CrossMark

Segmentation and Restoration of Images on Surfaces by Parametric Active Contours with Topology Changes

Heike Benninghoff[1] · **Harald Garcke**[2]

Abstract In this article, a new method for segmentation and restoration of images on two-dimensional surfaces is given. Active contour models for image segmentation are extended to images on surfaces. The evolving curves on the surfaces are mathematically described using a parametric approach. For image restoration, a diffusion equation with Neumann boundary conditions is solved in a postprocessing step in the individual regions. Numerical schemes are presented which allow to efficiently compute segmentations and denoised versions of images on surfaces. Also topology changes of the evolving curves are detected and performed using a fast sub-routine. Finally, several experiments are presented where the developed methods are applied on different artificial and real images defined on different surfaces.

1 Introduction

We study the problem of segmentation and restoration of images defined on two-dimensional surfaces. The Mumford–

✉ Heike Benninghoff
heike.benninghoff@dlr.de

Harald Garcke
harald.garcke@ur.de

[1] Deutsches Zentrum für Luft- und Raumfahrt (DLR), 82234 Weßling, Germany

[2] Fakultät für Mathematik, Universität Regensburg, 93040 Regensburg, Germany

Shah functional [24] can be extended and reformulated for image data given on surfaces. Segmentation aims at dividing an image on a surface in characteristic regions, for example in regions of similar gray value or similar color. The objective of image restoration is to denoise the original image while preserving sharp edges in the image.

Active contours [16] originally developed for planar images can also be used to segment images on surfaces. We let one or more time-dependent curves $\Gamma(t)$, $t \in [0, T]$, evolve in time on a surface $\mathcal{M} \subset \mathbb{R}^3$ according to a flow such that the curves are attracted by edges or region boundaries.

Existing studies and works on evolution of curves on surfaces and on active contours for images on surfaces differ in the way how the surface and the curves are described mathematically. A geometric scale space for images on parametric surfaces is introduced in [17] using level sets. The image is handled implicitly by considering its iso-gray levels. In [29], Spira and Kimmel consider flows of curves on parametric surfaces and perform edge detection using a variant of the geodesic active contour method [7] for images on surfaces. In [29], the authors restrict on surfaces which have a global parameterization. Evolution equations are solved with the level set method [26] by considering the pre-image of the curve resulting in a planar curve in the parameterization plane. However, this approach has intrinsic disadvantages since the pre-image of the curve is used. In [18], drawbacks of this approach concerning the scaling behavior are discussed. Another drawback is the fact that the method does not allow to incorporate a balloon term [18].

In [11], a different method to represent the surface is proposed, where the surface is modeled by the zero level set of a three-dimensional function. The zero level set of a second three-dimensional function, which is time dependent, is used additionally. The curve is represented by the intersection of the two zero level sets. This approach is used in

[18,34] for image segmentation with geodesic active contours. A drawback of the method is that a one-dimensional curve evolution problem is extended to a three-dimensional problem. To reduce the effort, a new narrow band technique is given in [18].

In [30], image segmentation is performed using the Chan–Vese [10] model for images on surfaces with level set methods in combination with a so-called closest point method. One iteration step of the Chan–Vese model in a small 3D neighborhood of the surface is computed followed by an interpolation step.

Flows of the form $V_n = \kappa_{\mathcal{M}} + F$ are studied in [23], where V_n is the velocity contribution normal to the curve, but tangential to the surface, $\kappa_{\mathcal{M}}$ is the geodesic curvature of the curve, and F is an external forcing term. The authors restrict on graph surfaces and solve a system of partial differential equations for the parameterization $\mathbf{x}$, the tangent angle, the geodesic curvature $\kappa_{\mathcal{M}}$, and for $\|\mathbf{x}_\rho\|$, where ρ is the parametrization variable.

In this work, we make use of a concept developed in [4], where Barrett et al. consider flows of curves on surfaces like geodesic curvature flow, geodesic surface diffusion, and geodesic Willmore flow of curves. Here, we apply the methods of [4] to image segmentation applications resulting in a curvature-driven flow: The normal velocity is the sum of the geodesic curvature $\kappa_{\mathcal{M}}$ weighted with a parameter σ and an external forcing term. The forcing term is designed for image segmentation based on an extension of the Chan–Vese method to images on surfaces. As in [4], a parametric approach is used to describe the curve. The surface is given implicitly, as zero level set of a smooth function Φ. However, for practical image segmentation, the function Φ needs not to be explicitly known. Our method requires only a normal vector to the surface at each point on the surface. The developed segmentation technique can handle multiple phases and networks of curves including triple junctions.

The second image processing task we consider in this article is image restoration. To denoise a given image, we solve a diffusion equation on the surface $\mathcal{M}$. For related works on surface partial differential equations, we refer to [14] and the references therein. For a practical application see [12], where data of the Earth's surface captured on board of satellites are filtered by nonlinear diffusion. An implicit representation of the surface is used in [6], where the surface is embedded as zero level set of a higher dimensional function and the partial differential equations are solved on a fixed Cartesian grid using a special embedding function.

Methods to solve total variation problems on surfaces are proposed in [19]. In detail, the method of [28] for denoising images on surfaces and the method of [8] (a convexified Chan–Vese model) for segmentation of images on surfaces are generalized. A direct, called *intrinsic*, approach is pursued: Lai and Chan [19] perform the resulting calculations

directly on the given surface using differential geometry techniques and finite elements for images given on triangulated surfaces. They also provide an overview and a comparison of several approaches for variational problems on surfaces (level set methods, parametric surfaces, and direct/intrinsic methods). Total variation-based image restoration and segmentation are also considered in [33], where the authors propose an extension of the augmented Lagrangian method (see e.g., [32]) for scalar and vectorial total variation problems to images on surfaces.

In this article, we will perform restoration of images on surfaces by considering an extension of the Mumford–Shah problem [24]. The image restoration is performed as a postprocessing step after the segmentation. We solve a diffusion equation with Neumann boundary conditions in the already-segmented regions. Thus, the image is not smoothed out across the region boundaries.

For both segmentation and restoration, we present efficient numerical schemes. Further, topology changes of the parametric curves are detected efficiently. In [5], we used and extended a method of [22] for efficient detection of topology changes, such that also topology changes involving triple junctions can be handled during the curve evolution. We will extend this idea to curves on surfaces to detect several topology changes including splitting and merging of curves, and creation of triple junctions. The main idea of the method based on [22] is the use of an auxiliary background grid. Together with the property that our scheme leads to a nearly equidistribution of mesh points along the curves (cf. [4]), topology changes of curves on surfaces are detected very robustly.

In practical applications, surfaces are often not given as smooth surfaces but in a discrete form, for example as triangulated surfaces. We present all necessary computational aspects when applying the segmentation and restoration models to real data.

In summing up, we develop a novel scheme for both segmentation and restoration of images defined on surfaces. Using our parametric approach, the evolution of curves is a one-dimensional problem; the postprocessing image diffusion is a two-dimensional problem computed only once after the segmentation has been finished. Compared to other approaches in the literature, where the curve is embedded in a higher dimensional space, our method is very efficient from a computational point of view.

2 Segmentation and Restoration of Images on Surfaces

2.1 Preliminaries

Let $\mathcal{M} \subset \mathbb{R}^3$ be a smooth two-dimensional manifold. We assume that we can describe $\mathcal{M}$ by the zero level set of a
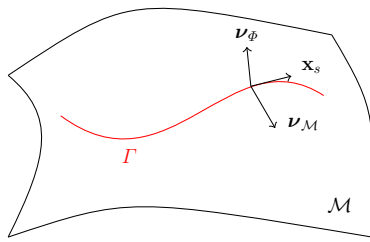
Fig. 1 Illustration of the vector fields $\mathbf{x}_s$, $\boldsymbol{\nu}_\Phi$, and $\boldsymbol{\nu}_\mathcal{M} = \mathbf{x}_s \times \boldsymbol{\nu}_\Phi$

function Φ:

$$\mathcal{M} = \left\{ \mathbf{z} \in \mathbb{R}^3 \,:\, \Phi(\mathbf{z}) = 0 \right\}, \tag{1}$$

where $\Phi \in C^2(\mathbb{R}^3, \mathbb{R})$ is a function with $\|\nabla\Phi(\mathbf{z})\| > 0$ for $\mathbf{z} \in \mathcal{M}$ ($\| . \|$ denoting the Euclidean norm). A unit normal vector field $\mathbf{n}_\Phi$ on $\mathcal{M}$ is given by $\mathbf{n}_\Phi(\mathbf{z}) := \nabla\Phi(\mathbf{z})/\|\nabla\Phi(\mathbf{z})\|$ for $\mathbf{z} \in \mathcal{M}$.

Let $\Gamma \subset \mathcal{M}$ be a curve on $\mathcal{M}$ parameterized by $\mathbf{x} : I \to \mathcal{M}$, where I is a one-dimensional reference manifold, for example the unit interval $I = [0, 1]$ for open curves or $\mathbb{R}/\mathbb{Z}$ for closed curves. We define $\boldsymbol{\nu}_\Phi : I \to \mathbb{R}^3$ such that $\boldsymbol{\nu}_\Phi(\rho) := \mathbf{n}_\Phi(\mathbf{x}(\rho))$ is the surface normal evaluated at $\mathbf{x}(\rho)$ for $\rho \in I$. Further, we define $\boldsymbol{\nu}_\mathcal{M} : I \to \mathbb{R}^3$ by $\boldsymbol{\nu}_\mathcal{M}(\rho) := \mathbf{x}_s(\rho) \times \boldsymbol{\nu}_\Phi(\rho)$, where s denotes the arc length. We note that $\boldsymbol{\nu}_\mathcal{M}$ is perpendicular to $\mathbf{x}_s$, i.e., normal to the curve, but lies in the tangent space to $\mathcal{M}$. Figure 1 illustrates a possible surface $\mathcal{M}$, a curve Γ, and the vector fields $\boldsymbol{\nu}_\Phi$, $\mathbf{x}_s$, and $\boldsymbol{\nu}_\mathcal{M}$.

Further, the vector $\mathbf{x}_{ss}$ which is perpendicular to $\mathbf{x}_s$ can be written as the sum of its component in $\boldsymbol{\nu}_\Phi$-direction and its component in $\boldsymbol{\nu}_\mathcal{M}$-direction. This motivates the following definition [4]:

Definition 1 We define the *geodesic curvature* $\kappa_\mathcal{M} : I \to \mathbb{R}$ and the *normal curvature* $\kappa_\Phi : I \to \mathbb{R}$ by

$$\kappa_\mathcal{M} = \mathbf{x}_{ss} \cdot \boldsymbol{\nu}_\mathcal{M}, \quad \kappa_\Phi = \mathbf{x}_{ss} \cdot \boldsymbol{\nu}_\Phi. \tag{2}$$

As a consequence, the vector field $\mathbf{x}_{ss}$ can be expressed as

$$\mathbf{x}_{ss} = \kappa_\mathcal{M} \boldsymbol{\nu}_\mathcal{M} + \kappa_\Phi \boldsymbol{\nu}_\Phi. \tag{3}$$

2.2 Active Contours on Surfaces Based on Extensions of the Mumford–Shah and Chan–Vese Models to Images on Surfaces

Let $u_0 : \mathcal{M} \to [0, 1]$ be the intensity function of an image given on a surface $\mathcal{M}$. For segmentation and restoration, we consider an extension of the Mumford–Shah functional [24] from the planar case to the case of images and curves on surfaces. For that, we consider the following minimization problem:

Find a union of curves $\Gamma = \Gamma_1 \cup \ldots \cup \Gamma_{N_C} \subset \mathcal{M}$ and a piecewise smooth approximation $u : \mathcal{M} \to \mathbb{R}$ of the original image u_0 such that

$$E^{\mathrm{MS}}(u, \Gamma) = \sigma|\Gamma| + \int_{\mathcal{M}\setminus\Gamma} \|\nabla_\mathcal{M} u\|^2 \, \mathrm{d}A$$
$$+ \lambda \int_\mathcal{M} (u_0 - u)^2 \, \mathrm{d}A \tag{4}$$

is minimized. Here, $\sigma, \lambda > 0$ are weighting parameters, $|\Gamma|$ denotes the total length of the curves, and $\nabla_\mathcal{M} u$ is the surface gradient of u, also called tangential gradient, cf. [14]. Further, $\mathrm{d}A$ denotes the area element.

We first consider the case of one closed curve Γ on $\mathcal{M}$ without any self-intersection which is homotopic to a point. Then, the curve divides $\mathcal{M}$ into two disjoint regions Ω_1 and Ω_2 such that

$$\mathcal{M} = \Omega_1 \cup \Gamma \cup \Omega_2.$$

The indices of the regions Ω_k, $k = 1, 2$, are chosen such that $\boldsymbol{\nu}_\mathcal{M}$ points from Ω_2 to Ω_1.

For segmentation, we consider a piecewise constant approximation with $u_{|\Omega_k} = c_k \in \mathbb{R}$, $k = 1, 2$. The Mumford–Shah functional for images on surfaces (4) reduces to

$$E(\Gamma, c_1, c_2) = \sigma|\Gamma| + \lambda \int_{\Omega_1} (u_0 - c_1)^2 \, \mathrm{d}A$$
$$+ \lambda \int_{\Omega_2} (u_0 - c_2)^2 \, \mathrm{d}A. \tag{5}$$

Similar to this functional, we can consider the analog of the planar Chan–Vese model [10] for images on surfaces:

$$E(\Gamma, c_1, c_2) = \sigma|\Gamma| + \mu \int_{\Omega_1} 1 \, \mathrm{d}A$$
$$+ \lambda_1 \int_{\Omega_1} f_1 \, \mathrm{d}A + \lambda_2 \int_{\Omega_2} f_2 \, \mathrm{d}A, \tag{6}$$

where $\sigma, \lambda_1, \lambda_2 > 0$, $\mu \geq 0$ are weighting parameters. Similar to the planar Chan–Vese method [10], the function f_k, $k = 1, 2$, is defined by

$$f_k(\mathbf{z}) = (u_0(\mathbf{z}) - c_k)^2, \quad \mathbf{z} \in \overline{\Omega_k} \subset \mathcal{M}. \tag{7}$$

The model (6) with $\mu = 0$ and $\lambda_1 = \lambda_2$ is the piecewise constant case (5) of the Mumford–Shah model.

Fixing now the curve Γ in (6), we obtain the following condition for the coefficients c_k, $k = 1, 2$:

$$c_k = \frac{\int_{\Omega_k} u_0 \, \mathrm{d}A}{\int_{\Omega_k} 1 \, \mathrm{d}A}. \tag{8}$$

Let $\mathbf{x} : I = \mathbb{R}/\mathbb{Z} \to \mathcal{M}$ be a smooth parameterization of Γ.

We now derive a flow for image segmentation as gradient flow using methods of the theory of calculus of variations. Since we consider curves on the surface $\mathcal{M}$, the variations are restricted to lie on $\mathcal{M}$. Therefore, as in [4], we consider the variations to be elements of

$$\underline{V}_\Phi = \left\{ \boldsymbol{\eta} : I \to \mathbb{R}^3 \, : \, \boldsymbol{\eta} \text{ is smooth and } \boldsymbol{\eta} \cdot \boldsymbol{v}_\phi = 0 \right\} \tag{9}$$

and define for functions $\boldsymbol{\eta}, \boldsymbol{\chi} : I \to \mathbb{R}^3$ the following inner product:

$$(\boldsymbol{\eta}, \boldsymbol{\chi})_{2,\mathcal{M},\text{nor}} := \int_\Gamma \mathbf{P}_\mathcal{M} \boldsymbol{\eta} \cdot \mathbf{P}_\mathcal{M} \boldsymbol{\chi} \, \mathrm{d}s, \tag{10}$$

where $\mathbf{P}_\mathcal{M}$ is the projection onto the part in direction $\boldsymbol{v}_\mathcal{M}$, i.e., $\mathbf{P}_\mathcal{M} \boldsymbol{\eta} = (\boldsymbol{\eta} \cdot \boldsymbol{v}_\mathcal{M}) \boldsymbol{v}_\mathcal{M}$ for $\boldsymbol{\eta} : I \to \mathbb{R}^3$.

