



Fast image segmentation and restoration
using parametric curve evolution with
junctions and topology changes

Heike Benninghoff and Harald Garcke

Preprint Nr. 15/2013

Fast image segmentation and restoration using parametric curve evolution with junctions and topology changes

Heike Benninghoff* and Harald Garcke†

Abstract

Curve evolution schemes for image segmentation based on a region based contour model allowing for junctions, vector-valued images and topology changes are introduced. Together with an a posteriori denoising in the segmented homogeneous regions this leads to a fast and efficient method for image segmentation and restoration. An uneven spread of mesh points is avoided by using the tangential degrees of freedom. Several numerical simulations on artificial test problems and on real images illustrate the performance of the method.

Key words. Image segmentation, restoration, active contours, region based, Mumford-Shah, Chan-Vese, parametric method, variational methods, topology changes, equidistribution, triple junctions.

AMS subject classification. 94A08, 68U10, 65K10, 35K55, 49Q10

1 Introduction

Image segmentation and image smoothing are two fundamental tasks in image processing. Image segmentation is the problem of partitioning an image into constituent parts and to identify edges in a given image. As image smoothing of the whole image typically leads to a blurring of edges a possible strategy in image processing is to first find homogeneous regions and in a second step to perform a smoothing only in these homogeneous regions.

Many different approaches for image segmentation have been proposed. Active contours (or ‘snakes’) have been first proposed by Kass et al. (1988) and have been later used by Caselles et al. (1997) and Kichenassamy et al. (1996). The basic idea is to evolve a curve in order to find local minima of a suitable energy functional, and thus localize the object. The energy is chosen such that small values are attained at edges. The original snake models by Kass et al. (1988) used an energy which was not independent of the parameterization chosen for a given curve. In addition it was argued that the inability to change topology was a significant disadvantage of this original snake model.

Caselles et al. (1997) and Malladi et al. (1995) then used a geometric functional instead in which the energy no longer depends on the curve parameterization. In addition the level set framework of Osher and Sethian (1988) made it possible to split and join curves and hence also topological changes were possible.

The above mentioned active contour models typically depend on the gradient of the given image which is used to stop the evolution of the curve and hence these models are restricted to objects

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

†Fakultät für Mathematik, Universität Regensburg, 93040 Regensburg, Germany, harald.garcke@ur.de

for which edges are defined by a gradient. However, there are regions whose boundaries are badly defined through the gradient as e.g. smeared boundaries or larger objects which are given by grouping smaller ones, see e.g. Aubert and Kornprobst (2006). In addition, gradient based methods heavily depend on noise and in particular in the presence of noise the evolution often get trapped as several local maxima of the image gradient exist. Later new region based active contour models have been developed which can detect objects which boundaries are not necessarily defined by the gradient of the image. These active contour models are called active contours ‘without edges’ and the main new ingredient is that the information inside the detected regions is used and not only information at their boundaries. We refer to Chan and Vese (2001), Ronfard (1994) and Tsai et al. (2001) for such approaches. The approach of Chan and Vese (2001) uses a two-phase, piecewise constant approximation of the functional of Mumford and Shah (1989) to partition a given image into two phases, i.e. object and background. In their approach a curve evolution in the context of a level set method is used to solve the partition problem. Topological derivatives and sensitivities have also been used to solve the Chan-Vese segmentation problem, see e.g. He and Osher (2007) and Hintermüller and Laurain (2009). In principle, also non-constant functions in the regions can be used and we refer to Tsai et al. (2001) who use the full Mumford-Shah functional to define a region-based descent direction for the curve evolution which is also solved in the level set context.

In principle the curve evolution problem in region based active contour models can also be solved with the evolution of parametric curves, see e.g. Doğan et al. (2008). As mentioned above the evolution of curves does not allow for an automatic change of topology. However, different authors proposed methods to handle topology changes, see Araki et al. (1997), Doğan et al. (2008) and the recent paper by Nakhmani and Tannenbaum (2012).

The direct parametric approach often has a typical undesirable behavior during the evolution, namely during the geometric flow some points bunch together whereas other points drift apart. Due to this uneven spread of points quantities like curvature and normal tangent are often not computed to a high precision and also the segmentation is not very accurate. This problem has been addressed in the context of the Mumford-Shah formulation by Cremers et al. (2001) who modified the usual length constrained on the contour by taking a so-called diffusion snake. As a result a hybrid model is obtained, which combines the external energy of the Mumford-Shah functional with the internal energy of the snake of Kass et al. (1988). However, this approach has the disadvantage that it is not parameter free and also topology changes have not been considered in Cremers et al. (2001). We also refer to Srikrishnan et al. (2007) who use the equidistribution strategy for the curve mesh points of Mikula and Ševčovič (2006) to stabilize the curve evolution by introducing a highly nonlinear tangential evolution term for the mesh points. This method has been successfully used for edge driven active contour models and for tracking problems, see Srikrishnan et al. (2007). Another important problem in image analysis is the segmentation of vector-valued images. To the knowledge of the authors this problem has not been tackled within the parametric approach.

In this paper we introduce a novel parametric method for segmentation and denoising of images which is based on a recent approach of Barrett et al. (2007a,b) which was developed for the numerical treatment of curvature driven geometric evolutions of curves and leads to asymptotically equidistributed mesh points. The nearly equidistributed spacing of mesh points is essential for a stable contour evolution and for a robust and efficient detection of topology changes.

Let us state the *main features of the proposed approach*.

- *Image segmentation and denoising* of images can be dealt with by first using a Chan-Vese type approach, see Chan and Vese (2001); Chan et al. (2000); Vese and Chan (2002), to segment the image and a posteriori an image smoothing by minimizing the Mumford-Shah functional for a fixed contour set is applied.

- The mesh properties of the discrete parametric curves stay good due to the tangential redistribution method of Barrett et al. (2007a,b) which keeps mesh points nearly *equidistributed*.
- The possibility of *junctions* is included into the method.
- The approach can deal with *vector valued images*, i.e. e.g. colored images can be considered.
- *Topological changes*, including topological changes involving junctions, are computed *efficiently* with an effort of $\mathcal{O}(N)$ where N is the number of mesh points in the curve network. This is possible due to the equidistribution property together with a generalization of the approach of Mikula and Urbán (2012).
- The approach has the advantage that for segmentation only one-dimensional geometric PDEs have to be solved which discretized versions can be solved very fast with a direct numerical algebra solver.

2 A model for image segmentation and denoising

Let $\Omega \subset \mathbb{R}^2$ be a rectangular image domain and $u_0 : \Omega \rightarrow [0, 1]^d$ a given image function. In case of a scalar, gray-scaled image $d = 1$, u_0 describes the gray value, where 0 corresponds to black and 1 corresponds to white. In case of a color image $d = 3$, the components of u_0 describe for example the intensities of the red, green and blue part of the color. We first consider the scalar case $d = 1$ and later state how color images can be handled.

