces, we proposed a parametric scheme and introduced an efficient numerical scheme based on a finite element approximation. A novel method to detect and perform topology changes of the surfaces has been presented which uses a virtual auxiliary background grid. Due to the fact that for the main computations only a two-dimensional grid is used, the developed method is very efficient from a computational point of view. Several artificial images have been studied to demonstrate splitting and merging of surfaces, and increase and decrease of the genus of a surface. We successfully applied our method to real medical 3D image data from computed tomography, including an example with a topology change.

Acknowledgements

References

[1] R. Ardon, L. D. Cohen, and A. Yezzi. A new implicit method for surface segmentation by minimal paths: applications in 3D medical images. In A. Rangarajan, B. Vemuri, and A. L. Yuille, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 3757 of *Lecture Notes in Computer Science*, pages 520–535. Springer Berlin Heidelberg, 2005.

[2] S. G. Armato, III, G. McLennan, L. Bidaut, M. F. McNitt-Gray, C. R. Meyer, A. P. Reeves, B. Zhao, D. R. Aberle, C. I. Henschke, E. A. Hoffman, E. A. Kazerooni, H. MacMahon, E. J. R. van Beek, D. Yankelevitz, A. M. Biancardi, P. H. Bland, M. S. Brown, R. M. Engelmann, G. E. Laderach, D. Max, R. C. Pais, D. P.-Y. Qing, R. Y. Roberts, A. R. Smith, A. Starkey, P. Batra, P. Caligiuri, A. Farooqi, G. W. Gladish, C. M. Jude, R. F. Munden, I. Petkovska, L. E. Quint, L. H. Schwartz, B. Sundaram, L. E. Dodd, C. Fenimore, D. Gur, N. Petrick, J. Freymann, J. Kirby, B. Hughes, A. Vande Casteele, S. Gupte, M. Sallam, M. D. Heath, M. H. Kuhn, E. Dharaiya, R. Burns, D. S. Fryd, M. Salganicoff, V. Anand, U. Shreter, S. Vastagh, B. Y. Croft, and L. P. Clarke. The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): A completed reference database of lung nodules on CT scans. *Medical Physics*, 38(2):915–931, 2011.

[3] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing*. Springer, New York, 2006.

[4] J.-F. Aujol and S. H. Kang. Color image decomposition and restoration. *Journal of Visual Communication and Image Representation*, 17(4):916–928, 2006.

[5] M. Balažovjech, K. Mikula, M. Petrášová, and J. Urbán. Lagrangean method with topological changes for numerical modelling of forest fire propagation. In *Proceedings of ALGORITMY 2012, 19th Conference on Scientific Computing*, pages 42–52, Vysoké Tatry, Podbansk'v, Slovakia, 2012.

[6] E. Bänsch, P. Morin, and R. H. Nochetto. A finite element method for surface diffusion: the parametric case. *Journal of Computational Physics*, 203(1):321–343, 2005.

[7] J. W. Barrett, H. Garcke, and R. Nürnberg. A variational formulation of anisotropic geometric evolution equations in higher dimensions. *Numerische Mathematik*, 109(1):1–44, 2008.

[8] J. W. Barrett, H. Garcke, and R. Nürnberg. On the parametric finite element approximation of evolving hypersurfaces in $\mathbb{R}^3$. *Journal of Computational Physics*, 227(9):4281–4307, 2008.

[9] J. W. Barrett, H. Garcke, and R. Nürnberg. Parametric approximation of Willmore flow and related geometric evolution equations. *SIAM Journal on Scientific Computing*, 31(1):225–253, 2008.

[10] H. Benninghoff and H. Garcke. Efficient image segmentation and restoration using parametric curve evolution with junctions and topology changes.

SIAM Journal on Imaging Sciences, 7(3):1451–1483, 2014.

[11] K. A. Brakke. The surface evolver. *Experimental Mathematics*, 1(2):141–165, 1992.

[12] T. Brochu and R. Bridson. Robust topological operations for dynamic explicit surfaces. *SIAM Journal on Scientific Computing*, 31(4):2472–2493, 2009.

[13] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

[14] V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert. Minimal surfaces: a geometric three dimensional segmentation approach. *Numerische Mathematik*, 77:423–451, 1997.