Fixing the coefficients c_1, c_2, we consider for $\boldsymbol{\eta} \in \underline{V}_\Phi$ a variation $\mathbf{y} : I \times (-\epsilon_0, \epsilon_0) \to \mathcal{M}$ with $\mathbf{y}(\rho, 0) = \mathbf{x}(\rho)$ and $\mathbf{y}_\epsilon(\rho, 0) = \boldsymbol{\eta}(\rho)$.

Let $\Gamma^\epsilon \subset \mathcal{M}$ be the image of $\mathbf{y}(., \epsilon)$, and $\Omega_1^\epsilon, \Omega_2^\epsilon \subset \mathcal{M}$ be regions such that $\mathcal{M} = \Omega_1^\epsilon \cup \Gamma^\epsilon \cup \Omega_2^\epsilon$.

Further, let $\boldsymbol{v}_\mathcal{M}^\epsilon(\rho) \in T_{\mathbf{y}(\rho,\epsilon)}\mathcal{M}$ be a vector in the tangent space to $\mathcal{M}$ in $\mathbf{y}(\rho, \epsilon)$ defined by $\boldsymbol{v}_\mathcal{M}^\epsilon(\rho) = \mathbf{y}_s(\rho, \epsilon) \times \mathbf{n}_\Phi(\mathbf{y}(\rho, \epsilon))$. The vector field $\boldsymbol{v}_\mathcal{M}^\epsilon(\rho)$ points in the direction Ω_1^ϵ. The vector $\boldsymbol{v}_\mathcal{M}^\epsilon(\rho)$ is normal to Γ^ϵ, but lies in the tangent space of $\mathcal{M}$ at the point $\mathbf{y}(\rho, \epsilon)$.

We define

$$(\delta E(\Gamma))(\boldsymbol{\eta}) := \frac{\mathrm{d}}{\mathrm{d}\epsilon}\bigg|_{\epsilon=0} \left(\sigma \int_{\Gamma^\epsilon} 1 \, \mathrm{d}s + \mu \int_{\Omega_1^\epsilon} 1 \, \mathrm{d}A \right.$$
$$\left. + \lambda_1 \int_{\Omega_1^\epsilon} f_1 \, \mathrm{d}A + \lambda_2 \int_{\Omega_2^\epsilon} f_2 \, \mathrm{d}A \right)$$

on noting that $\mathbf{y}(., \epsilon)$ and thus Γ^ϵ, Ω_1^ϵ, and Ω_2^ϵ depend on $\boldsymbol{\eta}$. We compute

$$(\delta E(\Gamma))(\boldsymbol{\eta})$$
$$= \frac{\mathrm{d}}{\mathrm{d}\epsilon}\bigg|_{\epsilon=0} \left(\sigma \int_I \|\mathbf{y}_\rho\| \, \mathrm{d}\rho + \mu \int_{\Omega_1^\epsilon} 1 \, \mathrm{d}A \right.$$
$$+ \lambda_1 \int_{\Omega_1^\epsilon} f_1 \, \mathrm{d}A + \lambda_2 \int_{\Omega_2^\epsilon} f_2 \, \mathrm{d}A \Bigg)$$
$$= \left(\sigma \int_I \frac{\mathbf{y}_\rho}{\|\mathbf{y}_\rho\|} \cdot \mathbf{y}_{\rho\epsilon} \, \mathrm{d}\rho + \mu \int_I (-\mathbf{y}_\epsilon \cdot \boldsymbol{v}_\mathcal{M}^\epsilon) \|\mathbf{y}_\rho\| \, \mathrm{d}\rho \right.$$
$$+ \lambda_1 \int_I f_1(\mathbf{y})(-\mathbf{y}_\epsilon \cdot \boldsymbol{v}_\mathcal{M}^\epsilon) \|\mathbf{y}_\rho\| \, \mathrm{d}\rho$$
$$\left. + \lambda_2 \int_I f_2(\mathbf{y})(\mathbf{y}_\epsilon \cdot \boldsymbol{v}_\mathcal{M}^\epsilon) \|\mathbf{y}_\rho\| \, \mathrm{d}\rho \right)\bigg|_{\epsilon=0}$$

$$= \sigma \int_\Gamma \mathbf{x}_s \cdot \boldsymbol{\eta}_s \, \mathrm{d}s + \int_\Gamma (-\mu - \lambda_1 f_1 + \lambda_2 f_2) \, \boldsymbol{v}_\mathcal{M} \cdot \boldsymbol{\eta} \, \mathrm{d}s$$
$$= \int_\Gamma (-\sigma \mathbf{x}_{ss} + (-\mu - \lambda_1 f_1 + \lambda_2 f_2) \, \boldsymbol{v}_\mathcal{M}) \cdot \boldsymbol{\eta} \, \mathrm{d}s$$
$$= \int_\Gamma (-\sigma \kappa_\mathcal{M} - \mu - \lambda_1 f_1 + \lambda_2 f_2) \, \boldsymbol{v}_\mathcal{M} \cdot \boldsymbol{\eta} \, \mathrm{d}s. \tag{11}$$

Here, we used a transport theorem for curves on surfaces [15]. We applied integration by parts for the second last identity. The last identity follows from (3) and $\boldsymbol{\eta} \cdot \boldsymbol{v}_\phi = 0$.

A time-dependent function $\mathbf{x} : I \times [0, T] \to \mathcal{M}$ with $\mathbf{x}_t(., t) \in \underline{V}_\Phi$ is called a solution to the gradient flow equation if

$$(\mathbf{x}_t, \boldsymbol{\eta})_{2,\mathcal{M},\text{nor}} = -(\delta E(\Gamma))(\boldsymbol{\eta}) \tag{12}$$

holds for all $\boldsymbol{\eta} \in \underline{V}_\Phi$.

Let $\boldsymbol{\eta} \in \underline{V}_\Phi$. We conclude from (11) and (12) on noting that $\boldsymbol{v}_\mathcal{M} \cdot \boldsymbol{\eta} = \boldsymbol{v}_\mathcal{M} \cdot \mathbf{P}_\mathcal{M} \boldsymbol{\eta}$

$$\mathbf{P}_\mathcal{M} \mathbf{x}_t = -(-\sigma \kappa_\mathcal{M} - \mu - \lambda_1 f_1 + \lambda_2 f_2) \, \boldsymbol{v}_\mathcal{M}. \tag{13}$$

Let $V_n := \mathbf{x}_t \cdot \boldsymbol{v}_\mathcal{M}$ denote the velocity in direction $\boldsymbol{v}_\mathcal{M}$, also called *normal velocity*. Then $\mathbf{P}_\mathcal{M} \mathbf{x}_t = V_n \boldsymbol{v}_\mathcal{M}$ and consequently the equation above leads to

$$V_n = \sigma \kappa_\mathcal{M} + F, \tag{14}$$

where F is given by

$$F(\mathbf{z}) = \mu + \lambda_1 f_1(\mathbf{z}) - \lambda_2 f_2(\mathbf{z})$$
$$= \mu + \lambda_1 (u_0(\mathbf{z}) - c_1)^2 - \lambda_2 (u_0(\mathbf{z}) - c_2)^2. \tag{15}$$

We rewrite equation (14) to a scheme for $\mathbf{x} : I \times [0, T] \to \mathbb{R}^3$ and $\kappa_\mathcal{M}, \kappa_\Phi : I \times [0, T] \to \mathbb{R}$. We assume that $\mathbf{x}(\rho, 0)$ lies on $\mathcal{M}$. To force the curve to stay on the manifold $\mathcal{M}$, the velocity in direction normal to the surface $\mathbf{x}_t \cdot \boldsymbol{v}_\Phi$ must be zero (i.e., $\mathbf{x}_t \in \underline{V}_\Phi$). We thus have the following scheme:

Let $\mathbf{x}(I, 0) = \Gamma(0) \subset \mathcal{M}$. For $t \in (0, T]$, find $\mathbf{x}(., t) : I \to \mathbb{R}^3$ and $\kappa_\mathcal{M}(., t), \kappa_\Phi(., t) : I \to \mathbb{R}$ such that

$$\mathbf{x}_t \cdot \boldsymbol{v}_\mathcal{M} = \sigma \kappa_\mathcal{M} + F, \tag{16a}$$
$$\mathbf{x}_t \cdot \boldsymbol{v}_\Phi = 0, \tag{16b}$$
$$\mathbf{x}_{ss} = \kappa_\mathcal{M} \boldsymbol{v}_\mathcal{M} + \kappa_\Phi \boldsymbol{v}_\Phi. \tag{16c}$$

2.3 Multiphase Image Segmentation with Possible Triple Junctions

We extend the above-presented two-phase segmentation with a single closed curve to more general situations. We consider a curve network with closed and open curves which partition the image domain in N_R regions. Also triple junctions can occur.

Therefore, we consider a decomposition of $\mathcal{M}$ in time-dependent regions $\Omega_1(t), \ldots, \Omega_{N_R}(t), t \in [0, T]$, separated by curves $\Gamma_1(t), \ldots, \Gamma_{N_C}(t)$. Each curve is parameterized by a time-dependent function $\mathbf{x}_i(.\,,t) : I_i \to \mathbb{R}^3$, where I_i is a one-dimensional reference manifold for $i = 1, \ldots, N_C$. Similar to the case of one curve, we set $\boldsymbol{\nu}_{\Phi,i} = \mathbf{n}_\Phi \circ \mathbf{x}_i$ and $\boldsymbol{\nu}_{\mathcal{M},i} = (\mathbf{x}_i)_s \times \boldsymbol{\nu}_{\Phi,i}$. The geodesic curvature $\kappa_{\mathcal{M},i}$ and the normal curvature $\kappa_{\Phi,i}$ are given by $\kappa_{\mathcal{M},i} = (\mathbf{x}_i)_{ss} \cdot \boldsymbol{\nu}_{\mathcal{M},i}$ and $\kappa_{\Phi,i} = (\mathbf{x}_i)_{ss} \cdot \boldsymbol{\nu}_{\Phi,i}$. All quantities are time dependent.

We define a piecewise constant image approximation by $u(.\,,t) = \sum_{k=1}^{N_R} c_k(t) \chi_{\Omega_k(t)}$, where $\chi_{\Omega_k(t)}$ is the characteristic function on the set $\Omega_k(t)$ and the coefficients $c_k(t)$ are computed by

$$c_k(t) = \frac{\int_{\Omega_k(t)} u_0 \, dA}{\int_{\Omega_k(t)} 1 \, dA}, \qquad (17)$$

i.e., they are set to the mean of u_0 in $\Omega_k(t)$.

Let $\mathbf{x}_i(.\,,0)$, $i = 1, \ldots, N_C$, be parameterizations of given curves $\Gamma_i(0) \subset \mathcal{M}$. We have to solve the following scheme for $t \in (0, T]$: find $\mathbf{x}_i(.\,,t) : I_i \to \mathbb{R}^3$, $\kappa_{\mathcal{M},i}(.\,,t), \kappa_{\Phi,i}(.\,,t) : I_i \to \mathbb{R}$ such that

$$(\mathbf{x}_i)_t \cdot \boldsymbol{\nu}_{\mathcal{M},i} = \sigma \kappa_{\mathcal{M},i} + F_i, \qquad (18a)$$

$$(\mathbf{x}_i)_t \cdot \boldsymbol{\nu}_{\Phi,i} = 0, \qquad (18b)$$

$$(\mathbf{x}_i)_{ss} = \kappa_{\mathcal{M},i} \, \boldsymbol{\nu}_{\mathcal{M},i} + \kappa_{\Phi,i} \, \boldsymbol{\nu}_{\Phi,i} \qquad (18c)$$

hold for $i = 1, \ldots, N_C$. The external force F_i is defined for $\mathbf{x} \in \mathcal{M}$ by

$$F_i(\mathbf{x}) = \mu + \lambda_{k^+(i)}(u_0(\mathbf{x}) - c_{k^+(i)})^2 + \\ - \lambda_{k^-(i)}(u_0(\mathbf{x}) - c_{k^-(i)})^2, \qquad (19)$$

where $k^+(i), k^-(i) \in \{1, \ldots, N_R\}$ are indices of two regions, such that $\boldsymbol{\nu}_{\mathcal{M},i}$ points from $\Omega_{k^-(i)}$ to $\Omega_{k^+(i)}$.

In the experiments, presented in Sect. 4, we always consider the case $\mu = 0$ and $\lambda_k = \lambda$ for all $k = 1, \ldots, N_R$, i.e., all segmentations in our demonstrations can be performed with only two weighting parameters σ and λ. The external forcing term is

$$F_i(\mathbf{x}) = \lambda \left[(u_0(\mathbf{x}) - c_{k^+(i)})^2 - (u_0(\mathbf{x}) - c_{k^-(i)})^2 \right]. \qquad (20)$$

We also allow open curves, i.e., curves with $\partial \Gamma_i(t) \neq \emptyset$. Since we consider interface curves, i.e., each curve $\Gamma_i(t)$ separates two different regions $\Omega_{k^+(i)}$ and $\Omega_{k^-(i)}$, we exclude free endpoints. This means, we exclude the case that a curve ends in $\mathcal{M}$ without meeting another curve at its endpoint. Further, we consider only smooth, compact surfaces $\mathcal{M}$ without boundary. Thus, endpoints of open curves are therefore part of triple junctions denoted with $\boldsymbol{\Lambda}_k \in \mathcal{M}$, $k = 1, \ldots, N_T$.

For each $k \in \{1, \ldots, N_T\}$, let the integers $i_{k,1}, i_{k,2}, i_{k,3} \in \{1, \ldots, N_C\}$ denote the indices of curves $\Gamma_{i_{k,l}}$, $l = 1, 2, 3$, $i_{k,1} \neq i_{k,2} \neq i_{k,3} \neq i_{k,1}$ with parameterizations $\mathbf{x}_{i_{k,l}} : I_{i_{k,l}} = [0, 1] \to \mathcal{M}$, such that

$$\mathbf{x}_{i_{k,1}}(\rho_{k,1}) = \mathbf{x}_{i_{k,2}}(\rho_{k,2}) = \mathbf{x}_{i_{k,3}}(\rho_{k,3}) = \boldsymbol{\Lambda}_k,$$

where $\rho_{k,l} \in \{0, 1\}$ corresponds to the start or endpoint of the curve $i_{k,l}$, $l = 1, 2, 3$.

At the triple junctions $\boldsymbol{\Lambda}_k$, $k = 1, \ldots, N_T$, an attachment condition and Young's law need to hold:

the triple junction $\boldsymbol{\Lambda}_k$ does not pull apart, $\qquad (21a)$

$$\sum_{l=1}^{3} (-1)^{\rho_{k,l}} \, \boldsymbol{\tau}_{i_{k,l}} = 0, \qquad (21b)$$

where $\boldsymbol{\tau}_{i_{k,l}} := (\mathbf{x}_{i_{k,l}})_s$ is a tangent vector field at $\Gamma_{i_{k,l}} \subset \mathcal{M}$, $l = 1, 2, 3$. We refer to [3] for the planar case of evolution of curves with triple junctions.

2.4 Vector-valued images

The segmentation method can be easily extended to vector-valued images such as color images. Only the external force F need to be adapted compared to scalar images. The adaptations of F can be done similarly as for planar images [5,9].