The objective of image segmentation is to identify objects and edges shown in the image. The model of Mumford and Shah (1989) for optimal approximation of images aims at finding a set of curves $\Gamma = \Gamma_1 \cup \dots \cup \Gamma_{N_C}$ and a piecewise smooth function $u : \Omega \rightarrow \mathbb{R}$ approximating u_0 with possible discontinuities across Γ . The energy to be minimized is

$$E(\Gamma, u) = \sigma |\Gamma| + \int_{\Omega \setminus \Gamma} \|\nabla u\|^2 dx + \lambda \int_{\Omega} (u_0 - u)^2 dx, \quad (2.1)$$

where $\sigma, \lambda > 0$ are weighting parameters, $|\Gamma|$ denotes the total length of the curves in Γ and $\|\cdot\|$ is the Euclidean norm. The first term penalizes the length of the curves, the second term does not allow u to change too much in $\Omega \setminus \Gamma$ and the third term requests that u is a good approximation of u_0 . The general Mumford-Shah problem is difficult to solve without restrictions, for example on the class of approximating functions u . Similar to Chan and Vese (2001), we consider for the segmentation of the image a reduced problem. We assume that the curves in Γ partition the set Ω in connected components and let $\Omega_1, \dots, \Omega_{N_R}$ be the connected components of $\Omega \setminus \Gamma$. Further, let each curve Γ_i be the interface between two regions $\Omega_{k^+(i)}$ and $\Omega_{k^-(i)}$ with $k^+(i), k^-(i) \in \{1, \dots, N_R\}$. Thus, we have the following decomposition of the domain Ω :

$$\Omega = \Omega_1 \cup \dots \cup \Omega_{N_R} \cup \Gamma_1 \cup \dots \cup \Gamma_{N_C}. \quad (2.2)$$

Figure 1 gives an example of a decomposition of a rectangular domain. The regions and interfaces can have more than one connected component. The interfaces can be closed, can meet at a triple junction and boundary intersections with $\partial\Omega$ can occur.

In a first step, we restrict ourselves to piecewise constant functions u of the form

$$u = \sum_{k=1}^{N_R} c_k \chi_{\Omega_k}, \quad (2.3)$$

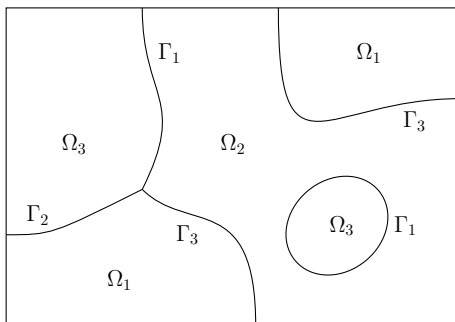


Figure 1: Example of a decomposition in regions Ω_k and interfaces Γ_i , $k = 1, \dots, N_R = 3$, $i = 1, \dots, N_C = 3$.

where χ_{Ω_k} denotes the characteristic function of Ω_k . As u is piecewise constant, we have $\nabla u = 0$ on $\Omega \setminus \Gamma$ and the functional (2.1) reduces to

$$E(\Gamma, c_1, \dots, c_{N_R}) = \sigma|\Gamma| + \lambda \sum_{k=1}^{N_R} \int_{\Omega_k} (u_0 - c_k)^2 dx. \quad (2.4)$$

The length term $\sigma|\Gamma|$ ensures that the curves $\Gamma_1, \dots, \Gamma_N$ have finite length. The second term is a bulk energy, designed such that Ω_k best approximates the several objects in the image. The interfaces Γ_i are attracted to the edges of objects in the image.

In so-called region based active contours methods like (2.1) and (2.4), only the raw image intensity function u_0 is involved in contrast to edge based active contours models which make use of the gradient of u_0 to detect edges of objects, cf. Kass et al. (1988); Malladi et al. (1995); Caselles et al. (1997). In particular, no prior smoothing of u_0 needs to be done when applying region based active contours models. Since the model does not depend on the gradient, noisy images and images with weak edges (Chan and Vese, 2001), i.e. smooth transitions between gray values with small $\|\nabla u\|$, can be handled.

2.1 Multi-phase image segmentation

Fixing Γ in (2.4) and considering a variation in c_k , $k \in \{1, \dots, N_R\}$, leads to

$$c_k = \frac{\int_{\Omega_k} u_0 dx}{\int_{\Omega_k} 1 dx}, \quad (2.5)$$

i.e. $c_k \in \mathbb{R}$ is set to the mean of u_0 in Ω_k .

For a variation of the edges Γ , we consider time-dependent curves and regions: Let $\Gamma_1(t), \dots, \Gamma_{N_C}(t) \subset \Omega$, $t \in [0, T]$, be smooth, evolving curves. Let an approximation $u(t)$ of u_0 be given by a piecewise constant function with $u(t)|_{\Omega_k(t)} = c_k(t) \in \mathbb{R}$, $k = 1, \dots, N_R$, where $c_k(t)$ is set to the mean of u_0 in $\Omega_k(t)$. The curve $\Gamma_i(t)$ should evolve in time such that the energy (2.4) decreases most quickly.

We now introduce a representation of the curves by smooth parameterizations. Let $\vec{x}_i : I_i \times [0, T] \rightarrow \mathbb{R}^2$ be a smooth function, such that $\vec{x}_i(\cdot, t)$ is a smooth parameterization of $\Gamma_i(t)$. The set I_i is a one-dimensional reference manifold, e.g. $I_i = [0, 1]$ for open curves, i.e. curves with $\partial\Gamma_i(t) \neq \emptyset$, and $I_i = S^1 \cong \mathbb{R}/\mathbb{Z}$ for closed curves, i.e. curves with $\partial\Gamma_i(t) = \emptyset$. Further, we define a normal vector field $\vec{\nu}_i(\cdot, t)$ on $\Gamma_i(t)$ by $\vec{\nu}_i : I_i \times [0, T] \rightarrow \mathbb{R}^2$ such that $\vec{\nu}_i(\rho, t)$ is a normal on $\Gamma_i(t)$ at $\vec{x}_i(\rho, t)$.

In detail, we set $\vec{v}_i(\rho, t) = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} (\vec{x}_i)_s(\rho, t)$, where s is the arc-length and $(\vec{x}_i)_s = (\vec{x}_i)_\rho / \|(\vec{x}_i)_\rho\|$ denotes the derivative of \vec{x}_i with respect to arc-length. We choose the parameterization such that the normal vector field defines an orientation of $\Gamma_i(t)$ with $\vec{v}_i(\cdot, t)$ pointing from phase $\Omega_{k^-(i)}(t)$ to $\Omega_{k^+(i)}(t)$.