[15] T. F. Chan, S. H. Kang, and J. Shen. Total variation denoising and enhancement of color images based on the CB and HSV color models. *Journal of Visual Communication and Image Representation*, 12(4):422–435, 2001.

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

[17] L. D. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 53(2):211–218, 1991.

[18] Laurent D. Cohen and Isaac Cohen. Finite element methods for active contour models and balloons for 2D and 3D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, 1993.

[19] K. Deckelnick, G. Dziuk, and C. M. Elliott. Computation of geometric partial differential equations and mean curvature flow. *Acta Numerica*, 14:139–232, 2005.

[20] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[21] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.

[22] H. Li, A. Yezzi, and L. D. Cohen. Fast 3D brain segmentation using dual-front active contours with optional user-interaction. In Y. Liu, T. Jiang, and C. Zhang, editors, *Computer Vision for Biomedical Image Applications*, volume 3765 of *Lecture Notes in Computer Science*, pages 335–345. Springer Berlin Heidelberg, 2005.

[23] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, 1995.

[24] K. Mikula, N. Peyriéras, M. Remešíková, and Stašová. Segmentation of 3D cell membrane images by PDE methods and its applications. *Computers in Biology and Medicine*, 41(6):326–339, 2011.

[25] K. Mikula, N. Peyriéras, and R. Špir. Numerical algorithm for tracking cell dynamics in 4D biomedical images. *Accepted for publication in Discrete and Continuous Dynamical Systems - Series S*, 2015.

[26] K. Mikula and J. Urbán. New fast and stable Lagrangean method for image segmentation. In *Proceedings of the 5th International Congress on Image and Signal Processing (CISP 2012)*, pages 834–842, Chongqing, China, 2012.

[27] J. Mille. Narrow band region-based active contours and surfaces for 2D and 3D segmentation. *Computer Vision and Image Understanding*, 113(9):946–965, 2009.

[28] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989.

[29] S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[30] A. P. Reeves and A. M. Biancardi. The Lung Image Database Consortium (LIDC) Nodule Size Report, Release: 2011-10-27, 2011.

[31] A. P. Reeves, A. M. Biancardi, T. V. Apanasovich, C. R. Meyer, H. MacMahon, E. J.R. van Beek, E. A. Kazerooni, D. Yankelevitz, M. F. McNitt-Gray, G. McLennan, S. G. Armato III, C. I. Henschke, D. R. Aberle, B. Y. Croft, and L. P. Clarke. The Lung Image Database Consortium (LIDC): A comparison of different size metrics for pulmonary nodule measurements. *Academic Radiology*, 14(12):1475–1485, 2007.

[32] R. Ronfard. Region-based strategies for active contour models. *International Journal of Computer Vision*, 13(2):229–251, 1994.

[33] O. Rousseau and Y. Bourgault. Heart segmentation with an iterative Chan-Vese algorithm, 2009.

[34] O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, and F. Lenzen. *Variational methods in imaging*. Number 167 in Applied Mathematical Sciences. Springer, 2009.

[35] N. Sharma and L. M. Aggarwal. Automated medical image segmentation techniques. *Journal of Medical Physics*, 35(1):3–14, 2010.

[36] T. Shen and X. Huang. 3D medical image segmentation by multiple-surface active volume models. In G.-Z. Yang, D. Hawkes, D. Rueckert, A. Noble, and C. Taylor, editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI 2009*, volume 5762 of *Lecture Notes in Computer Science*, pages 1059–1066. Springer Berlin Heidelberg, 2009.

[37] B. Tang, G. Sapiro, and V. Caselles. Color image enhancement via chromaticity diffusion. *IEEE Transactions on Image Processing*, 10(5):701–707, 2002.

[38] A. Tsai, A. 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.

[39] J. K. Udupa and G. T. Herman. *3D imaging in medicine*. CRC Press, 2nd edition, 1999.

[40] A. Yezzi Jr., S. Kichenassamy, A. Kumar, P. Olver, and A. Tannenbaum. A geometric snake model for segmentation of medical imagery. *IEEE Transactions on Medical Imaging*, 16(2):199–209, 1997.