Let $\mathbf{u}_0 = (u_{0,1}, u_{0,2}, u_{0,3}) : \mathcal{M} \to \mathbb{R}^3$ represent a color image in the RGB color space. The three components of the vector-valued image function represent the red, green, and blue color channels. The external force F_i, $i = 1, \ldots, N_C$, in (20) has to be modified. We therefore set

$$F_i = \sum_{j=1}^{3} \lambda_j \left[(u_{0,j} - c_{k^+(i),j})^2 - (u_{0,j} - c_{k^-(i),j})^2 \right],$$

where λ_j weights the j-th color component, $j = 1, 2, 3$. The coefficients $\mathbf{c}_k = (c_{k,1}, c_{k,2}, c_{k,3})$, $k = 1, \ldots, N_R$, are given by setting $c_{k,j}$ to the mean of $u_{0,j}$ in Ω_k.

Another color space, we will use for segmentation of color images, is the chromaticity–brightness space. Let $\mathbf{v}_0 = \mathbf{u}_0 / \|\mathbf{u}_0\|$ be the chromaticity and $b_0 = \|\mathbf{u}_0\|$ the brightness of an image with image function $\mathbf{u}_0$. We modify F_i to

$$F_i = \lambda_C \left[\|\mathbf{v}_0 - \mathbf{v}_{k^+(i)}\|^2 - \|\mathbf{v}_0 - \mathbf{v}_{k^-(i)}\|^2 \right] \\ + \lambda_B \left[(b_0 - b_{k^+(i)})^2 - (b_0 - b_{k^-(i)})^2 \right],$$

where λ_C weights the chromaticity term and λ_B weights the brightness term. Here, $\mathbf{v}_k$ is a normalized mean of $\mathbf{v}_0$ in the region Ω_k (see [5] for details) and b_k is the mean of b_0 in Ω_k.

2.5 Restoration of Images on Surfaces with Edge Enhancement

Searching for a minimizer of the Mumford–Shah functional for images on surfaces (4) involves both a set of curves Γ and an image approximation u. The segmentation technique presented above uses piecewise constant image approximations to divide an image into characteristic regions of similar image intensity or color. For image restoration, more details of the original image u_0 should be kept; the piecewise constant approximation would be a too large simplification. Therefore, we aim at approximating $u_0 : \mathcal{M} \to \mathbb{R}$ by a piecewise smooth function $u : \mathcal{M} \to \mathbb{R}$.

We propose to first perform a segmentation of the image with piecewise constant approximations by solving the evolution equations (18) with (21) in case of triple junctions. This is followed by a restoration using the already identified regions. By denoising the image in this way during a post-processing step, the edges in the image will not be smoothed out if the curves Γ match with these edges.

We thus consider piecewise smooth approximations $u_{|\Omega_k} = u_k$, where $u_k : \Omega_k \to [0, 1]$ is a smooth function defined on the region $\Omega_k \subset \mathcal{M}$, $k = 1, \ldots, N_R$. For smoothing the image in the regions Ω_k, we derive surface partial differential equations from the Mumford–Shah functional. Since the curve set Γ has already been determined, we can fix Γ in the functional (4) and consider variations of u of the form $u + \epsilon v$, for $v : \mathcal{M} \to \mathbb{R}$ and $\epsilon > 0$. We compute

$$
\begin{aligned}
&\left.\frac{\mathrm{d}}{\mathrm{d}\epsilon}\right|_{\epsilon=0} E^{\mathrm{MS}}(u + \epsilon v, \Gamma) \\
&= \lim_{\epsilon \to 0} \frac{1}{\epsilon} \left(E^{\mathrm{MS}}(u + \epsilon v, \Gamma) - E^{\mathrm{MS}}(u, \Gamma) \right) \\
&= \lim_{\epsilon \to 0} \frac{1}{\epsilon} \left(\int_{\mathcal{M} \setminus \Gamma} \left(2\epsilon \nabla_{\mathcal{M}} u \cdot \nabla_{\mathcal{M}} v + \epsilon^2 \|\nabla_{\mathcal{M}} v\|^2 \right) \mathrm{d}A \right. \\
&\quad \left. + \lambda \int_{\mathcal{M}} \left(2\epsilon (u - u_0) v + \epsilon^2 v^2 \right) \mathrm{d}A \right) \\
&= 2 \int_{\mathcal{M} \setminus \Gamma} \nabla_{\mathcal{M}} u \cdot \nabla_{\mathcal{M}} v \, \mathrm{d}A + 2\lambda \int_{\mathcal{M}} (u - u_0) v \, \mathrm{d}A \\
&= 2 \sum_{k=1}^{N_R} \int_{\Omega_k} (\nabla_{\mathcal{M}} u_k \cdot \nabla_{\mathcal{M}} v + \lambda (u_k - u_0) v) \, \mathrm{d}A.
\end{aligned}
$$

For a stationary solution, we search for a function u satisfying $0 = \left.\frac{\mathrm{d}}{\mathrm{d}\epsilon}\right|_{\epsilon=0} E^{\mathrm{MS}}(u + \epsilon v, \Gamma)$. This leads to

$$
0 = \int_{\Omega_k} (\nabla_{\mathcal{M}} u_k \cdot \nabla_{\mathcal{M}} v + \lambda (u_k - u_0) v) \, \mathrm{d}A
$$

for each $k = 1, \ldots, N_R$ and an arbitrary function v which is smooth on Ω_k. Using an integration by parts formula [14], we obtain

$$
\begin{aligned}
0 = &\int_{\Omega_k} (-\Delta_{\mathcal{M}} u_k + \lambda (u_k - u_0)) \, v \, \mathrm{d}A \\
&+ \int_{\partial \Omega_k} \nabla_{\mathcal{M}} u_k \cdot \boldsymbol{\mu}_k \, v \, \mathrm{d}s,
\end{aligned} \tag{22}
$$

where $\Delta_{\mathcal{M}}$ is the Laplace–Beltrami operator and $\boldsymbol{\mu}_k(\mathbf{p})$ is a unit outer normal vector on $\partial \Omega_k$ in $T_{\mathbf{p}}\mathcal{M}$, i.e., it is tangent to the surface for each $\mathbf{p} \in \partial \Omega_k \subset M$ but normal to $\partial \Omega_k$ in $\mathbf{p}$. Since $\mathcal{M}$ is a smooth, compact surface without boundary, the boundary $\partial \Omega_k$ of the region Ω_k consists of one or more curves Γ_i, $i \in \{1, \ldots, N_C\}$. Thus, locally, $\boldsymbol{\mu}_k$ is $\pm \boldsymbol{\nu}_{\mathcal{M},i}$. Since v is arbitrarily chosen, we have to solve the following surface partial differential equation with Neumann boundary condition for $k = 1, \ldots, N_R$:

$$
-\frac{1}{\lambda} \Delta_{\mathcal{M}} u_k + u_k = u_0, \quad \text{in} \quad \Omega_k, \tag{23a}
$$

$$
\nabla_{\mathcal{M}} u_k \cdot \boldsymbol{\mu}_k = 0, \quad \text{on} \quad \partial \Omega_k. \tag{23b}
$$

The smoothing effect is due to the Laplace–Beltrami operator. We can control the smoothing extent using the weighting parameter $\lambda > 0$. The smaller the λ, the larger is the denoising. The larger the λ, the closer is the approximation to the original image. The Neumann boundary condition provides that edges in the image are not smoothed out.

Vector-valued images can be smoothed by considering each component individually like a scalar image.

3 Numerical Approximation

3.1 Finite Element Approximation of the Image Segmentation Scheme

We introduce a finite element approximation for the scheme (18) with (21) in the case of triple junctions. The evolution equations (18a), for $i = 1, \ldots, N_C$, can be interpreted as a weighted geodesic curvature flow with external forcing terms. Therefore, we make use of the finite element scheme for geodesic curvature flow developed in [4] for closed curves and generalize the approach for possible open curves with triple junctions and for image segmentation problems.

In order to formulate a finite element scheme, we first introduce a spatial and time discretization, and discrete function spaces and discrete inner products.

For $i = 1, \ldots, N_C$, let $0 = q_0^i < q_1^i < \ldots < q_{N_i}^i = 1$ be a decomposition of the interval $I_i = I = [0, 1]$. If Γ_i is a closed curve, we make use of the periodicity $N_i = 0$, $N_i + 1 = 1$, $-1 = N_i - 1$, etc.

We introduce the following discrete function spaces:

$$
W^h := \left\{ (\eta_1, \ldots, \eta_{N_C}) \in [C(I, \mathbb{R})]^{N_C} : \eta_i|_{\left[q_{j-1}^i, q_j^i\right]} \text{ is} \right.
$$

linear, $\forall i = 1, \ldots, N_C, \ j = 1, \ldots, N_i \Big\}$, (24a)

$$\underline{V}^h := \Big\{ (\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_{N_C}) \in \big[C(I, \mathbb{R}^3) \big]^{N_C} : \boldsymbol{\eta}_{i_{k,1}}(\rho_{k,1})$$

$$= \boldsymbol{\eta}_{i_{k,2}}(\rho_{k,2}) = \boldsymbol{\eta}_{i_{k,3}}(\rho_{k,3}), \ \forall k = 1, \ldots, N_T,$$

$$\boldsymbol{\eta}_i |_{[q_{j-1}^i, q_j^i]} \text{ is linear}, \ \forall i = 1, \ldots, N_C,$$

$$j = 1, \ldots, N_i \Big\}. \tag{24b}$$

The attachment conditions for triple junctions are incorporated in the definition of the space $\underline{V}^h$.

A basis of W^h is given by functions $\chi_{i,j} := ((\chi_{i,j})_1, \ldots, (\chi_{i,j})_{N_C}) \in W^h$, where $(\chi_{i,j})_k(q_l^k) := \delta_{ik}\delta_{jl}$ for $i, k = 1, \ldots, N_C, \ j = j_0^i, \ldots, N_i, \ l = j_0^k, \ldots, N_k$, where $j_0^i = 1$ if Γ_i is closed and $j_0^i = 0$ else.

Further, let $0 = t_0 < t_1 < \ldots < t_M = T$ be a partitioning of the time interval $[0, T]$ into possibly variable time steps $\tau_m := t_{m+1} - t_m, m = 0, \ldots, M - 1$.

Let $\mathbf{X}^m = (\mathbf{X}_1^m, \ldots, \mathbf{X}_{N_C}^m) \in \underline{V}^h$ be an approximation of $\mathbf{x}(., t_m) = (\mathbf{x}_1(., t_m), \ldots, \mathbf{x}_{N_C}(., t_m))$. Let $\Gamma^m = (\Gamma_1^m, \ldots, \Gamma_{N_C}^m)$ denote the image of $\mathbf{X}^m$.

In case of triple junctions, let $j_{k,l} \in \{0, N_{i_{k,l}}\}$ denote the index of the corresponding curve endpoint such that $q_{j_{k,l}}^{i_{k,l}} = \rho_{k,l}, k = 1, \ldots, N_T, l = 1, 2, 3$.

For scalar or vector functions $u = (u_1, \ldots, u_{N_C}), v = (v_1, \ldots, v_{N_C}) \in \big[L^2(I, \mathbb{R}^{(3)}) \big]^{N_C}$, the L^2-inner product $\langle \cdot, \cdot \rangle_m$ over the current polygonal curve network Γ^m is given by

$$\langle u, v \rangle_m := \int_{\Gamma^m} u \cdot v \, \mathrm{d}s := \sum_{i=1}^{N_C} \int_{I_i} u_i \cdot v_i \, \|(\mathbf{X}_i^m)_\rho\| \, \mathrm{d}\rho. \tag{25}$$

We follow the ideas of [2–4], and define a mass lumped inner product $\langle \cdot, \cdot \rangle_m^h$ for piecewise continuous functions $u = (u_1, \ldots, u_{N_C})$ and $v = (v_1, \ldots, v_{N_C})$ by

$$\langle u, v \rangle_m^h := \frac{1}{2} \sum_{i=1}^{N_C} \sum_{j=1}^{N_i} h_{i,j-\frac{1}{2}}^m \Big[(u_i \cdot v_i) \left([q_j^i]^- \right)$$

$$+ (u_i \cdot v_i) \left([q_{j-1}^i]^+ \right) \Big], \tag{26}$$

where $u_i([q_j^i]^\pm) := \lim_{\epsilon \to 0, \epsilon > 0} u_i(q_j^i \pm \epsilon)$ and $h_{i,j-\frac{1}{2}}^m := \|\mathbf{X}_i^m(q_j^i) - \mathbf{X}_i^m(q_{j-1}^i)\| > 0$ is the distance between two neighbor nodes.

Let $h^m := \max_{i=1,\ldots,N_C, j=1,\ldots,N_i} h_{i,j-\frac{1}{2}}^m$ denote the maximum distance between two neighbor nodes of the polygonal curves.

Let $\mathbf{X}^m \in \underline{V}^h$ be a given parameterization of the polygonal curve network Γ^m satisfying the following assumption:

($\mathcal{A}$) The distance between two neighbor nodes of Γ^m is positive, i.e., $h_{i,j-\frac{1}{2}}^m > 0$ for $i = 1, \ldots, N_C, j = 1, \ldots, N_i$ and $\mathbf{X}_i^m(q_{j+1}^i) \neq \mathbf{X}_i^m(q_{j-1}^i)$ for $i = 1, \ldots, N_C$ and $j = 1, \ldots, N_i$ if $\partial \Gamma_i^m = \emptyset$ and $j = 1, \ldots, N_i - 1$ if $\partial \Gamma_i^m \neq \emptyset$.

We define $\boldsymbol{\omega}_\Phi^m = (\boldsymbol{\omega}_{\Phi,1}^m, \ldots, \boldsymbol{\omega}_{\Phi,N_C}^m)$, by

$$\boldsymbol{\omega}_{\Phi,i}^m(q_j^i) = \mathbf{n}_\Phi(\mathbf{X}_i^m(q_j^i)) = \frac{\nabla \Phi(\mathbf{X}_i^m(q_j^i))}{\|\nabla \Phi(\mathbf{X}_i^m(q_j^i))\|},$$

for $i = 1, \ldots, N_C$ and $j = j_0^i, \ldots, N_i$, i.e., $\boldsymbol{\omega}_{\Phi,i}^m$ approximates $\boldsymbol{v}_{\Phi,i}$ at time t_m. Further, the tangent vector field is approximated by $\boldsymbol{\omega}_d^m = (\boldsymbol{\omega}_{d,1}^m, \ldots, \boldsymbol{\omega}_{d,N_C}^m)$: We set

$$\boldsymbol{\omega}_{d,i}^m(q_j^i) = \frac{\mathbf{X}_i^m(q_{j+1}^i) - \mathbf{X}_i^m(q_{j-1}^i)}{\|\mathbf{X}_i^m(q_{j+1}^i) - \mathbf{X}_i^m(q_{j-1}^i)\|}$$

if Γ_i^m is closed, or if Γ_i^m is an open curve and $j \neq 0, N_i$. For closed curves, we make use of the periodicity $N_i = 0$, $N_i + 1 = 1$ and $-1 = N_i - 1$. For the endpoints of an open curve, we define

$$\boldsymbol{\omega}_{d,i}^m(q_0^i) = \frac{\mathbf{X}_i^m(q_1^i) - \mathbf{X}_i^m(q_0^i)}{\|\mathbf{X}_i^m(q_1^i) - \mathbf{X}_i^m(q_0^i)\|},$$

$$\boldsymbol{\omega}_{d,i}^m(q_{N_i}^i) = \frac{\mathbf{X}_i^m(q_{N_i}^i) - \mathbf{X}_i^m(q_{N_i-1}^i)}{\|\mathbf{X}_i^m(q_{N_i}^i) - \mathbf{X}_i^m(q_{N_i-1}^i)\|}.$$

Furthermore, we define $\boldsymbol{\omega}_\mathcal{M}^m = (\boldsymbol{\omega}_{\mathcal{M},1}^m, \ldots, \boldsymbol{\omega}_{\mathcal{M},N_C}^m)$ by

$$\boldsymbol{\omega}_{\mathcal{M},i}^m(q_j^i) = \boldsymbol{\omega}_{d,i}^m(q_j^i) \times \boldsymbol{\omega}_{\Phi,i}^m(q_j^i).$$

Thus, $\boldsymbol{\omega}_{\mathcal{M},i}^m$ approximates $\boldsymbol{v}_{\mathcal{M},i}$ at time t_m.