Using methods from the calculus of variations one obtains the following evolution law for the curves $\Gamma_i(t)$, see e.g. Chan and Vese (2001); Deckelnick et al. (2005); Aubert and Kornprobst (2006),

$$(V_n)_i = \sigma \kappa_i + F_i, \quad (2.6)$$

where $(V_n)_i$ denotes the normal velocity and κ_i the curvature of the curve $\Gamma_i(t)$ and F_i is an external forcing term defined by

$$F_i(t) = \lambda[(c_{k^+(i)}(t) - u_0)^2 - (c_{k^-(i)}(t) - u_0)^2], \quad (2.7)$$

where $k^+(i)$ and $k^-(i)$ denote the indices of the two phases separated by $\Gamma_i(t)$.

The curvature term in (2.6) can be derived by a variation of the length functional weighted with a constant $\sigma > 0$. Using the parametric description the evolution equation (2.6) can be rewritten as

$$(\vec{x}_i)_t \cdot \vec{v}_i = \sigma \kappa_i + F_i \quad (2.8a)$$

and the curvature $\kappa_i : I_i \times [0, T] \rightarrow \mathbb{R}$ is related to \vec{x}_i by

$$\kappa_i \vec{v}_i = (\vec{x}_i)_{ss}, \quad i = 1, \dots, N_C. \quad (2.8b)$$

Without the external energy term, the evolution would reduce to the well-known curvature flow. The term F_i can be derived by considering a variation of Γ and using a transport theorem for the external energy in (2.4).

2.2 Triple junctions and boundary intersection

The methods presented above can be generalized to complex structures of curves which involve triple junctions and boundary intersection points. At these points, additional conditions need to be stated, cf. (Barrett et al., 2007a). Figure 2 shows an example of a curve network with triple and boundary intersection points.

Let Γ_i be a non-closed curve with a smooth parameterization $\vec{x}_i : I_i \rightarrow \mathbb{R}^2$ and $I_i = [0, 1]$. Let $\vec{\Lambda}_k \in \Omega$, $k = 1, \dots, N_T$, denote the triple junctions. For each $k \in \{1, \dots, N_T\}$ let $i_{k,1}, i_{k,2}, i_{k,3} \in \{1, \dots, N_C\}$ denote the indices of curves $\Gamma_{i_{k,l}}$, $l = 1, 2, 3$, $i_{k,1} \neq i_{k,2} \neq i_{k,3} \neq i_{k,1}$, such that

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

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

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

$$\text{the triple junction } \vec{\Lambda}_k \text{ does not pull apart,} \quad (2.9a)$$

$$\sum_{l=1}^3 (-1)^{\rho_{k,l}} \vec{\tau}_{i_{k,l}} = 0, \quad (2.9b)$$

where $\vec{\tau}_{i_{k,l}} := (\vec{x}_{i_{k,l}})_s$ is a tangent vector field at $\Gamma_{i_{k,l}}$, $l = 1, 2, 3$. The condition (2.9b) is equivalent to a 120° angle condition at triple junctions, see Barrett et al. (2007a).

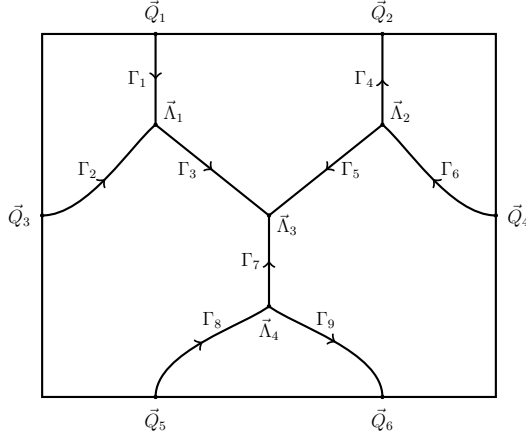


Figure 2: Example of a curve network with triple junctions and boundary intersection points

Let $\vec{Q}_k \in \partial\Omega$, $k = 1, \dots, N_I$, be the set of boundary intersection points. For $k = 1, \dots, N_I$ let $i_{I,k}$ denote the curve index and $\rho_{I,k} \in \{0, 1\}$ such that $\vec{x}_{i_{I,k}}(\rho_{I,k}) = \vec{Q}_k$. Let $\vec{\tau}_{i_{I,k}} := (\vec{x}_{i_{I,k}})_s$ be the corresponding tangent vector field at $\Gamma_{i_{I,k}}$. The following conditions need to hold at the boundary intersection points \vec{Q}_k , $k = 1, \dots, N_I$:

$$\text{the curve endpoint } \vec{Q}_k \text{ remains attached to } \partial\Omega, \quad (2.10a)$$

$$\vec{\tau}_{i_{I,k}}(\rho_{I,k}) \cdot \vec{n}_{\partial\Omega}(\vec{Q}_k)^\perp = 0, \quad (2.10b)$$

where $\vec{n}_{\partial\Omega}$ is a normal vector field at the boundary of the rectangular image domain Ω . The first equation is an attachment condition and the second equation enforces a 90 degree angle condition at the boundary intersection point.

2.3 Image smoothing with edge enhancement

The presented image segmentation method automatically provides a piecewise constant approximation of a possibly noisy image given by (2.3). For a variety of real images, a piecewise constant approximation poses a too large simplification. A piecewise smooth approximation can be found by reconsidering the Mumford-Shah functional (2.1), with approximations $u|_{\Omega_k} = u_k$, $k = 1, \dots, N_R$, where $u_k \in C^1(\Omega_k, \mathbb{R})$. Fixing curves $\Gamma_1, \dots, \Gamma_{N_C}$, the energy functional reduces to

$$E^{MS,2}(u_1, \dots, u_{N_R}) = \sum_{k=1}^{N_R} \left(\lambda_k \int_{\Omega_k} (u_0 - u_k)^2 dx + \int_{\Omega_k} \|\nabla u_k\|^2 dx \right). \quad (2.11)$$

In contrast to (2.1), we also allow region-dependent parameters $\lambda_k > 0$. By considering variations of the form $u_k + \epsilon\eta$, $\epsilon \in (-\epsilon_0, \epsilon_0)$, $\eta \in C^1(\overline{\Omega_k}, \mathbb{R})$ in (2.11), we obtain using the classical methods from the theory of calculus of variations the following boundary value problem:

Find $u : \Omega \rightarrow \mathbb{R}$ such that

$$-\frac{1}{\lambda_k} \Delta u + u = u_0 \quad \text{in } \Omega_k, \quad k = 1, \dots, N_R, \quad (2.12a)$$

$$\nabla u \cdot \vec{\nu}_i = 0 \quad \text{on } \Gamma_i, \quad i = 1, \dots, N_C, \quad (2.12b)$$

$$\nabla u \cdot \vec{n}_{\partial\Omega} = 0 \quad \text{on } \partial\Omega. \quad (2.12c)$$

We separately solve the partial differential equations with Neumann boundary conditions on the regions $\Omega_1, \dots, \Omega_{N_R}$. The solution u poses an approximation of the original image u_0 . The parameter

$\frac{1}{\lambda_k} > 0$ controls the smoothing effect of the Laplace operator. The bigger λ_k , the closer is the approximation to the original image. The smaller λ_k , the smoother is u . As we solve the equation separately in each region, the interfaces Γ_i are not smoothed out. Consequently the edges of objects in the image remain sharp if the interfaces match with the edges.