The assumption ($\mathcal{A}$) is necessary, such that $\boldsymbol{\omega}_d^m$ is well defined, see also [4].

We define a discrete analog to the space $\underline{V}_\Phi$ by

$$\underline{V}_\Phi^h = \Big\{ \boldsymbol{\eta} \in \underline{V}^h : \boldsymbol{\eta}_i \cdot \boldsymbol{\omega}_{\Phi,i}^m = 0, \quad i = 1, \ldots, N_C \Big\}. \tag{27}$$

We now propose the following discrete scheme: Let $\mathbf{X}^0 \in \underline{V}^h$ be a given parameterization of a polygonal curve network Γ^0. We assume that the initial nodes $\mathbf{X}_i^0(q_j^i)$ lie on the surface $\mathcal{M}$. Further, we assume that assumption ($\mathcal{A}$) holds for $\mathbf{X}^m$, $m = 0, \ldots, M - 1$.

For $m = 0, \ldots, M - 1$, find $\delta\mathbf{X}^{m+1} \in \underline{V}_\Phi^h$ and $\kappa_\mathcal{M}^{m+1} \in W^h$ such that

$$\Big\langle \frac{\delta\mathbf{X}^{m+1}}{\tau_m}, \chi \, \boldsymbol{\omega}_\mathcal{M}^m \Big\rangle_m^h - \sigma \Big\langle \kappa_\mathcal{M}^{m+1}, \chi \Big\rangle_m^h$$

$$= \langle F^m, \chi \rangle_m^h, \quad \forall \chi \in W^h, \tag{28a}$$

$$\langle \kappa_\mathcal{M}^{m+1} \, \boldsymbol{\omega}_\mathcal{M}^m, \boldsymbol{\eta} \rangle_m^h + \langle \nabla_s \delta\mathbf{X}^{m+1}, \nabla_s \boldsymbol{\eta} \rangle_m$$

$$= -\langle \nabla_s \mathbf{X}^m, \nabla_s \boldsymbol{\eta} \rangle_m, \quad \forall \boldsymbol{\eta} \in \underline{V}_\Phi^h, \tag{28b}$$

where $F^m = (F_1^m, \ldots, F_{N_C}^m) \in W^h$, with F_i^m, $i = 1, \ldots, N_C$, being the piecewise linear function uniquely given by

$$F_i^m(q_j^i) = \lambda \left[\left(u_0(\mathbf{X}_i^m(q_j^i)) - c_{k+(i)}^m \right)^2 + \right.$$
$$\left. - \left(u_0(\mathbf{X}_i^m(q_j^i)) - c_{k-(i)}^m \right)^2 \right], \tag{29}$$

where $c_{k\pm(i)}^m$ are approximations of the coefficients $c_{k\pm(i)}$ at t_m, cf. (17). We will later state how the coefficients c_k^m, $k = 1, \ldots, N_R$, can be computed.

Having found $\delta\mathbf{X}^{m+1} \in \underline{V}_\Phi^h$, we set $\mathbf{X}^{m+1} := \delta\mathbf{X}^{m+1} + \mathbf{X}^m \in \underline{V}^h$.

Before we proceed to prove existence and uniqueness of a solution of the scheme (28), we state some very mild assumptions.

($\mathcal{A}_1$) Let $i \in \{1, \ldots, N_C\}$. If $\partial\Gamma_i^m = \emptyset$, we assume that $\dim \mathrm{span}\{\boldsymbol{\omega}_{\mathcal{M},i}^m(q_j^i), \boldsymbol{\omega}_{\Phi,i}^m(q_j^i)\}_{j=1}^{N_i} = 3$.

($\mathcal{A}_2$) For each $k \in \{1, \ldots, N_T\}$, we assume that

$$\dim \mathrm{span}\{\{\boldsymbol{\omega}_{\mathcal{M},i_{k,l}}^m(q_j^{i_{k,l}}), \boldsymbol{\omega}_{\Phi,i_{k,l}}^m(q_j^{i_{k,l}})\}_{j=1}^{N_{i_{k,l}}-1}\}_{l=1}^3 = 3.$$

The two assumptions are violated only in very rare cases. An example where ($\mathcal{A}_1$) is violated is the following (see [2], Remark 2.2): Let a surface be locally flat, and Γ_i^m a polygonal curve such that the normal vectors to the surface $\boldsymbol{\omega}_{\Phi,i}^m(q_j^i)$, $j = 1, \ldots N_i$, evaluated at the mesh points, span a one-dimensional space, i.e., $\dim \mathrm{span}\{\boldsymbol{\omega}_{\Phi,i}^m(q_j^i)\}_{j=1}^{N_i} = 1$. On the locally flat surface, we assume an even number of nodes, where the nodes $\mathbf{X}_i^m(q_1^i)$, $\mathbf{X}_i^m(q_3^i)$, $\mathbf{X}_i^m(q_5^i)$, ... lie on one straight line, and the nodes $\mathbf{X}_i^m(q_2^i)$, $\mathbf{X}_i^m(q_4^i)$, $\mathbf{X}_i^m(q_6^i)$, ... lie on another straight line, parallel to the first one. Using the definitions of $\boldsymbol{\omega}_{d,i}$ and $\boldsymbol{\omega}_{\mathcal{M},i}$, the vectors $\boldsymbol{\omega}_{\mathcal{M},i}(q_j^i)$ span a one-dimensional space. The assumption ($\mathcal{A}_1$) is violated, since $\dim \mathrm{span}\{\boldsymbol{\omega}_{\mathcal{M},i}^m(q_j^i), \boldsymbol{\omega}_{\Phi,i}^m(q_j^i)\}_{j=1}^{N_i} = 2$, The constructed curve is a zig-zagging line with intersections. Similarly, assumption ($\mathcal{A}_2$) is only violated in very rare occasions.

Theorem 1 *Let the assumptions ($\mathcal{A}$), ($\mathcal{A}_1$), and ($\mathcal{A}_2$) hold. Then there exists a unique solution $(\delta\mathbf{X}^{m+1}, \kappa_{\mathcal{M}}^{m+1}) \in \underline{V}_\Phi^h \times W^h$ to the system (28).*

Proof The system (28) is linear. Therefore, existence of a solution follows from its uniqueness. To prove uniqueness, consider the following system: Find $\{\mathbf{X}, \kappa_{\mathcal{M}}\} \in \underline{V}_\Phi^h \times W^h$ such that

$$\langle \mathbf{X}, \chi \boldsymbol{\omega}_{\mathcal{M}}^m \rangle_m^h - \sigma \tau_m \langle \kappa_{\mathcal{M}}, \chi \rangle_m^h = 0, \quad \forall \chi \in W^h, \tag{30a}$$
$$\langle \kappa_{\mathcal{M}} \boldsymbol{\omega}_{\mathcal{M}}^m, \boldsymbol{\eta} \rangle_m^h + \langle \nabla_s \mathbf{X}, \nabla_s \boldsymbol{\eta} \rangle_m = 0, \quad \forall \boldsymbol{\eta} \in \underline{V}_\Phi^h. \tag{30b}$$

We obtain choosing $\chi = \kappa_{\mathcal{M}} \in W^h$ in (30a) and $\boldsymbol{\eta} = \mathbf{X} \in \underline{V}_\Phi^h$ in (30b)

$$\sigma \tau_m \langle \kappa_{\mathcal{M}}, \kappa_{\mathcal{M}} \rangle_m^h + \langle \nabla_s \mathbf{X}, \nabla_s \mathbf{X} \rangle_m = 0.$$

From this equation, we conclude $\kappa_{\mathcal{M}} \equiv 0$ and $\mathbf{X} \equiv \mathbf{X}^c$ for a constant $\mathbf{X}^c = (\mathbf{X}_1^c, \ldots, \mathbf{X}_{N_C}^c) \in (\mathbb{R}^3)^{N_C}$ with $\mathbf{X}_{i_{k,1}}^c = \mathbf{X}_{i_{k,2}}^c = \mathbf{X}_{i_{k,3}}^c$ for all $k \in \{1, \ldots, N_T\}$. Further, $\mathbf{X}_i^c \in \mathbb{R}^3$ satisfies

$$\mathbf{X}_i^c \cdot \boldsymbol{\omega}_{\Phi,i}^m(q_j^i) = 0 \tag{31}$$

for all $i = 1, \ldots, N_C$ and $j = j_0^i, \ldots, N_i$, since $\mathbf{X} \in \underline{V}_\Phi^h$. Inserting $\kappa \equiv 0$ and $\mathbf{X} \equiv \mathbf{X}^c$, (30a) reduces to

$$\langle \mathbf{X}^c, \chi \boldsymbol{\omega}_{\mathcal{M}}^m \rangle_m^h = 0, \quad \forall \chi \in W^h. \tag{32}$$

We now choose $\chi = \chi_{i,j} \in W^h$, $i \in \{1, \ldots, N_C\}$, $j \in \{j_0^i, \ldots, N_i\}$ in (32). This yields

$$\mathbf{X}_i^c \cdot \boldsymbol{\omega}_{\mathcal{M},i}^m(q_j^i) = 0. \tag{33}$$

We conclude $\mathbf{X}_i^c \equiv 0$ using (31), (33), and the assumptions ($\mathcal{A}_1$) and ($\mathcal{A}_2$). □

3.2 Solution of the Discrete System

Let $N = \sum_{i=1}^{N_C} N_i^*$, with $N_i^* = N_i$ for closed curves and $N_i^* = N_i + 1$ for open curves. We make use of a small abuse of notation and consider functions in W^h as elements in $\mathbb{R}^N$ and functions in $\underline{V}^h$ as elements in

$$\mathbb{X} = \Big\{ (\mathbf{z}_1, \ldots, \mathbf{z}_{N_C}) \in (\mathbb{R}^3)^N : [\mathbf{z}_{i_{k,1}}]_{j_{k,1}} = [\mathbf{z}_{i_{k,2}}]_{j_{k,2}} = $$
$$= [\mathbf{z}_{i_{k,3}}]_{j_{k,3}}, k = 1, \ldots N_T \Big\},$$

where $\mathbf{z}_i \in (\mathbb{R}^3)^{N_i^*}$ and $[\mathbf{z}_i]_j \in \mathbb{R}^3$ is the j-th component of the vector $\mathbf{z}_i$. Functions in $\underline{V}_\Phi^h$ are considered as elements in

$$\mathbb{X}_\Phi = \Big\{ (\mathbf{z}_1, \ldots, \mathbf{z}_{N_C}) \in \mathbb{X} : [\mathbf{z}_i]_j \cdot \boldsymbol{\omega}_{\Phi,i}^m(q_j^i) = 0, $$
$$i = 1, \ldots, N_C, j = j_0^i, \ldots, N_i \Big\},$$

with $j_0^i = 0$ for open curves and $j_0^i = 1$ for closed curves. Let $\mathbf{P}_\Phi : (\mathbb{R}^3)^N \to \mathbb{X}_\Phi$ denote the orthogonal projection onto the space $\mathbb{X}_\Phi$.

In order to state a matrix formulation for the discrete system (28), we introduce the following matrices:

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

$$\mathbf{N}_{\mathcal{M}} := \begin{pmatrix} \mathbf{N}_{\mathcal{M}}^1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{N}_{\mathcal{M}}^{N_C} \end{pmatrix},$$

$$\mathbf{A} := \begin{pmatrix} \mathbf{A}^1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{A}^{N_C} \end{pmatrix},$$

where $M^i \in \mathbb{R}^{N_i^* \times N_i^*}$, $\mathbf{N}_{\mathcal{M}}^i \in (\mathbb{R}^3)^{N_i^* \times N_i^*}$, and $\mathbf{A}^i \in (\mathbb{R}^{3\times3})^{N_i^* \times N_i^*}$, $i = 1, \ldots, N_C$, are defined by

$$M_{jl}^i := \langle \chi_{i,j}, \chi_{i,l} \rangle_m^h, \quad (\mathbf{N}_{\mathcal{M}}^i)_{jl} := \langle \chi_{i,j}, \chi_{i,l} \, \omega_{\mathcal{M}}^m \rangle_m^h,$$
$$\mathbf{A}_{jl}^i := \langle \nabla_s \chi_{i,j}, \nabla_s \chi_{i,l} \rangle_m \, \mathbf{Id}_3,$$

where $\mathbf{Id}_3$ denotes the 3×3 identity matrix. We define $b^m = (b_1^m, \ldots, b_{N_C}^m) \in \mathbb{R}^N$ by

$$b_i^m = \left(b_{i,j_0^i}^m, \ldots, b_{i,N_i}^m \right), \quad \text{with} \quad b_{i,j}^m := \langle F_i^m, \chi_{i,j} \rangle_m^h,$$
$$i = 1, \ldots, N_C, \; j = j_0^i, \ldots, N_i. \tag{34}$$

The discrete system (28) can be rewritten into the following matrix–vector formulation: Find $\kappa_{\mathcal{M}}^{m+1} \in \mathbb{R}^N$ and $\delta \mathbf{X}^{m+1} \in \mathbb{X}_\Phi$, such that

$$\begin{pmatrix} -\sigma \tau_m M & \mathbf{N}_{\mathcal{M}}^T \mathbf{P}_\Phi \\ \mathbf{P}_\Phi \mathbf{N}_{\mathcal{M}} & \mathbf{P}_\Phi \mathbf{A} \mathbf{P}_\Phi \end{pmatrix} \begin{pmatrix} \kappa_{\mathcal{M}}^{m+1} \\ \delta \mathbf{X}^{m+1} \end{pmatrix} = \begin{pmatrix} \tau_m b^m \\ -\mathbf{P}_\Phi \mathbf{A} \mathbf{X}^m \end{pmatrix} \tag{35}$$

holds, on assuming that $\mathbf{X}^0 \in \mathbb{X}$.

Since M is non-singular, we can apply a Schur complement approach and obtain

$$\kappa_{\mathcal{M}}^{m+1} = \frac{1}{\sigma \tau_m} M^{-1} \left(\mathbf{N}_{\mathcal{M}}^T \mathbf{P}_\Phi \, \delta \mathbf{X}^{m+1} - \tau_m b^m \right), \tag{36a}$$

$$\left(\mathbf{P}_\Phi \mathbf{A} \mathbf{P}_\Phi + \frac{1}{\sigma \tau_m} \mathbf{P}_\Phi \mathbf{N}_{\mathcal{M}} M^{-1} \mathbf{N}_{\mathcal{M}}^T \mathbf{P}_\Phi \right) \delta \mathbf{X}^{m+1}$$
$$= \frac{1}{\sigma} \mathbf{P}_\Phi \mathbf{N}_{\mathcal{M}} M^{-1} b^m - \mathbf{P}_\Phi \mathbf{A} \mathbf{X}^m. \tag{36b}$$

Since $\mathbf{P}_\Phi$ is a projection to a subspace of $(\mathbb{R}^3)^N$, the system matrix of the linear equation (36b) is singular as a mapping of $(\mathbb{R}^3)^N \to (\mathbb{R}^3)^N$. However, considered as a mapping of $\mathbb{X}_\Phi \to \mathbb{X}_\Phi$ it is non-singular if the assumptions $(\mathcal{A}_1)$ and $(\mathcal{A}_2)$ hold.

Since the system matrix is sparse, (36b) can be efficiently solved with linear effort using an iterative solver (with possible preconditioning) or using a direct solver for sparse matrices. In the examples presented later in Sect. 4, we use the UMFPACK algorithm [13] (direct solver) as MATLAB built-in routine for sparse linear systems.

In the following, we will use the abbreviation $\mathbf{X}_{i,j}^m := \mathbf{X}_i^m(q_j^i)$.

3.3 Semidiscrete Scheme

We consider a scheme which is discrete in space and continuous in time. We consider time-dependent polygonal curves $\Gamma_i(t)$, $i = 1, \ldots, N_C$, $t \in [0, T]$, and show an equidistribution property concerning the distribution of mesh points along the curves on surfaces.

We make use of a similar notation as in the fully discrete case by just omitting the superscripts m and $m + 1$. In detail, let $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_{N_C})$ and $\kappa = (\kappa_1, \ldots, \kappa_{N_C})$ with $\mathbf{X}_i : I_i \times [0, T] \to \mathbb{R}^3$, and $\kappa_i : I_i \times [0, T] \to \mathbb{R}$, $i = 1, \ldots, N_C$, such that $\mathbf{X}(\,.\,, t)$ and $\kappa(\,.\,, t)$ are piecewise linear on $[q_{j-1}^i, q_j^i]$, $j = 1, \ldots, N_i$, for each $t \in [0, T]$. Further, we set $\mathbf{X}_{i,j} = \mathbf{X}_i(q_j^i)$ and $h_{i,j-\frac{1}{2}} = \|\mathbf{X}_{i,j} - \mathbf{X}_{i,j-1}\|$ for $i = 1, \ldots, N_C$, $j = 1, \ldots, N_i$. We will make use of inner products $\langle \,.\,, . \rangle$ and $\langle \,.\,, . \rangle^h$ which are defined as in (25) and (26) by replacing $\mathbf{X}^m$ by $\mathbf{X}(\,.\,, t)$ and Γ_i^m by the current polygonal curve $\Gamma_i(t)$, $i = 1, \ldots, N_C$, $t \in [0, T]$. We make use of vector fields ω_Φ, ω_d, and $\omega_{\mathcal{M}}$ which are defined similar as in the fully discrete case by omitting the superscript m. The space $\underline{V}_\Phi^h$ is defined as in (27) using ω_Φ instead of ω_Φ^m.

Theorem 2 (See also [4]) *The semidiscrete scheme*

$$\langle \mathbf{X}_t, \chi \, \omega_{\mathcal{M}} \rangle^h - \sigma \langle \kappa_{\mathcal{M}}, \chi \rangle^h = \langle F, \chi \rangle^h, \quad \forall \chi \in W^h, \tag{37a}$$

$$\langle \kappa_{\mathcal{M}} \, \omega_{\mathcal{M}}, \eta \rangle^h + \langle \nabla_s \mathbf{X}, \nabla_s \eta \rangle = 0, \quad \forall \eta \in \underline{V}_\Phi^h, \tag{37b}$$

for $\kappa \in W^h$ and $\mathbf{X} \in \underline{V}_\Phi^h$, provides an equidistribution of the mesh points along the curves $\Gamma_i(t)$, $i = 1, \ldots, N_C$, $t \in [0, T]$.

Proof Testing (37b) with $\eta = \chi_{i,j} \, \omega_{d,i}(q_j^i)$, $i = 1, \ldots, N_C$, $j = 1, \ldots, N_i$ if $\partial \Gamma_i(t) = \emptyset$ and $j = 1, \ldots, N_i - 1$ else, leads to

$$0 = \left(\frac{\mathbf{X}_{i,j+1} - \mathbf{X}_{i,j}}{h_{i,j+\frac{1}{2}}} - \frac{\mathbf{X}_{i,j} - \mathbf{X}_{i,j-1}}{h_{i,j-\frac{1}{2}}} \right) \cdot$$
$$(\mathbf{X}_{i,j+1} - \mathbf{X}_{i,j-1}),$$

where we use that $\boldsymbol{\omega}_{\mathcal{M}}$ and $\boldsymbol{\omega}_d$ are perpendicular. From this equation, we can conclude that (see also [2], Remark 2.3)

$\|\mathbf{X}_{i,j+1} - \mathbf{X}_{i,j}\| = \|\mathbf{X}_{i,j} - \mathbf{X}_{i,j-1}\|$ or
$(\mathbf{X}_{i,j+1} - \mathbf{X}_{i,j})$ is parallel to $(\mathbf{X}_{i,j} - \mathbf{X}_{i,j-1})$.

Consequently, three neighbor nodes are equally distributed or lie on one straight line. □

We could show that the Euclidean distance between mesh points is equal (or three neighbor nodes lie on one straight line). We did not show an equidistribution using the geodesic distance, the length of the smallest curve on the surface connecting two neighbor nodes. However, for most practical examples, we will also obtain a good mesh quality concerning the distribution of nodes with respect to the geodesic distance.

As in [4], we cannot prove equidistribution for the fully discrete scheme. However in experiments, we always observe a good mesh quality during the evolution of the curves on the surface.

Equation (37a) has not been used to prove equidistribution of the mesh points; only (37b) has been used. Therefore, the equidistribution property is also valid for other flows; see [4] where different flows of curves on surfaces are presented.

3.4 Topology Changes

Our scheme is based on a parametrization of evolving curves on surfaces. Topology changes concerning the curves are not automatically handled which is often considered as the main drawback of parametric methods.

There are different topology changes that can occur: A curve can split into two sub-curves (splitting), two curves separating the same regions can merge to one single curve (merging), two curves separating different regions can touch and a new curve and two new triple junctions occur (creation of triple junctions) and a curve can shrink and has to be deleted.

The latter can be simply detected by considering the length of the curve. If the length of a curve is smaller than a predefined tolerance, it will be deleted. The other topology changes will occur if two points from different curves or two points from one curve which are not neighbors have a small distance. A simple comparison of all nodes would lead to an effort of $\mathcal{O}(N^2)$, where N is the total number of nodes of the polygonal curves $\Gamma_1^m, \ldots, \Gamma_{N_C}^m$. Since we can solve the linear equation of our main algorithm (36b) efficiently with linear effort, a sub-algorithm to detect topology changes should not result in a too large computational effort.

For curves in the plane, we extended an efficient method to detect topology changes [5] which was originally developed by Mikula and Urbán [22], see also [1]. The method to detect topology changes is based on an artificial background grid covering the image domain and consisting of a finite set of arrays (squares). If two nodes from different curves or two nodes from different parts of one curve lie in the same array of the virtual background grid, a topology change likely occurs close to the two points. Using this method, the effort to detect topology changes is $\mathcal{O}(N)$.

In principle, the idea of an artificial background grid can be extended to detect topology changes involving curves on surfaces. One can construct a Cartesian 3D background grid around the surface $\mathcal{M}$. Again, we can check whether two points of different curves or different parts of one curve belong to the same array of the background grid.

However, using the Euclidean distance in $\mathbb{R}^3$ to detect topology changes can lead to false detections: Surfaces exist where two points can have a small Euclidean distance, but their geodesic distance is large. In such situations, a topology change does not occur. The geodesic distance would be a better indicator for detection of topology changes compared to the Euclidean distance. However, the computation of the geodesic distance between two points on a surface is very expensive from a computational point of view and cannot be used in a sub-algorithm in practice.

Let $a > 0$ be the grid size of the cubes of the 3D background grid. For extending the method to detect topology changes from the planar case [5] to the case of curves on surfaces, we have to choose the grid size a small enough to exclude such wrong detections as described above.

For $\mathbf{p} \in \mathcal{M}$, let $T_{\mathbf{p}}\mathcal{M}$ denote the tangent space and $N_{\mathbf{p}}\mathcal{M} = (T_{\mathbf{p}}\mathcal{M})^{\perp}$ the normal space. Let $N\mathcal{M} = \{(\mathbf{p}, \mathbf{n}) : \mathbf{p} \in \mathcal{M}, \mathbf{n} \in N_{\mathbf{p}}\mathcal{M}\}$ denote the normal bundle.

For the smooth, embedded hypersurface, we consider the map

$$E : N\mathcal{M} \to \mathbb{R}^3, \quad (\mathbf{p}, \mathbf{n}) \mapsto \mathbf{p} + \mathbf{n}.$$

Theorem 3 (Tubular neighborhood theorem) *Every embedded hypersurface $\mathcal{M}$ of $\mathbb{R}^3$ has a tubular neighborhood, i.e., a neighborhood $U \subset \mathbb{R}^3$ that is the diffeomorphic image under $E : N\mathcal{M} \to \mathbb{R}^3$ of an open subset $V \subset N\mathcal{M}$ of the form:*

$$V = \{(\mathbf{p}, \mathbf{n}) \in N\mathcal{M} : \|\mathbf{n}\| < \delta(\mathbf{p})\}, \tag{38}$$

for some positive continuous function $\delta : \mathcal{M} \to \mathbb{R}$.

Proof See [21], Chapter 6, Embedding and Approximation Theorems, or [20], Chapter 4, Vector Fields and Differential Equations. □

For images on surfaces, we assume the surface $\mathcal{M}$ to be a compact, embedded hypersurface. As a consequence, set $\delta_0 = \min\{\delta(\mathbf{p}) : \mathbf{p} \in \mathcal{M}\} > 0$. For each $\mathbf{p} \in \mathcal{M}$, the intersection $B_{\delta_0}(\mathbf{p}) \cap \mathcal{M}$ is simply connected, which is a

consequence of the fact that $E|_V : V \to U$ is a diffeomorphism.

Thus, the key idea when extending the algorithm from the planar case to curves on surfaces is to choose the grid size a of the auxiliary 3D background grid small enough with respect to δ_0, such that points from two different parts of the surface (with nearly opposite normal vector $\boldsymbol{\nu}_\Phi$) cannot lie in one array of the grid.

It is possible to establish the auxiliary 3D grid only locally around the surface to reduce memory requirements. A theoretical alternative option to a Cartesian 3D grid would be a non-Cartesian 2D grid constructed over the surface. However, a Cartesian 3D grid is much simpler to handle. If it is only created in a small neighborhood of the surface, the memory requirements of such a cubic grid are comparable with those of a 2D grid.

Topology changes are now detected as follows:

– Construct an underlying 3D grid with grid size a with $a\sqrt{3} < \delta_0$. Note that the intersection of a grid element (=cube of grid length a) with $\mathcal{M}$ is simply connected.
– Mark the grid elements with the indices of the curves and the mesh points: We successively consider the mesh points $\mathbf{X}_{i,j}^m, i = 1, \ldots, N_C, j = j_0^i, \ldots, N_i$. If the corresponding grid array, in which $\mathbf{X}_{i,j}^m$ lies, is empty, the grid is marked with (i, j).
– If a grid array is already marked with (i_1, j_1) and if $\mathbf{X}_{i,j}^m$ and $\mathbf{X}_{i_1,j_1}^m$ are no neighbor nodes, a topology change is detected.
– Since $\mathbf{X}_{i,j}^m$ and $\mathbf{X}_{i_1,j_1}^m$ may not be the pair with the smallest distance, we consider a few neighbor nodes around $\mathbf{X}_{i,j}^m$ and $\mathbf{X}_{i_1,j_1}^m$. Let $\mathbf{X}_{i,l}^m$ and $\mathbf{X}_{i_1,l_1}^m$ be the pair with the smallest Euclidean distance in these two small groups of nodes. They can be found by a pairwise comparison, which is not computationally expensive since only a few nodes are involved.

The topology changes splitting, merging, and creation of triple junctions (see explanations above) are distinguished as follows:

– If $i = i_1$, a splitting of the curve Γ_i^m is detected.
– If $i \neq i_1$, we consider the regions separated by Γ_i^m and $\Gamma_{i_1}^m$: If $k^+(i) = k^+(i_1) \wedge k^-(i) = k^-(i_1)$, or alternatively $k^+(i) = k^-(i_1) \wedge k^-(i) = k^+(i_1)$ holds, a merging occurs.
– Otherwise, a creation of a new contour and a creation of two new triple junctions occur.

After having detected and identified the topology change, the curves need to be adapted near $\mathbf{X}_{i,l}^m$ and $\mathbf{X}_{i_1,l_1}^m$. This involves changing the neighbor relations, changing curve indices in case of merging or splitting, and creation of a small

new contour with a few nodes in case of triple junctions. Details are given in [5].

In case of triple junctions, a new curve is created. When creating new nodes, one has to ensure that these nodes lie on the surface $\mathcal{M}$. In the next section, we describe how nodes can be efficiently projected to the surface.

It has to be noted that the topology changes can be detected robustly since the mesh points of the curves are nearly equidistributed (cf. Sect. 3.3 and [4]). Therefore, false-positive or false-negative errors in the detection of topology changes cannot occur, since there are no locations along the curve where mesh points locally bunch together or where mesh points locally drift apart.

3.5 Additional Computational Aspects

Triangulated surfaces In practical applications, a smooth function $\Phi : \mathbb{R}^3 \to \mathbb{R}$, such that $\mathcal{M}$ is the zero level set of Φ, is usually not provided. Moreover, a surface $\mathcal{M}$ is typically given as a triangulated surface instead of a smooth surface.

Therefore, we assume that $\mathcal{M}$ is a union of triangles of a triangulation $\mathcal{T}^h$, i.e., $\mathcal{M} = \bigcup_{\sigma^h \in \mathcal{T}^h} \overline{\sigma^h}$. Note that the function Φ was only needed to compute $\mathbf{n}_\Phi$. Normal vectors to the surface can now be easily computed for each triangle σ^h. For a point $\mathbf{p}$ on a curve $\Gamma^m \subset \mathcal{M}$, we first need to assign $\mathbf{p}$ to a triangle $\sigma^h \in \mathcal{T}^h$ in which the node lies, to compute $\mathbf{n}_\Phi(\mathbf{p})$. Further, the color data u_0 is often piecewise constant and uniquely given by its value on the triangles. To evaluate $u_0(\mathbf{p})$, we also need to assign the node to its corresponding triangle.

For each simplex, we can project a vector in $\mathbb{R}^3$ to the simplex plane and can use barycentric coordinates to determine if the projected node lies inside the triangle. Surfaces are often composed of 10^5 to 10^6 triangles. Therefore, for a given point, finding the corresponding triangle in which the point lies results in a high computational effort if no additional knowledge is used.

For $m = 0$ and a curve $\Gamma_i^m, i \in \{1, \ldots, N_C\}$, with nodes $\mathbf{X}_{i,j}^m, j = j_0^i, \ldots, N_i$, we perform a global search only for $\mathbf{X}_{i,j_0^i}^m$. For $j > j_0^i$, we consider first the simplex to which $\mathbf{X}_{i,j-1}^m$ has been assigned. If the node $\mathbf{X}_{i,j}^m$ is not located in the same simplex, we start a search considering successively the neighbor simplices. For $m > 0$, we can assume that a node has moved only slightly on the surface from step $m-1$ to m. Therefore, we start the search using the triangle to which the node was assigned in time step $m-1$. Consequently, a global search has to be performed only N_C times at the beginning of the segmentation.

After the linear system (36b) has been solved, some of the nodes may not lie exactly on the surface. For smooth surfaces (like spheres, tori, etc.), the nodes stay very close

to the surface if small time steps are used, see [4]. However, for triangulated surfaces, a reprojection onto the surface is necessary since $\boldsymbol{v}_\Phi$ is not continuous. A reprojection onto the surface does not result in an additional computational effort: In the next time step, we need to compute $\boldsymbol{v}_\Phi = \mathbf{n}_\Phi \circ \mathbf{x}$ and need to evaluate u_0 again for each node. For both, we have to determine again the closest triangle for a point. As described above, this is done by projection of the original node to the triangle plane and by using barycentric coordinates. That is, we already need to determine a projection of the original point to the corresponding triangle.

Computation of regions and coefficients For the external forcing term, we need to determine the regions Ω_k^m, approximations of $\Omega_k(t_m)$, $k = 1, \ldots, N_R$. The regions Ω_k^m are separated and thus determined by the union of discrete curves $\Gamma^m = \Gamma_1^m \cup \ldots \cup \Gamma_{N_C}^m$. Further, we need to compute the coefficients c_k^m which are the average color values of the image function u_0 in the corresponding regions.