The system (2.12) is close to the scheme proposed by Tsai et al. (2001), where the authors derived a level set method for two phases from the Mumford-Shah functional by considering the approximation u as the optimal estimate of a stochastic process.

2.4 Color images

A color image is given as a vector-valued image function $\vec{u}_0 = (u_{0,1}, u_{0,2}, u_{0,3}) : \Omega \rightarrow [0, 1]^3$ where the components of \vec{u}_0 for example denote the red, green and blue color channel (RGB) of the image. Using the RGB image data, we consider the following energy, cf. Chan et al. (2000):

$$E(\Gamma, \vec{u}) = \sigma|\Gamma| + \sum_{j=1}^3 \lambda_j \int_{\Omega} (u_{0,j} - u_j)^2 dx, \quad (2.13)$$

where $\vec{u} = (u_1, u_2, u_3) : \Omega \rightarrow \mathbb{R}^3$ is piecewise constant, i.e. $\vec{u}|_{\Omega_k} =: \vec{c}_k = (c_{k,1}, c_{k,2}, c_{k,3})$, $k = 1, \dots, N_R$. Considering variations in the $c_{k,j}$ leads to

$$c_{k,j} = \frac{\int_{\Omega_k} u_{0,j} dx}{\int_{\Omega_k} 1 dx}, \quad j = 1, 2, 3, \quad k = 1, \dots, N_R. \quad (2.14)$$

Thus, each component of \vec{c}_k is set to the mean of $u_{0,j}$ in Ω_k . Considering a variation in Γ leads to the evolution equation

$$(V_n)_i = \sigma \kappa_i + F_i, \quad i = 1, \dots, N_C, \quad (2.15a)$$

$$F_i = \sum_{j=1}^3 \lambda_j [(u_{0,j} - c_{k+(i),j})^2 - (u_{0,j} - c_{k-(i),j})^2]. \quad (2.15b)$$

By modifying the weighting parameters λ_j , $j = 1, 2, 3$, we can control the segmentation with respect to the red, green and blue part of the image. However for some images, one may make use of other color spaces which allow to consider a color's chromaticity, brightness and saturation component separately. Two possible spaces are the CB space (chromaticity-brightness) and the HSV color space (hue, saturation and value). The color spaces CB and HSV can be used for a variety of image processing tasks like segmentation, denoising or enhancement of color images, see Chan et al. (2001), Tang et al. (2002), Aujol and Kang (2006).

For an RGB image function \vec{u}_0 , the chromaticity function $\vec{v}_0 : \Omega \rightarrow S^2$ and the brightness function $b_0 : \Omega \rightarrow \mathbb{R}$ are defined by

$$\vec{v}_0 = \frac{\vec{u}_0}{\|\vec{u}_0\|}, \quad b_0 = \|\vec{u}_0\|. \quad (2.16)$$

The energy to be minimized is

$$E(\Gamma, \vec{v}, b) = \sigma|\Gamma| + \lambda_C \int_{\Omega} \|\vec{v}_0 - \vec{v}\|^2 dx + \lambda_B \int_{\Omega} (b_0 - b)^2 dx, \quad (2.17)$$

where $\vec{v} : \Omega \rightarrow \mathbb{R}^3$, $b : \Omega \rightarrow \mathbb{R}$ are piecewise constant, i.e. $\vec{v}|_{\Omega_k} = \vec{v}_k \in \mathbb{R}^3$, $b|_{\Omega_k} = b_k \in \mathbb{R}$, $k = 1, \dots, N_R$. The parameters $\sigma, \lambda_C, \lambda_B > 0$ weight the length, chromaticity and brightness term. Further, the constraint $\|\vec{v}\| = 1$ needs to be satisfied. Considering variations in b_k leads to

$$b_k = \frac{\int_{\Omega_k} b_0 \, dx}{\int_{\Omega_k} 1 \, dx}, \quad (2.18)$$

i.e. $b_k \in \mathbb{R}$ is set to the mean of the brightness $b_0 = \|\vec{u}_0\|$ in Ω_k . We now consider a variation of \vec{v} . This leads to a constrained minimization problem where the constraint $\|v_k\| = 1$ has to be enforced. This leads to the following computation of \vec{v}_k :

$$\vec{V}_k = \int_{\Omega_k} \vec{v}_0 \, dx, \quad \vec{v}_k = \frac{\vec{V}_k}{\|\vec{V}_k\|}. \quad (2.19)$$

Finally, fixing \vec{v} and \vec{b} and considering a variation of Γ leads to the evolution equation

$$(V_n)_i = \sigma \kappa_i + F_i, \quad i = 1, \dots, N_C, \quad (2.20a)$$

$$F_i = \lambda_C [\|\vec{v}_0 - \vec{v}_{k+(i)}\|^2 - \|\vec{v}_0 - \vec{v}_{k-(i)}\|^2] + \lambda_B [(b_0 - b_{k+(i)})^2 - (b_0 - b_{k-(i)})^2]. \quad (2.20b)$$

In the HSV space, a color is given by three components: The periodical hue component describes the chromaticity ranging from red to yellow, green, cyan, blue, magenta and back to red. The saturation ranges from 0 to 1. A gray color (all three components of the corresponding RGB color are equal) has a saturation value of 0. A saturation of 1 is a maximum saturated color, i.e. it has one RGB value equal to zero and is therefore a mixture of only two RGB basic colors. The value component describes the luminosity of the color and is set to the maximum of the red, green or blue value.

Let $\vec{h}_0 : \Omega \rightarrow S^1 \subset \mathbb{R}^2$, $s_0 : \Omega \rightarrow [0, 1]$ and $v_0 : \Omega \rightarrow [0, 1]$ be the HSV components corresponding to an RGB image function \vec{u}_0 . The energy to be minimized is

$$E(\Gamma, \vec{h}, s, v) = \sigma |\Gamma| + \lambda_H \int_{\Omega} \|\vec{h}_0 - \vec{h}\|^2 \, dx + \lambda_S \int_{\Omega} (s_0 - s)^2 \, dx + \lambda_V \int_{\Omega} (v_0 - v)^2 \, dx, \quad (2.21)$$

where $\vec{h} : \Omega \rightarrow \mathbb{R}^2$, $s : \Omega \rightarrow [0, 1]$, $v : \Omega \rightarrow [0, 1]$ are piecewise constant, i.e. $\vec{h}|_{\Omega_k} =: \vec{h}_k \in \mathbb{R}^2$, $s|_{\Omega_k} =: s_k \in \mathbb{R}$, $v|_{\Omega_k} =: v_k \in \mathbb{R}$. The constraint $\|\vec{h}\| = 1$ needs to be satisfied. The parameters $\sigma, \lambda_H, \lambda_S, \lambda_V > 0$ weight the length, hue, saturation and value term. Similar as above, considering variations of the single components \vec{h}_k, s_k, v_k leads to

$$\vec{H}_k = \int_{\Omega_k} \vec{h}_0 \, dx, \quad \vec{h}_k = \frac{\vec{H}_k}{\|\vec{H}_k\|}, \quad s_k = \frac{\int_{\Omega_k} s_0 \, dx}{\int_{\Omega_k} 1 \, dx}, \quad v_k = \frac{\int_{\Omega_k} v_0 \, dx}{\int_{\Omega_k} 1 \, dx}, \quad (2.22)$$

where a constrained minimization problem has to be solved for \vec{h}_k .

Fixing \vec{h}, s and v and considering a variation of Γ leads to the evolution equation

$$(V_n)_i = \sigma \kappa_i + F_i, \quad i = 1, \dots, N_C, \quad (2.23a)$$

$$F_i = \lambda_H [\|\vec{h}_0 - \vec{h}_{k+(i)}\|^2 - \|\vec{h}_0 - \vec{h}_{k-(i)}\|^2] + \lambda_S [(s_0 - s_{k+(i)})^2 - (s_0 - s_{k-(i)})^2] + \lambda_V [(v_0 - v_{k+(i)})^2 - (v_0 - v_{k-(i)})^2]. \quad (2.23b)$$

To sum up, only the external forcing term F_i needs to be adapted for color images. The parametric scheme (2.8) can be used for both scalar and vector-valued images.

3 Numerical Approximation

3.1 Finite difference approximation

We introduce a finite difference approximation for the equations (2.8) with (2.9) and (2.10) in case of triple junctions and boundary intersections. The approach follows ideas of Barrett et al. (2007a,b) who consider curvature flows and formulate the discretization in a finite element context.

For $i = 1, \dots, N_C$, let $0 = q_0^i < q_1^i < \dots < q_{N_i}^i = 1$ be a decomposition of the interval I_i . 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. Let $0 = t_0 < t_1 < \dots < t_M = T$ be a partitioning of the time interval $[0, T]$ into possibly variable time steps $\tau_m := t_{m+1} - t_m$, $m = 0, \dots, M - 1$.

Let $\kappa^m = (\kappa_1^m, \dots, \kappa_{N_C}^m) \in C(I_1, \mathbb{R}) \times \dots \times C(I_{N_C}, \mathbb{R})$ be an approximation of $\kappa(\cdot, t_m) = (\kappa_1(\cdot, t_m), \dots, \kappa_{N_C}(\cdot, t_m))$ and $\vec{X}^m = (\vec{X}_1^m, \dots, \vec{X}_{N_C}^m) \in C(I_1, \mathbb{R}^2) \times \dots \times C(I_{N_C}, \mathbb{R}^2)$ an approximation of $\vec{x}(\cdot, t_m) = (\vec{x}_1(\cdot, t_m), \dots, \vec{x}_{N_C}(\cdot, t_m))$ such that κ_i^m and \vec{X}_i^m are piecewise linear on $[q_{j-1}^i, q_j^i]$, $i = 1, \dots, N_C$, $j = 1, \dots, N_i$. Further, we make use of the short hand notations

$$\kappa_{i,j}^m := \kappa_i^m(q_j^i), \quad \vec{X}_{i,j}^m := \vec{X}_i^m(q_j^i), \quad h_{i,j-\frac{1}{2}}^m := \|\vec{X}_{i,j}^m - \vec{X}_{i,j-1}^m\|.$$

Let $h^m := \max_{i=1, \dots, N_C, j=1, \dots, N_i} h_{i,j-\frac{1}{2}}^m$ be the maximal distance between two neighboring nodes of the polygonal curves. Let $\vec{\nu}^m := (\vec{\nu}_1^m, \dots, \vec{\nu}_{N_C}^m)$ such that $\vec{\nu}_i^m$, given by

$$\vec{\nu}_i^m|_{[q_{j-1}^i, q_j^i]} := \vec{\nu}_{i,j-\frac{1}{2}}^m := \frac{(\vec{X}_{i,j}^m - \vec{X}_{i,j-1}^m)^\perp}{h_{i,j-\frac{1}{2}}^m},$$

is a discrete normal vector field on Γ_i^m , $i = 1, \dots, N_C$. We define the following weighted approximating normal vector at $\vec{X}_{i,j}^m$ by

$$\vec{\omega}_{i,j}^m := \frac{h_{i,j-\frac{1}{2}}^m \vec{\nu}_{i,j-\frac{1}{2}}^m + h_{i,j+\frac{1}{2}}^m \vec{\nu}_{i,j+\frac{1}{2}}^m}{h_{i,j-\frac{1}{2}}^m + h_{i,j+\frac{1}{2}}^m} = \frac{(\vec{X}_{i,j+1}^m - \vec{X}_{i,j-1}^m)^\perp}{h_{i,j-\frac{1}{2}}^m + h_{i,j+\frac{1}{2}}^m}, \quad (3.1)$$

for $j = 1, \dots, N_i$ if $\partial\Gamma_i^m = \emptyset$ and for $j = 1, \dots, N_i - 1$ if $\partial\Gamma_i^m \neq \emptyset$. In the latter case, we set

$$\vec{\omega}_{i,0}^m := \vec{\nu}_{i,\frac{1}{2}}^m = \frac{(\vec{X}_{i,1}^m - \vec{X}_{i,0}^m)^\perp}{h_{i,\frac{1}{2}}^m}, \quad \vec{\omega}_{i,N_i}^m := \vec{\nu}_{i,N_i-\frac{1}{2}}^m = \frac{(\vec{X}_{i,N_i}^m - \vec{X}_{i,N_i-1}^m)^\perp}{h_{i,N_i-\frac{1}{2}}^m}. \quad (3.2)$$

As κ_i^{m+1} and \vec{X}_i^{m+1} are piecewise linear, they are uniquely defined by their values at the nodes q_j^i . Therefore, we consider κ^{m+1} and \vec{X}^{m+1} as elements in \mathbb{R}^N and $(\mathbb{R}^2)^N$ with $N = \sum_{i=1}^{N_C} N_i^*$, where $N_i^* = N_i$ for closed curves and $N_i^* = N_i + 1$ for open curves. Similarly, we consider $\vec{X}^m \in (\mathbb{R}^2)^N$ and set $\delta\vec{X}^{m+1} := \vec{X}^{m+1} - \vec{X}^m \in (\mathbb{R}^2)^N$.