For $m = 0$, we need to assign each simplex $\sigma^h \in \mathcal{T}^h$ to a region Ω_k^0. For $m > 0$, we need to update the assignment only in a neighborhood of the curves.

Let $\mathbf{p}_{\sigma^h, j}$, $j = 1, 2, 3$, denote the vertices of a triangle σ_h. We assign the simplex to a region Ω_k^m if its center $\mathbf{p}_{\sigma^h} = (\mathbf{p}_{\sigma^h, 1} + \mathbf{p}_{\sigma^h, 2} + \mathbf{p}_{\sigma^h, 3})/3$ belongs to the region. In the rare case that $\mathbf{p}_{\sigma^h}$ lies directly on a curve Γ_i^m, it is assigned either to $\Omega_{k^+(i)}^m$ or $\Omega_{k^-(i)}^m$. For image segmentation, we do not apply any special treatment to simplices which are truncated by a curve.

For a simplex σ^h close to a curve with center $\mathbf{p}_{\sigma^h}$, we can search for the closest node $\mathbf{X}_{i,j}^m$ and consider the sign of $(\mathbf{p}_{\sigma^h} - \mathbf{X}_{i,j}^m) \cdot \boldsymbol{\omega}_{\mathcal{M}}^m(q_j^i)$.

For $m = 0$, we also need to consider simplices which are not close to a curve. The direction $\boldsymbol{\omega}_{\mathcal{M}}^m(q_j^i)$ cannot be used for remote simplices if the surface $\mathcal{M}$ is curved. Having assigned a small band of simplices around the curves, the remaining simplices can inherit the region index using the neighbor relation between the simplices of the triangulation.

Motivated by these thoughts, we propose the following algorithm for computation of the regions:

– For all nodes $\mathbf{X}_{i,j}^m$, $i = 1, \ldots, N_C$, $j = j_0^i, \ldots, N_i$, we consider the triangle σ^h to which $\mathbf{X}_{i,j}^m$ belongs (see determination of the closest triangle described above). If $(\mathbf{p}_{\sigma^h} - \mathbf{X}_{i,j}^m) \cdot \boldsymbol{\omega}_{\mathcal{M}}^m(q_j^i)$ is positive, the simplex is assigned to $\Omega_{k^+(i)}^m$, otherwise to $\Omega_{k^-(i)}^m$. The indices of neighbor simplices of σ^h are stored in a list.
– We consider the simplices of the auxiliary list, which have not been assigned to a region yet. For a simplex σ^h of the list, we search for the closest node point $\mathbf{X}_{i,j}^m$ and determine a region index using the sign of $(\mathbf{p}_{\sigma^h} - \mathbf{X}_{i,j}^m) \cdot \boldsymbol{\omega}_{\mathcal{M}}^m(q_j^i)$. We store the neighbor simplices of σ^h in a new list.
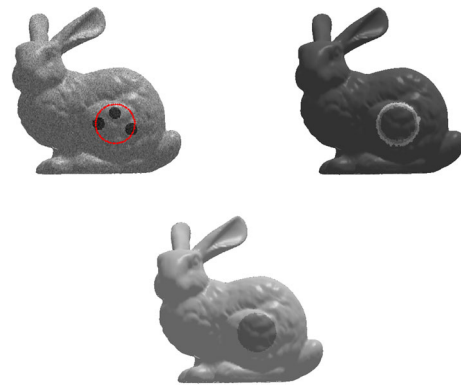

Fig. 2 Illustration of how *triangles* are assigned to a region. *1st sub-figure* image and initial curve. *2nd sub-figure* small band after $n_0 = 4$ steps. *3rd sub-figure* regions *colored* with mean brightness value after assignment of all triangles. The surface data are from the Stanford Computer Graphics Laboratory, cf. [31]

– Having assigned all simplices of the current list to a region, the current list is deleted and the simplices of the new list are considered. We repeat the procedure n_0 times. Afterwards, a small band of simplices around each curve is assigned to regions.
– For $m = 0$, the remaining simplices are considered successively using again lists of neighbor simplices. After the step n_0, we do not determine the closest node $\mathbf{X}_{i,j}^m$. A new simplex inherits the region index directly from its neighbor.

Figure 2 illustrates the assignment of regions for simplices of a triangulated surface. It shows the Stanford Bunny[1] from the Stanford Computer Graphics Laboratory, cf. [31], and an image with three small disks on its surface. We used $n_0 = 4$ levels of neighbor simplices to assign the simplices of a small band around the initial curve to one of the two regions separated by the initial curve. The remaining simplices (marked with dark color in the second sub-figure) are finally assigned to a region by heritage of the region index.

The final coefficients are computed as follows: Let $C_k^m = \sum_{\sigma^h \in \Omega_k^m} u_0|_{\sigma^h}$ denote the sum of the color data and n_k^m the number of simplices belonging to Ω_k^m. The coefficient for the Chan–Vese model is computed by setting $c_k^m = C_k^m / n_k^m$. For color spaces like the CB space, we need to make use of normalized means for some components of the color [5].

For $m > 0$, we need to update the coefficients only close to the curve, i.e., we consider only the simplices of a small band around the curves (using again n_0 levels of neighbor simplices around the curves). If a simplex σ^h changes its region assignment from Ω_l^m to Ω_k^m, we set

$$n_k^m = n_k^m + 1, \quad n_l^m = n_l^m - 1,$$
$$C_k^m = C_k^m + u_0|_{\sigma^h}, \quad C_l^m = C_l^m - u_0|_{\sigma^h}. \tag{39}$$

---

[1] https://graphics.stanford.edu/data/3Dscanrep/.

Global refinement-coarsening strategy We can perform a global refinement and coarsening of the curves using two thresholds $l_{\max}$ and $l_{\min}$ for the average distance between neighboring nodes. Let N_i^m be the number of nodes belonging to a curve Γ_i^m. If $|\Gamma_i^m| / N_i^m > l_{\max}$, we perform a global refinement of the curve by inserting a new node between two neighbor nodes. On the contrary, if $|\Gamma_i^m| / N_i^m < l_{\min}$, we perform a global coarsening, i.e., we delete every second node of the polygonal curve which is not a boundary point.

When inserting a new node between two nodes $\mathbf{X}_{i,j}^m$ and $\mathbf{X}_{i,j+1}^m$ during a refinement, we first compute $\mathbf{p} = (\mathbf{X}_{i,j}^m + \mathbf{X}_{i,j+1}^m)/2$ and determine the closest triangle (again the search for the closest triangle is done efficiently by starting with the triangle in which, e.g., $\mathbf{X}_{i,j}^m$ lies). Having found the closest triangle σ^h, $\mathbf{p}$ is projected orthogonally to σ^h. Consequently, all newly generated nodes lie on the surface.

3.6 Numerical Solution of the Image Restoration Scheme

The scheme (23) for image restoration can be solved numerically with a finite element approach. Again, we consider a polyhedral surface $\mathcal{M}$ given by a set $\mathcal{T}^h$ of triangles.

The image restoration is performed as postprocessing step using the final regions from the time step $m = M$. The surface thus consists of polyhedral regions $\Omega_k^h := \Omega_k^M$, $k = 1, \ldots, N_R$. Let $\mathcal{T}_k^h = \{\sigma^h \in \mathcal{T}^h : \sigma^h \subset \Omega_k^h\}$ denote the set of triangles belonging to Ω_k^h and let $\mathbf{p}_{k,j}$, $j = 1, \ldots, N_k^h$, denote the vertices of the triangles belonging to $\mathcal{T}_k^h$.

For each $k = 1, \ldots, N_R$, we define the following finite element space:

$$S_k^h := \left\{ u^h \in C(\overline{\Omega_k^h}, \mathbb{R}) : u^h_{|\sigma^h} \text{ is linear } \forall \sigma^h \in \mathcal{T}_k^h \right\}. \tag{40}$$

For piecewise continuous functions $u^h, v^h : \Omega_k^h \rightarrow \mathbb{R}^{(3)}$ with possible jumps at edges of simplices $\sigma^h \in \mathcal{T}_k^h$, we define the mass lumped inner product

$$\langle u^h, v^h \rangle^h := \frac{1}{3} \sum_{\sigma^h \in \mathcal{T}_k^h} |\sigma^h| \sum_{j=1}^{3} (u^h \cdot v^h)((\mathbf{p}_{\sigma^h, j})^-), \tag{41}$$

where $|\sigma^h|$ denotes the area of σ^h, and as above $\mathbf{p}_{\sigma^h, j}$, $j = 1, 2, 3$, denote the vertices of the triangle $\sigma^h \in \mathcal{T}_k^h$ and

$$u^h((\mathbf{p}_{\sigma^h, j})^-) := \lim_{\mathbf{p} \to \mathbf{p}_{\sigma^h, j}, \, \mathbf{p} \in \sigma^h} u^h(\mathbf{p}).$$

Further, for functions $u^h, v^h \in L^2(\Omega_k^h, \mathbb{R}^{(3)})$, we define

$$\langle u^h, v^h \rangle := \int_{\Omega_k^h} u^h \cdot v^h \, dA. \tag{42}$$

We consider the following discrete system for each region $k \in \{1, \ldots, N_R\}$: Find $u^h \in S_k^h$ such that

$$\frac{1}{\lambda} \langle \nabla_{\mathcal{M}} u^h, \nabla_{\mathcal{M}} v^h \rangle + \langle u^h, v^h \rangle^h = \langle u_0, v^h \rangle^h, \quad \forall v^h \in S_k^h, \tag{43}$$

where $\lambda > 0$ is a weighting parameter (cf. (4)).

Let $\{\phi_{k,i}^h\}_{i=1}^{N_k^h}$ with $\phi_{k,i}^h(\mathbf{p}_{k,j}) = \delta_{ij}$ denote the standard basis of S_k^h. Using this standard basis, we can identify each element in S_k^h with its coefficient vector in $\mathbb{R}^{N_k^h}$. Further, we define the matrices $M_k^h, A_k^h \in \mathbb{R}^{N_k^h \times N_k^h}$ by

$$(M_k^h)_{ij} := \langle \phi_{k,i}^h, \phi_{k,j}^h \rangle^h,$$
$$(A_k^h)_{ij} := \langle \nabla_{\mathcal{M}} \phi_{k,i}^h, \nabla_{\mathcal{M}} \phi_{k,j}^h \rangle, \quad i, j = 1, \ldots, N_k^h.$$

The entries of the matrices M_k^h and A_k^h are computed by considering each triangle $\sigma^h \in \mathcal{T}_k^h$ and computing the contribution of σ^h to the entries corresponding to the indices of its vertices. For computing the contribution of σ^h to A_k^h, we need to compute surface gradients.

For that, we consider three nodes $\mathbf{p}_{k,j_1}$, $\mathbf{p}_{k,j_2}$, and $\mathbf{p}_{k,j_3}$, $j_1, j_2, j_3 \in \{1, \ldots, N_k^h\}$, being the vertices of a triangle σ^h in $\mathcal{T}_k^h$. For the ease of notation, we assume $j_1 = 1$, $j_2 = 2$, and $j_3 = 3$. One tangential vector is given by

$$\boldsymbol{\tau}_1 := \frac{\mathbf{p}_{k,3} - \mathbf{p}_{k,2}}{\|\mathbf{p}_{k,3} - \mathbf{p}_{k,2}\|}.$$

A second tangential vector which is orthogonal to $\boldsymbol{\tau}_1$ can be obtained by

$$\boldsymbol{\tau}_2 := \frac{\mathbf{p}_{k,1} - \mathbf{q}_k}{\|\mathbf{p}_{k,1} - \mathbf{q}_k\|},$$

with $\mathbf{q}_k = \mathbf{p}_{k,2} + ((\mathbf{p}_{k,1} - \mathbf{p}_{k,2}) \cdot \boldsymbol{\tau}_1) \boldsymbol{\tau}_1$.

We note that $\partial_{\boldsymbol{\tau}_1} \phi_{k,1}^h = 0$. For $\partial_{\boldsymbol{\tau}_2} \phi_{k,1}^h$, we consider a curve $\gamma : [0, \|\mathbf{p}_{k,1} - \mathbf{q}_k\|] \rightarrow \mathbb{R}^3$, $\gamma(\epsilon) = \mathbf{q}_k + \epsilon \boldsymbol{\tau}_2$. The composition $\phi_{k,1}^h \circ \gamma$ is given by

$$(\phi_{k,1}^h \circ \gamma)(\epsilon) = \frac{\epsilon}{\|\mathbf{p}_{k,1} - \mathbf{q}_k\|}.$$

The derivative of $\phi_{k,1}^h$ in direction $\boldsymbol{\tau}_2$ is

$$\partial_{\boldsymbol{\tau}_2} \phi_{k,1}^h = \frac{d}{d\epsilon} \phi_{k,1}(\gamma(\epsilon))|_{\epsilon=0} = \frac{1}{\|\mathbf{p}_{k,1} - \mathbf{q}_k\|}.$$

The surface gradient is then given by

$$\nabla_{\mathcal{M}} \phi_{k,1}^h|_{\sigma^h} = \partial_{\tau_1} \phi_{k,1}^h \, \tau_1 + \partial_{\tau_2} \phi_{k,1}^h \, \tau_2 = \frac{\mathbf{p}_{k,1} - \mathbf{q}_k}{\|\mathbf{p}_{k,1} - \mathbf{q}_k\|^2}.$$

Similarly, we compute $\nabla_{\mathcal{M}} \phi_{k,2}^h|_{\sigma^h}$ and $\nabla_{\mathcal{M}} \phi_{k,3}^h|_{\sigma^h}$.

The discrete equation (43) can be rewritten to the following linear system: Find $u^h \in \mathbb{R}^{N_k^h}$ such that

$$\frac{1}{\lambda} A_k^h u^h + M_k^h u^h = M_k^h U_0 \tag{44}$$

holds, where $U_0 \in \mathbb{R}^{N_k^h}$ is given by

$$(U_0)_j = \frac{\sum_{\sigma^h \in \mathcal{T}_{k,j}^h} |\sigma^h| \, (u_0)_{|\sigma^h}}{\sum_{\sigma^h \in \mathcal{T}_{k,j}^h} |\sigma^h|}, \quad j = 1, \dots, N_k^h, \tag{45}$$

with $\mathcal{T}_{k,j}^h = \left\{ \sigma^h \in \mathcal{T}_k^h : \mathbf{p}_{k,j} \in \overline{\sigma^h} \right\}$. Using this definition of U_0, we obtain $\langle u_0, \phi_{k,j}^h \rangle^h = (M_k^h U_0)_j$ for $j = 1, \dots, N_k^h$.

Note that the Neumann boundary conditions are automatically incorporated in the scheme (44). The finite element approach is based on a weak formulation of (23) which contains the Neumann boundary conditions as natural conditions.

We obtain an element-wise constant image approximation U^h by setting

$$U^h_{|\sigma^h} = \frac{1}{3} \sum_{j=1}^{3} u^h(\mathbf{p}_{\sigma^h, j}) \tag{46}$$

for simplices $\sigma^h \in \mathcal{T}_k^h$.

By solving the diffusion equation on each region independently, the boundaries of the regions are not smoothed out. Vector-valued images are denoised by applying the method on each component.

3.7 Summary of the Image Processing Algorithm

We propose the following algorithm for image segmentation with postprocessing image restoration: Given a set of polygonal curves $\Gamma^0 = (\Gamma_1^0, \dots, \Gamma_{N_C}^0)$ and $\mathbf{X}^0 = (\mathbf{X}_1^0, \dots, \mathbf{X}_{N_C}^0)$ with $\mathbf{X}_i^0(I_i) = \Gamma_i^0$, $\mathbf{X}_i^0(q_j^i) \in \mathcal{M}$, $i = 1, \dots, N_C$, $j = j_0^i, \dots, N_i$, perform the following steps for $m = 0, 1, \dots, M - 1$:

(1) Compute the regions $\Omega_k^m \subset \mathcal{M}$ and the coefficients c_k^m, $k = 1, \dots, N_R$, as described in Sect. 3.5.
(2) Compute b^m as defined in (34) using the coefficients c_k^m of step 1. Compute $\mathbf{X}^{m+1} = \mathbf{X}^m + \delta \mathbf{X}^{m+1}$ by solving the linear equation (36b), see Sect. 3.2.