An approximation for (2.8a) is given by

$$\frac{1}{\tau_m} \left(\delta\vec{X}_{i,j}^{m+1} \right) \cdot \vec{\omega}_{i,j}^m = \sigma \kappa_{i,j}^{m+1} + F_{i,j}^m \quad (3.3)$$

for $i = 1, \dots, N_C$, $j = 1, \dots, N_i$ if $\partial\Gamma_i^m = \emptyset$ and $j = 0, 1, \dots, N_i$, if $\partial\Gamma_i^m \neq \emptyset$. Here, $F_{i,j}^m := F_i(\vec{X}_{i,j}^m)$ denotes the external forcing term evaluated at the node $\vec{X}_{i,j}^m$.

In order to propose a finite difference approximation of (2.8b), we need to define an approximation of $\vec{x}_{ss}(q_j^i, t_{m+1})$. For $i = 1, \dots, N_C$ and $j = 1, \dots, N_i$, if Γ_i^m is closed, and $j = 1, \dots, N_i - 1$, if Γ_i^m is not closed, we set

$$\Delta_2^{h,m} \vec{X}_{i,j}^{m+1} := \frac{2}{h_{i,j-\frac{1}{2}}^m + h_{i,j+\frac{1}{2}}^m} \left(\frac{\vec{X}_{i,j+1}^{m+1} - \vec{X}_{i,j}^{m+1}}{h_{i,j+\frac{1}{2}}^m} - \frac{\vec{X}_{i,j}^{m+1} - \vec{X}_{i,j-1}^{m+1}}{h_{i,j-\frac{1}{2}}^m} \right). \quad (3.4)$$

In case of equal spatial step sizes $h_{i,j-\frac{1}{2}}^m = h_{i,j+\frac{1}{2}}^m =: h_i^m$, the term reduces to $(\vec{X}_{i,j-1}^m - 2\vec{X}_{i,j}^m + \vec{X}_{i,j+1}^m)/((h_i^m)^2)$.

An approximation of (2.8b) is given by

$$\kappa_{i,j}^{m+1} \vec{\omega}_{i,j}^m = \Delta_2^{h,m} \vec{X}_{i,j}^{m+1}, \quad (3.5)$$

for $i = 1, \dots, N_C$, $j = 1, \dots, N_i$ in case of closed curves and $j = 1, \dots, N_i - 1$ in case of open curves.

In case of $\partial\Gamma_i^m \neq \emptyset$, the boundary points belong either to a triple junction $\vec{\Lambda}_k$, $k \in \{1, \dots, N_T\}$, or to a boundary intersection point $\vec{Q}_k \in \partial\Omega$, $k \in \{1, \dots, N_I\}$. Let i_1, i_2, i_3 denote the indices of three curves meeting at a triple point $\vec{\Lambda} \in \{\vec{\Lambda}_1, \dots, \vec{\Lambda}_{N_T}\}$. For the ease of illustration, assume that the initial points of the curves meet at the junction. The attachment condition (2.9a) enforces

$$\vec{X}_{i_1,0}^{m+1} = \vec{X}_{i_2,0}^{m+1} = \vec{X}_{i_3,0}^{m+1}. \quad (3.6)$$

Let $\vec{Q}_k \in \partial\Omega$ be a boundary intersection point and i a curve index such that $\vec{Q}_k = \vec{X}_{i,0}^m$. The attachment condition (2.10a) is approximated by

$$\left(\vec{X}_{i,0}^{m+1} - \vec{X}_{i,0}^m \right) \cdot \vec{n}_{\partial\Omega}(\vec{X}_{i,0}^m) = 0. \quad (3.7)$$

If the equation above is satisfied, the curve will only move in direction tangential to $\partial\Omega$. As the image domain Ω is rectangular, $\partial\Omega$ is locally flat and \vec{Q}_k keeps attached to $\partial\Omega$.

The considerations above motivate to define an operator

$$\begin{aligned} \vec{P} : (\mathbb{R}^2)^N \rightarrow \mathbb{X} := \{ (\vec{z}_1, \dots, \vec{z}_{N_C}) \in (\mathbb{R}^2)^N : [\vec{z}_{i_{k,1}}]_{j_{k,1}} = [\vec{z}_{i_{k,2}}]_{j_{k,2}} = [\vec{z}_{i_{k,3}}]_{j_{k,3}}, k = 1, \dots, N_T, \\ \text{and } [\vec{z}_{i_{I,k}}]_{j_{I,k}} \cdot \vec{n}_{\partial\Omega}(\vec{X}_{i_{I,k},j_{I,k}}^m) = 0, k = 1, \dots, N_I \}, \end{aligned}$$

which is the orthogonal projection onto \mathbb{X} . In the definition of \mathbb{X} , $\vec{z}_i \in (\mathbb{R}^2)^{N_i^*}$ and $[\vec{z}_i]_j \in \mathbb{R}^2$ is the j -th component of the vector \vec{z}_i . We use an index notation according to the notation introduced in Section 2.2.

In order to state a matrix formulation for the discrete system (3.3) and (3.5), we introduce the following matrices

$$M := \begin{pmatrix} M^1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & M^{N_C} \end{pmatrix}, \vec{N} := \begin{pmatrix} \vec{N}^1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \vec{N}^{N_C} \end{pmatrix}, \vec{A} := \begin{pmatrix} \vec{A}^1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \vec{A}^{N_C} \end{pmatrix},$$

where $M^i \in \mathbb{R}^{N_i^* \times N_i^*}$, $\vec{N}^i \in (\mathbb{R}^2)^{N_i^* \times N_i^*}$, $\vec{A}^i \in (\mathbb{R}^{2 \times 2})^{N_i^* \times N_i^*}$, $i = 1, \dots, N_C$, are defined by

$$\begin{aligned} M_{j,l}^i &= \frac{1}{2} (h_{i,j-\frac{1}{2}}^m + h_{i,j+\frac{1}{2}}^m) \delta_{jl}, & \vec{N}_{j,l}^i &= \frac{1}{2} (h_{i,j-\frac{1}{2}}^m + h_{i,j+\frac{1}{2}}^m) \vec{\omega}_{i,j}^m \delta_{jl}, \\ \vec{A}_{j,j}^i &= \left(\frac{1}{h_{i,j-\frac{1}{2}}^m} + \frac{1}{h_{i,j+\frac{1}{2}}^m} \right) \vec{\text{Id}}_{2 \times 2}, & \vec{A}_{j,j-1}^i &= -\frac{1}{h_{i,j-\frac{1}{2}}^m} \vec{\text{Id}}_{2 \times 2}, \\ \vec{A}_{j,j+1}^i &= -\frac{1}{h_{i,j+\frac{1}{2}}^m} \vec{\text{Id}}_{2 \times 2}, & \vec{A}_{j,l}^i &= \vec{0}, \text{ for } l \notin \{j-1, j, j+1\}, \end{aligned}$$

if $j \notin \{0, N_i\}$ for open curves. For open curves, we set in addition

$$\begin{aligned}
M_{0,l}^i &= \frac{1}{2} h_{i,\frac{1}{2}}^m \delta_{0l}, & \vec{N}_{0,l}^i &= \frac{1}{2} h_{i,\frac{1}{2}}^m \vec{\omega}_{i,0}^m \delta_{0l}, \\
\vec{A}_{0,0}^i &= \frac{1}{h_{i,\frac{1}{2}}^m} \vec{\text{Id}}_{2 \times 2}, & \vec{A}_{0,1}^i &= -\frac{1}{h_{i,\frac{1}{2}}^m} \vec{\text{Id}}_{2 \times 2}, \\
M_{N_i,l}^i &= \frac{1}{2} h_{i,N_i-\frac{1}{2}}^m \delta_{N_i l}, & \vec{N}_{N_i,l}^i &= \frac{1}{2} h_{i,N_i-\frac{1}{2}}^m \vec{\omega}_{i,N_i}^m \delta_{N_i l}, \\
\vec{A}_{N_i,N_i}^i &= \frac{1}{h_{i,N_i-\frac{1}{2}}^m} \vec{\text{Id}}_{2 \times 2}, & \vec{A}_{N_i,N_i-1}^i &= -\frac{1}{h_{i,N_i-\frac{1}{2}}^m} \vec{\text{Id}}_{2 \times 2}, \\
\vec{A}_{0,l}^i &= \vec{0}, \text{ for } l \notin \{0, 1\}, & \vec{A}_{N_i,l}^i &= \vec{0}, \text{ for } l \notin \{N_i, N_i - 1\}.
\end{aligned}$$

In the terms above, $\vec{\text{Id}}_{2 \times 2} \in \mathbb{R}^{2 \times 2}$ denotes the identity matrix, $\vec{0} \in \mathbb{R}^{2 \times 2}$ a matrix with all entries equal to zero and δ_{jl} denotes the Kronecker symbol.

Further, we define $b^m = (b_1^m, \dots, b_{N_C}^m) \in \mathbb{R}^N$ by

$$b_{i,j}^m = \frac{1}{2} (h_{i,j-\frac{1}{2}}^m + h_{i,j+\frac{1}{2}}^m) F_{i,j}^m, \quad \text{resp.} \quad b_{i,j}^m = \frac{1}{2} h_{i,j \pm \frac{1}{2}}^m F_{i,j}^m, \quad (3.8)$$

where the latter term holds in case of boundary nodes.

We propose the following linear system which includes the discrete scheme (3.3) and (3.5), the attachment conditions (3.6) and (3.7) and approximations of Young's law at triple junctions and of the 90 degrees angle condition at boundary intersection points: Find $\kappa^{m+1} \in \mathbb{R}^N$ and $\delta X^{m+1} \in \mathbb{X}$ such that

$$\begin{pmatrix} -\sigma \tau_m M & \vec{N}^T \vec{P} \\ \vec{P} \vec{N} & \vec{P} \vec{A} \vec{P} \end{pmatrix} \begin{pmatrix} \kappa^{m+1} \\ \delta \vec{X}^{m+1} \end{pmatrix} = \begin{pmatrix} \tau_m b^m \\ -\vec{P} \vec{A} \vec{X}^m \end{pmatrix}. \quad (3.9)$$

The attachment conditions at triple junctions and boundary intersection points are satisfied by $\delta \vec{X}^{m+1} \in \mathbb{X}$ on assuming that \vec{X}^0 fulfills the attachment condition at triple junctions. Recall, that $\vec{P}^T = \vec{P}$ and \vec{P} is the identity on \mathbb{X} . Approximations of Young's law and the angle condition at boundary intersection points can be derived from the second equation in (3.9). We now consider a triple junction $\vec{\Lambda} \in \{\vec{\Lambda}_1, \dots, \vec{\Lambda}_{N_T}\}$ with $\vec{X}_{i_1,0}^{m+1} = \vec{X}_{i_2,0}^{m+1} = \vec{X}_{i_3,0}^{m+1} = \vec{\Lambda}$. The second equation of the system (3.9) provides

$$\sum_{l=1}^3 \frac{1}{2} h_{i_l, \frac{1}{2}}^m \vec{\omega}_{i_l, 0}^m \kappa_{i_l, 0}^{m+1} + \sum_{l=1}^3 \frac{1}{h_{i_l, \frac{1}{2}}^m} \left(\vec{X}_{i_l, 0}^{m+1} - \vec{X}_{i_l, 1}^{m+1} \right) = 0. \quad (3.10)$$

For $h^m \rightarrow 0$, the first sum approaches zero, whereas the second sum approaches $-(\vec{\tau}_{i_1}(\vec{\Lambda}) + \vec{\tau}_{i_2}(\vec{\Lambda}) + \vec{\tau}_{i_3}(\vec{\Lambda}))$. In the limit, we therefore can derive Young's law from (3.10)

$$\vec{\tau}_{i_1}(\vec{\Lambda}) + \vec{\tau}_{i_2}(\vec{\Lambda}) + \vec{\tau}_{i_3}(\vec{\Lambda}) = 0. \quad (3.11)$$

Consider a boundary intersection point $\vec{Q} \in \{\vec{Q}_1, \dots, \vec{Q}_{N_I}\}$. Let i be a curve index such that $\vec{X}_{i,0}^m = \vec{Q}$. Then, the second equation in (3.9) and the definition of \vec{P} provides

$$\left(\vec{\text{Id}}_{2 \times 2} - \vec{n}_{\partial\Omega}(\vec{Q}) \otimes \vec{n}_{\partial\Omega}(\vec{Q}) \right) \left(\frac{1}{2} h_{i, \frac{1}{2}}^m \vec{\omega}_{i, 0}^m \kappa_{i, 0}^{m+1} + \frac{1}{h_{i, \frac{1}{2}}^m} \left(\vec{X}_{i, 0}^{m+1} - \vec{X}_{i, 1}^{m+1} \right) \right) = 0. \quad (3.12)$$

The limit of these terms, as h^m approaches 0, is

$$\vec{\tau}_i(\vec{Q}) - \left(\vec{\tau}_i(\vec{Q}) \cdot \vec{n}_{\partial\Omega}(\vec{Q}) \right) \vec{n}_{\partial\Omega}(\vec{Q}) = 0. \quad (3.13)$$

Thus, $\vec{\tau}_i(\vec{Q})$ is parallel to $\vec{n}_{\partial\Omega}(\vec{Q})$. Consequently, the curve meets the external boundary with a 90 degrees angle at \vec{Q} .