(3) Check if topology changes occur, see Sect. 3.4. In case of a topology change, except for a pure deletion of a curve, repeat the steps 1 and 2 n_{sub}-times with a step size of τ_m/n_{sub} and execute the topology change when it occurs in a substep.
(4) If necessary, perform global coarsening or refinement as described in Sect. 3.5.

Having found a final segmentation of the image at time $m = M$, perform a restoration by computing a piecewise smooth approximation of the image function as presented in Sect. 3.6.

4 Results and Discussion

4.1 Artificial Test Images

Images on the Stanford bunny We test the developed algorithm for image segmentation by considering artificial images on the Standford bunny (Stanford Computer Graphics Laboratory [31]). In a first experiment, we consider a gray-scaled image showing three dark disks. This example is similar to an experiment presented by Krüger et al. [18], who use a level set method to solve a geodesic active contour model with a balloon force for images on surfaces, whereas we use a parametric method for a Chan–Vese-like model for images on surfaces.

Figure 3 shows our image segmentation result using the developed direct, parametric approach for image segmentation. We use the parameters $\sigma = 2$ and $\lambda = 50$ to weight the curvature and external term. Let $\Delta t = \tau_m$ denote the time step size. The time step size is set to $\Delta t = 0.01$. This example demonstrates topology changes; in detail, it shows how one initial closed curve is split up into three single curves. The contours at four different time steps and the corresponding piecewise constant approximation are presented in Fig. 3. Of course, a level set technique as used in [18] can handle splitting automatically, whereas we need to detect the change in topology explicitly using the method described in Sect. 3.4. However, our method to detect topology changes is efficient, since it has a computational effort of $\mathcal{O}(N)$, where N is the number of node points of the polygonal curves.

In a second experiment, we demonstrate the creation and handling of triple junctions for a curve network on the Stanford bunny, cf. Fig. 4. For this experiment, we set $\sigma = 1$, $\lambda = 40$, and $\Delta t = 0.01$. The possibility of triple junctions is not considered in [18].

Further, we apply the image denoising method described in Sect. 2.5 and 3.6, cf. Fig. 5. Since the term $\Delta_{\mathcal{M}} u_k$ in (23) is weighted with $1/\lambda$, the denoised image is close to the original image, and the larger λ is chosen, cf. Fig. 5. Setting $\lambda = 1000$ or $\lambda = 10,000$, the noise is not completely smoothed out.
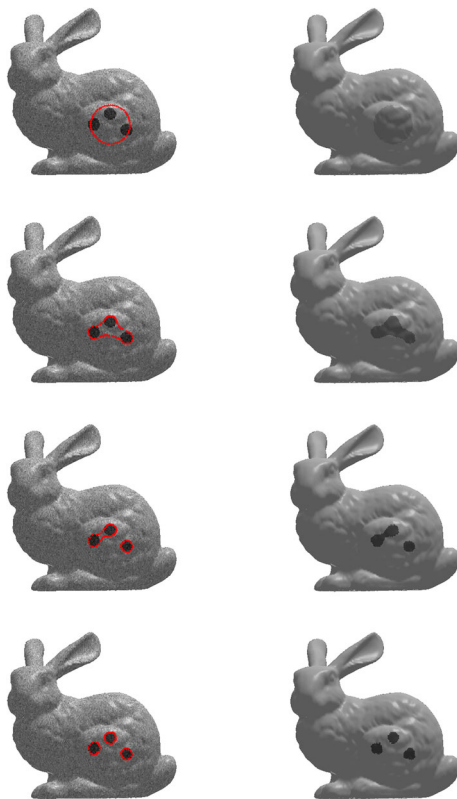
Fig. 3 Image segmentation of a *gray-scaled* image on the Stanford bunny. Demonstration of topology changes (splitting). *Left* original image and contours for $m = 1, 100, 140, 150$. *Right* piecewise constant approximation. The surface is from the Stanford Computer Graphics Laboratory, cf. [31]
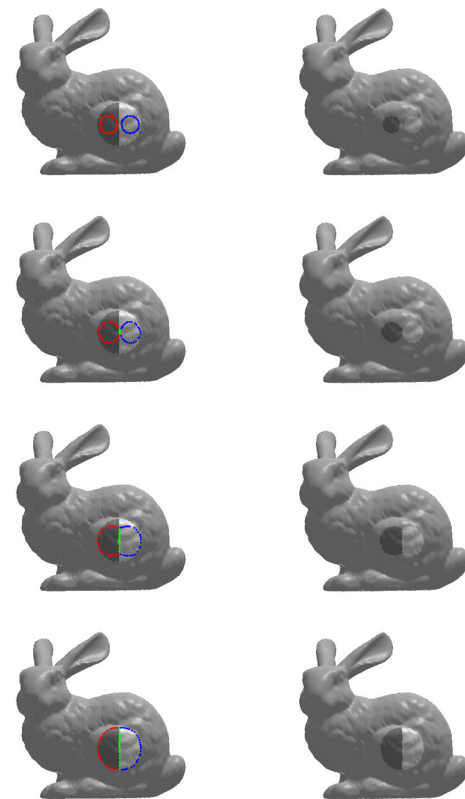

Fig. 4 Image segmentation of a *gray-scaled* image on the Stanford bunny. Curve network with triple junctions. *Left* original image and contours for $m = 1, 42, 100, 250$. *Right* piecewise constant approximation. The surface is from the Stanford Computer Graphics Laboratory, cf. [31]
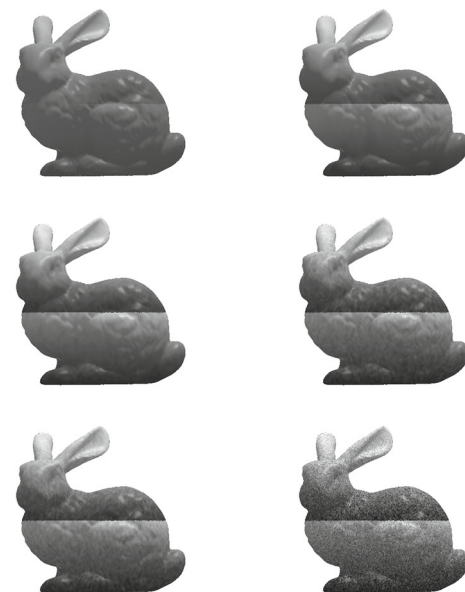
Setting $\lambda = 0.1$, the resulting denoised version is close to a piecewise constant image approximation. Setting $\lambda = 100$ results in a good denoised image.

Image on a torus We consider an artificial image on a torus to demonstrate that the method can be applied on surfaces of arbitrary topology type (for example arbitrary genus). Figure 6 shows a torus with a color image from different viewing angles and the curves at different iteration steps during the evolution. The RGB color space is used for the segmentation. As weighting parameter for the curvature $\sigma = 1$ is used, all three components of the color are weighted equally with $\lambda_1 = \lambda_2 = \lambda_3 = 20$. The time step size in this experiment is set to $\Delta t = 0.0001$.

Two different topology changes occur in this example: Around $m = 335$, two triple junctions and a new curve are created. Shortly after $m = 422$, one blue curve splits into two single curves.

4.2 Real Images

Lip contour segmentation We consider an application where lip contours should be detected on given face image data.

Fig. 5 Image denoising with edge enhancement. *Sub-figure 1–5* (row-wise) denoising result using $\lambda = 0.1, 1, 100, 1000, 10,000$. *Sub-figure 6* original image with noise. The surface is from the Stanford Computer Graphics Laboratory, cf. [31]
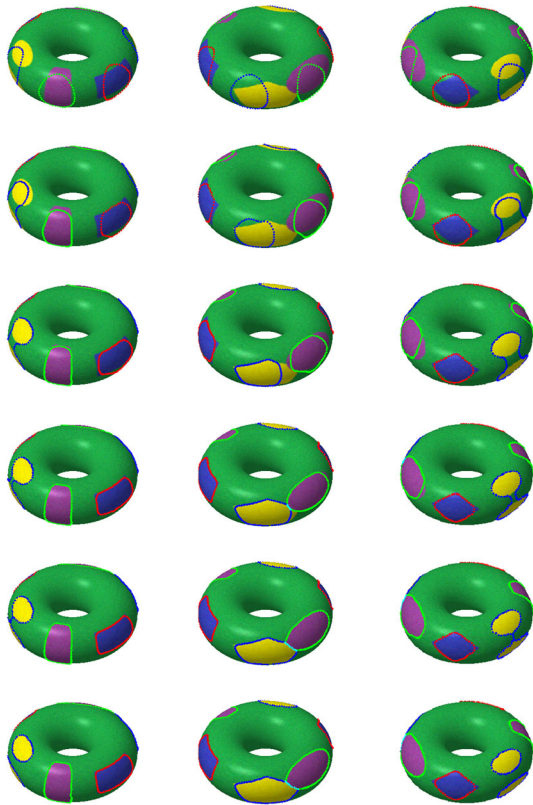
Fig. 6 Segmentation of an artificial image showing different objects on a torus. *First–sixth rows* original image and contours for $m = 1, 100, 200, 335, 422, 600$. *First–third columns* different viewing angles
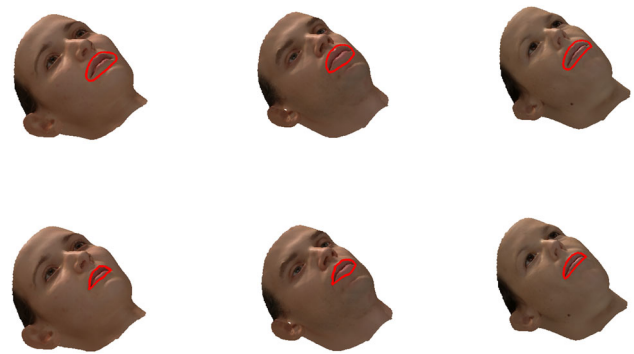

Fig. 7 Lip contour detection with a two-phase segmentation. Initial (*first row*) and final (*second row*) contours. The surfaces and images are from the 3D basel face model (BFM) of the Computer Science department of the University of Basel, cf. [27]
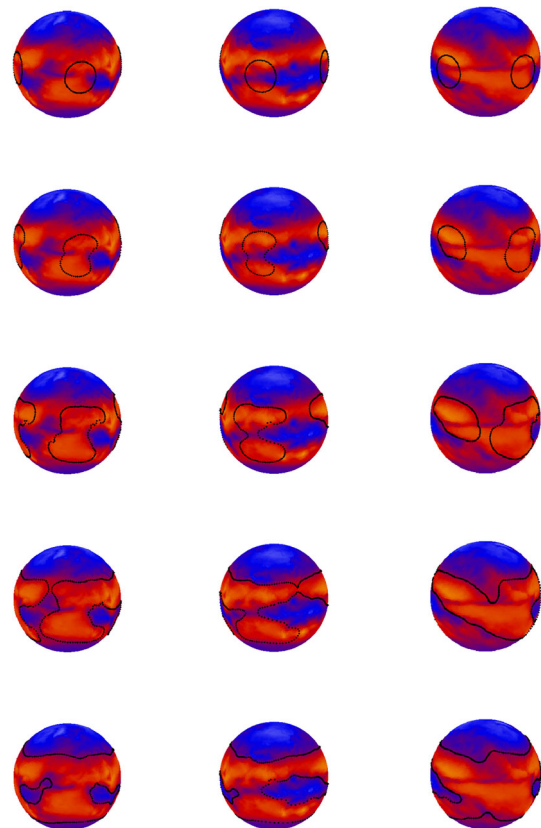
Figure 7 shows the results where the image segmentation algorithm is applied on three sample images of the 3D face scans from the 3D Basel Face Model (BFM) published by the Computer Science department of the University of Basel[2], see also [27]. The images are segmented using the chromaticity–brightness color space with $\sigma = 0.25$, $\lambda_C = 200$, $\lambda_B = 20$, and $\Delta t = 0.001$. The initial contour is a closed curve placed around the lips. Applying our algorithm for image segmentation, we obtain the final lip contours.

Processing of global Earth observation data Another application is the processing of Earth observation data. Global Earth observation data can be interpreted as an image given on a sphere. We apply the image segmentation and denoising method on data from the NASA Earth Observation data set[3], cf. [25]. We segment an image showing outgoing longwave radiation[4], see Figs. 8, 9 and 10. The colors represent the

Fig. 8 Segmentation of longwave radiation data given on the Earth's surface. *First–fifth rows* original image and contours for $m = 1, 20, 50, 75, 150$. *First–third columns* different viewing angles. The original image is from the NASA Earth Observation data set, [25]

amount of outgoing longwave radiation leaving the Earth's atmosphere in one month (here: January 2014). Yellow and orange colors represent greater heat emission (around 300–350 $\mathrm{Wm^{-2}}$); purple and blue colors represent intermediate emissions (around 200 $\mathrm{Wm^{-2}}$).

---

[2] http://faces.cs.unibas.ch/bfm/main.php?nav=1-0&id=basel_face_model.

[3] http://neo.sci.gsfc.nasa.gov/

[4] Imagery by Jesse Allen, NASA Earth Observatory, based on FLASH-Flux data. FLASHFlux data are produced using CERES observations

---

convolved with MODIS measurements from both the Terra and Aqua satellite. Data provided by the FLASHFlux team, NASA Langley Research Center.
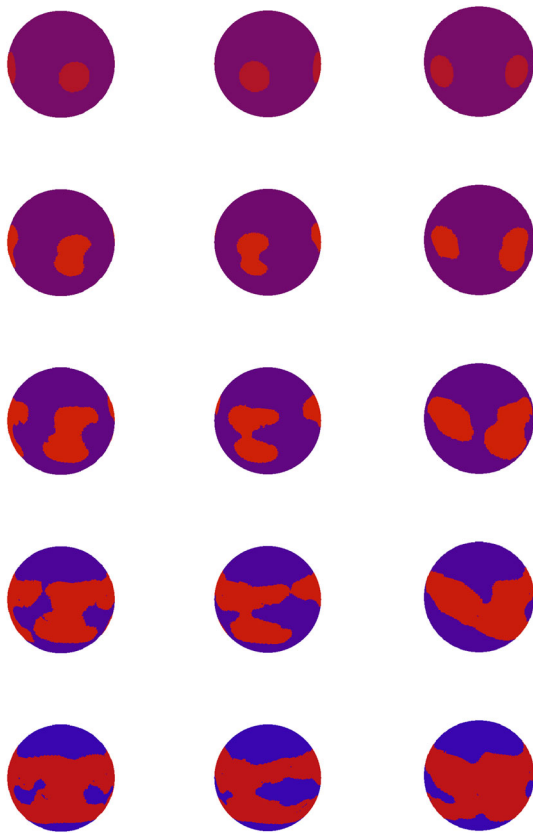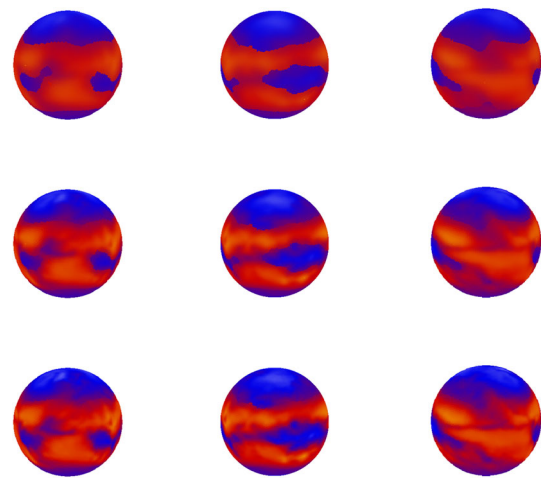
Fig. 10 Denoising of the longwave radiation data using the detected regions from time step $m = M = 150$. *First row* $\lambda = 100$. *Second row* $\lambda = 1000$. *Third row* $\lambda = 10{,}000$. *First–third columns* different viewing angles. The original image is from the NASA Earth Observation data set, [25]
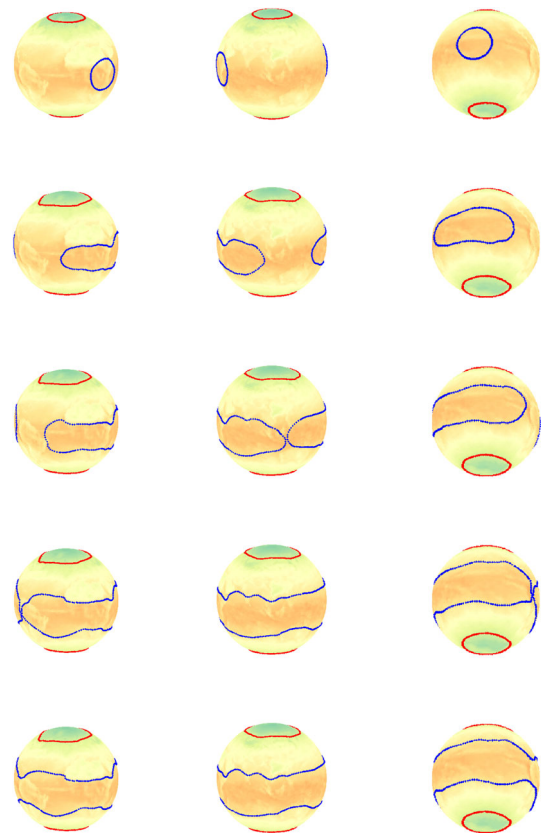

Fig. 9 Segmentation of longwave radiation data given on the Earth's surface. *First–fifth rows* piecewise constant approximation for $m = 1, 20, 50, 75, 150$. *First–third columns* different viewing angles. The original image is from the NASA Earth Observation data set, [25]

Figure 8 presents the given image data and the contours at different time steps (rows), observed from different viewing angles (columns). For the segmentation, we used the RGB color space and set the weighting parameters for the curvature and the external term to $\sigma = 1$ and $\lambda_1 = \lambda_2 = \lambda_3 = \lambda = 50$. As time step size we set $\Delta t = 0.0005$. Several topology changes (splitting and merging) occur during the segmentation (cf. e.g., $m = 75$). Figure 9 shows the corresponding piecewise constant approximations. Figure 10 presents the result of the postprocessing image restoration using the parameters $\lambda = 100$, $\lambda = 1000$, and $\lambda = 10{,}000$.

For demonstration of multiphase image segmentation, we consider a second example image from the NASA Earth Observation data set, cf. [25]. We now segment an image showing the Earth's net radiation. The net radiation is defined as the difference between the amount of solar energy which enters the Earth system and the amount of heat energy which escapes into space during one month (here: March 2014). Red color represents a net radiation around $280 \, \mathrm{Wm}^{-2}$, yellow color a net radiation around $0 \, \mathrm{Wm}^{-2}$, and blue-green color a net radiation of $-280 \, \mathrm{Wm}^{-2}$.

Figure 11 presents the original image with the contours at different time steps (rows), observed from different viewing

Fig. 11 Segmentation of net radiation data given on the Earth's surface. *First–fifth rows* original image and contours for $m = 1, 50, 71, 149, 180$. *First–third columns* different viewing angles. The original image is from the NASA Earth Observation data set, [25]

angles (columns). For the segmentation, we used $\sigma = 1$ and $\lambda_1 = \lambda_2 = \lambda_3 = \lambda = 300$ and the RGB color space. As time step size we set $\Delta t = 0.002$. The detected regions are not
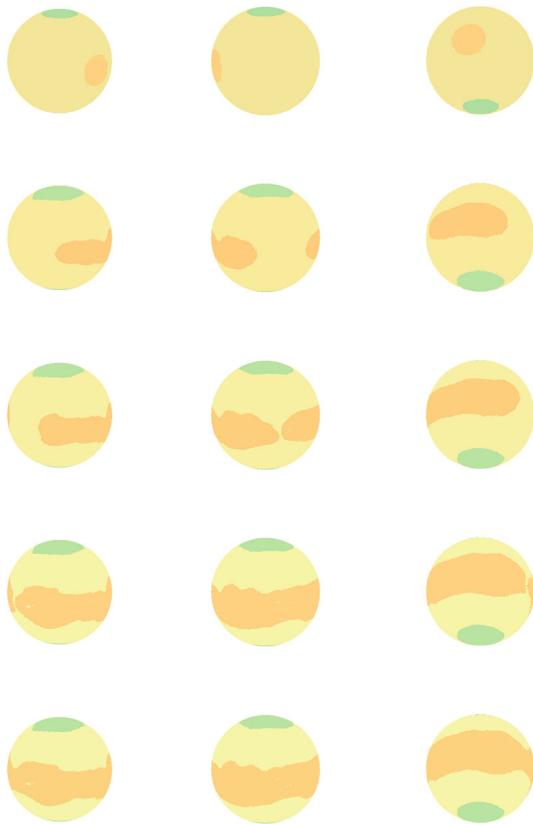
Fig. 12 Segmentation of net radiation data given on the Earth's surface. *First–fifth rows* piecewise constant approximation for $m = 1, 50, 71, 149, 180$. *First–third columns* different viewing angles. The original image is from the NASA Earth Observation data set, [25]

Fig. 13 Denoising of the net radiation data (image with added noise). *First row* noise-added image to be smoothed. *Second row* smoothing result using $\lambda = 100$. *Third row* $\lambda = 1000$. *Forth row* $\lambda = 10,000$. *First–third columns* different viewing angles. The original image is from the NASA Earth Observation data set, [25]
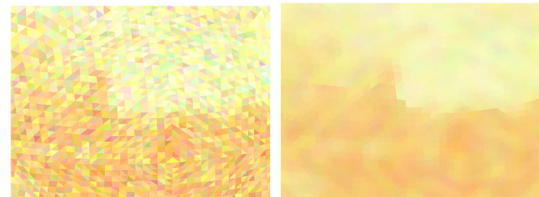
Fig. 14 Magnification of a part of the image. *Left* noisy, original image. *Right* smoothed version with $\lambda = 10,000$

separated by sharp image edges. Here, the detected boundaries are weak edges, i.e., they lie at locations in the image where the color smoothly changes from yellow to orange or yellow to green, respectively. At $m = 71$ a merging and at $m = 149$ a splitting occurs. Figure 12 shows the corresponding piecewise constant approximation.

Since the original image has little noise, we add some artificial, Gaussian noise to the image and repeat the segmentation using the generated image. As postprocessing step, we smooth the noisy image applying our restoration scheme, cf. Sect. 3.6. Figure 13 shows the image with added noise and the denoised image for different parameters λ under different viewing angles. We compare the results for several choices of λ: Using $\lambda = 100$, the denoised image is very close to the piecewise constant image. For $\lambda = 1000$ and $\lambda = 10,000$, we obtain images with removed noise, but which still contain sufficient details of the original data set. The data are not smoothed out too strong compared to $\lambda = 100$. Figure 14 shows a magnification of a part of the surface. It shows the noisy image (left) and the result of the denoising using $\lambda = 10,000$ (right). We observe that the noise is well smoothed out.

We have shown how the developed method can be applied on segmentation of global Earth data. The given data need not be a classical image generated by a camera. The data can be any data defined on the Earth's surface, like radiation data as in the examples presented here.

One may argue that global Earth data can also be processed on a flat 2D domain, with a rectangular grid given by a discrete set of longitudes and latitudes. In principle, one can apply a two-dimensional image segmentation and restoration method as developed in [5] for 2D images. However, performing the image processing using the 2D image has some disadvantages: To map the global Earth data to a rectangular 2D image, image boundaries at the poles and at longitude 0 have to be created. Points which are close to each other on the sphere (like at the two opposite sides of longitude 0) are not close to each other in the 2D image. Also topology changes like boundary intersection will occur in the 2D image. The

coordinates of the boundary nodes on the left and the right boundary of the image may not fit, i.e., they lie on longitude 0, but can have different latitude values resulting in two different 3D points. Further, the length of curves and the area of regions near the poles are differently scaled compared to those near the equator. Polar regions always appear larger in 2D images. After a segmentation is performed, the size of the segmented regions are often of interest. Therefore, the area of the regions can be easily computed in a postprocessing step using the sphere data. This is not directly possible from the 2D image due to the different scaling of the polar and equatorial regions. In summary, it is beneficial to consider global Earth data directly as images on a surface.

5 Conclusion

We presented how images on a surface can be efficiently segmented by curve evolution with a parametric approach. Furthermore, we showed how restoration with edge enhancement can be performed as a postprocessing step. We considered extensions of the Mumford–Shah [24] and Chan–Vese [10] models to images on surfaces. The velocity of the parameterized curves was restricted to the tangent space of the surface, which guarantees that all curves modeled as parametric curves in $\mathbb{R}^3$ stay on the surface during the evolution. We introduced an efficient numerical scheme based on a method for geodesic curvature flow [4]. Topology changes can be detected fast with an effort of $\mathcal{O}(N)$, where N is the number of node points of the discretized curves. The applicability of the developed schemes on different images and different surfaces has been demonstrated in several experiments.

References

1. Balažovjech, M., Mikula, K., Petrášová, M., Urbán, J.: Lagrangean method with topological changes for numerical modelling of forest fire propagation. In: Proceedings of ALGORITMY 2012, 19th Conference on Scientific Computing, pp. 42–52. Vysoké Tatry, Podbansk'v, Slovakia (2012)
2. Barrett, J.W., Garcke, H., Nürnberg, R.: A parametric finite element method for fourth order geometric evolution equations. J. Comput. Phys. **222**(1), 441–467 (2007)
3. Barrett, J.W., Garcke, H., Nürnberg, R.: On the variational approximation of combined second and fourth order geometric evolution equations. SIAM J. Sci. Comput. **29**(3), 1006–1041 (2007)
4. Barrett, J.W., Garcke, H., Nürnberg, R.: Numerical approximation of gradient flows for closed curves in $\mathbb{R}^d$. IMA J. Numer. Anal. **30**(1), 4–60 (2010)
5. Benninghoff, H., Garcke, H.: Efficient image segmentation and restoration using parametric curve evolution with junctions and topology changes. SIAM J. Imaging Sci. **7**(3), 1451–1483 (2014)
6. Bertalmio, M., Cheng, L.T., Osher, S., Sapiro, G.: Variational problems and partial differential equations on implicit surfaces: the framework and examples in image processing and pattern formation. J. Comput. Phys. **174**(2), 759–780 (2001)
7. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int. J. Comput. Vision **22**(1), 61–79 (1997)
8. Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. SIAM J. Appl. Math. **66**, 1632–1648 (2006)
9. Chan, T.F., Sandberg, B.Y., Vese, L.A.: Active contours without edges for vector-valued images. J. Vis. Commun. Image R. **11**(2), 130–141 (2000)
10. Chan, T.F., Vese, L.A.: Active contours without edges. IEEE Trans. Image Process. **10**(2), 266–277 (2001)
11. Cheng, L.T., Burchard, P., Merriman, B., Osher, S.: Motion of curves constrained on surfaces using a level-set approach. J. Comput. Phys. **175**(2), 604–644 (2002)
12. Čunderlík, R., Mikula, K., Tunega, M.: Nonlinear diffusion filtering of data on the Earth's surface. J. Geod. **87**(2), 143–160 (2013)
13. Davis, T.: Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method. ACM Trans. Math. Softw. **30**(2), 196–199 (2004)
14. Dziuk, G., Elliott, C.M.: Finite element methods for surface PDEs. Acta Numer. **22**, 289–396 (2013)
15. Garcke, H., Wieland, S.: Surfactant spreading on thin viscous films: nonnegative solutions of a coupled degenerate system. SIAM J. Math. Anal. **37**(6), 2025–2048 (2006)
16. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. Int. J. Comput. Vision **1**(4), 321–331 (1988)
17. Kimmel, R.: Intrinsic scale space for images on surfaces: the geodesic curvature flow. Graph. Model. Im. Proc. **59**(5), 365–372 (1997)
18. Krüger, M., Delmas, P., Gimelfarb, G.: Active contour based segmentation of 3D surfaces. In: Proceedings of the European Conference on Computer Vision, pp. 350–363. Marseille, France (2008)
19. Lai, R., Chan, T.F.: A framework for intrinsic image processing on surfaces. Comput. Vis. Image Und. **115**(12), 1647–1661 (2011)
20. Lang, S.: Introduction to differentiable manifolds, 2nd edn. Universitext. Springer, Berlin (2002)
21. Lee, J.M.: Introduction to smooth manifolds, graduate texts in mathematics, vol. 218. Springer, Heidelberg (2002)
22. Mikula, K., Urbán, J.: New fast and stable Lagrangean method for image segmentation. In: Proceedings of the 5th International Congress on Image and Signal Processing (CISP 2012), pp. 834–842. Chongquing (2012)
23. Mikula, K., Ševčovič, D.: Evolution of curves on a surface driven by the geodesic curvature and external force. Appl. Anal. **85**(4), 345–362 (2006)
24. Mumford, D., Shah, J.: Optimal approximation by piecewise smooth functions and associated variational problems. Commun. Pure. Appl. Math. **42**, 577–685 (1989)
25. NASA: NASA Earth Observations (2014). http://neo.sci.gsfc.nasa.gov/
26. Osher, S., Sethian, J.A.: Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. J. Comput. Phys. **79**(1), 12–49 (1988)
27. Paysan, P., Knothe, R., Amberg, B., Romdhani, S., Vetter, T.: A 3D face model for pose and illumination invariant face recognition. In: Proceedings of the 6th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) for Security. Safety and Monitoring in Smart Environments, pp. 296–301. Genova (2009)
28. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D **60**(1–4), 259–268 (1992)
29. Spira, A., Kimmel, R.: Geometric curve flows on parametric manifolds. J. Comput. Phys. **223**, 235–249 (2007)
30. Tian, L., Macdonald, C.B., Ruuth, S.J.: Segmentation on surfaces with the closest point method. In: Proceedings of the 16th IEEE

International Conference on Image Processing, pp. 3009–3012. Cairo (2009)

31. Turk, G., Levoy, M.: Zippered polygon meshes from range images. In: Proceedings of the 21st annual conference on Computer graphics and interactive techniques (SIGGRAPH '94), pp. 311–318. ACM, New York (1994)

32. Wu, C., Tai, X.C.: Augmented Lagrangian method dual methods, and split bregman iteration for ROF, vectorial TV, and high order models. SIAM J. Imaging Sci. **3**(3), 300–339 (2010)

33. Wu, C., Zhang, J., Duan, Y., Tai, X.C.: Augmented Lagrangian method for total variation based image restoration and segmentation over triangulated surfaces. J. Sci. Comput. **50**(1), 145–166 (2012)

34. Zhou, L., Li, J.: Image segmentation on implicit surface based on Chan-Vese model. J. Theor. Appl. Inf. Technol. **48**(1), 206–209 (2013)

Harald Garcke received the Ph.D. degree in applied mathematics from Bonn University, Germany, in 1993. He subsequently went on postdoctoral positions at Sussex University (UK) and Bonn where he finished his habilitation thesis in 2000. He became a full professor at the University of Regensburg in 2002. His research interests include nonlinear partial differential equations, geometric evolution equations, numerical analysis, optimization, and image analysis.

Heike Benninghoff is a research associate at German Aerospace Center (DLR) in Oberpfaffenhofen, Germany, working since 2010 in the field of autonomous optical navigation and control. She received her Ph.D. in applied mathematics in 2015 from the University of Regensburg, Germany. Her research interests include visual navigation, image processing, and numerical simulation.