As M is non-singular, (3.9) can be reformulated to

$$\kappa^{m+1} = \frac{1}{\sigma\tau_m} M^{-1} \left(\vec{N}^T \vec{P} \delta \vec{X}^{m+1} - \tau_m b^m \right), \quad (3.14a)$$

$$\left(\vec{P} \vec{A} \vec{P} + \frac{1}{\sigma\tau_m} \vec{P} \vec{N} M^{-1} \vec{N}^T \vec{P} \right) \delta \vec{X}^{m+1} = \frac{1}{\sigma} \vec{P} \vec{N} M^{-1} b^m - \vec{P} \vec{A} \vec{X}^m, \quad (3.14b)$$

by applying a Schur complement approach. The linear equation (3.14b) can be solved with an iterative solver, for example with the method of conjugate gradients with possible preconditioning, or with a direct solver for sparse matrices.

Under very mild assumptions one can show that the system matrix in (3.14b) is symmetric, positive definite and hence a unique solution of (3.14b) exists. A detailed proof of existence and uniqueness is given in Barrett, Garcke, and Nürnberg (2007a).

3.2 Semi-discrete scheme

We now consider a scheme which is discrete in space and continuous in time. Therefore let $\vec{X} = (\vec{X}_1, \dots, \vec{X}_{N_C})$, $\kappa = (\kappa_1, \dots, \kappa_{N_C})$, such that $\vec{X}_i : I_i \times [0, T] \rightarrow \mathbb{R}^2$, and $\kappa_i : I_i \times [0, T] \rightarrow \mathbb{R}$, $i = 1, \dots, N_C$, are piecewise linear on $[q_{j-1}^i, q_j^i]$, $j = 1, \dots, N_i$. We make use of a similar notation as in the fully discrete case by just omitting the superscripts m and $m + 1$. All quantities are time-dependent in this section. We now state a result which demonstrates that the semi-discrete scheme leads to equidistributed meshes.

Lemma 3.1. *The semi-discrete scheme*

$$(\vec{X}_{i,j})_t \cdot \vec{\omega}_{i,j} = \sigma \kappa_{i,j} + F_{i,j}, \quad (3.15a)$$

$$\kappa_{i,j} \vec{\omega}_{i,j} = \Delta_2^h \vec{X}_{i,j}, \quad (3.15b)$$

for $i = 1, \dots, N_C$, $j = 1, \dots, N_i$, if Γ_i is open and $j = 1, \dots, N_i - 1$, if Γ_i is closed, provides an equidistribution of the mesh points along those parts of the curves Γ_i which are not locally flat.

The proof is given in Barrett et al. (2007b) (Remark 2.3). For the fully discrete scheme (3.3) and (3.5) we cannot prove the equidistribution of mesh points. However practical experiments show that the nodes are good distributed and therefore no *local* refinement or coarsening strategy needs to be developed. As a curve can grow and shrink during the evolution, a *global* refinement or coarsening is recommended. For that, the ratio of curve length and number of node points is considered.

3.3 Topological changes

During the evolution of curves the following topological changes can occur: (1) Splitting of a curve into two curves, (2) merging of two curves to one single curve, (3) emergence of a triple junction and (4) emergence of a boundary intersection point. If a triple junction is detected, a new curve will be created. Similarly, if a curve length shrinks, a curve has to be deleted. If the curve connects two triple junctions, the triple junctions will also be deleted and other curves which previously ended at the junctions have to be adapted. Therefore, the number of curves N_C^m , the number of nodes N_i^m , $i = 1, \dots, N_C^m$, the number of triple junctions N_T^m and of boundary intersection points N_I^m are time dependent.

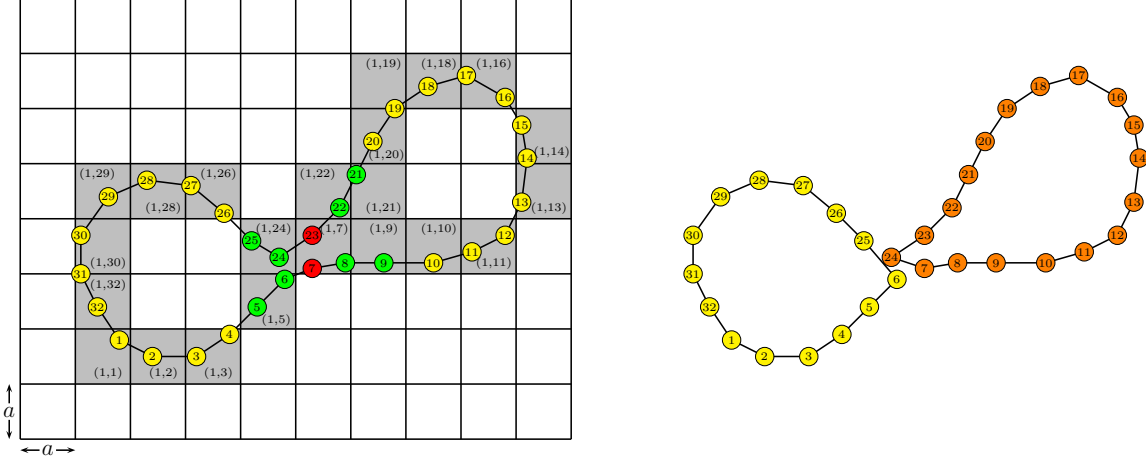


Figure 3: Left: Detection of splitting of a curve Γ_1^m near the nodes 7 and 23, Right: Splitting of the curve in two curves

Using a parametric approach, topological changes are not automatically handled in contrast to level-set methods. Rather, topological changes can be interpreted as singularities. We propose a method how to continue after such singularities occur. The detection of the topological changes used in this paper is close to the method proposed by Balažovjeh et al. (2012) who consider the case without triple junctions and boundary intersections. We generalize their approach such that also junctions and boundary points can be dealt with.

To detect a change in topology, we construct a uniform background grid which covers the rectangular image domain Ω . The grid size $a > 0$ can be adaptively chosen, for example it can be set according to the minimum or average distance between two neighboring node points of the polygonal curves. The grid can be stored as a sparse matrix or two dimensional array where the elements correspond to squares of the grid of size $a \times a$. Using a background grid, topological changes can efficiently be detected in two steps:

- (i) For $i = 1, \dots, N_C^m$ and $j = 0, \dots, N_i^m$, we initialize the square in which $\vec{X}_{i,j}^m$ lies with $(-1, -1)$. If the square is close to the boundary of $\partial\Omega$, the node is marked as a boundary intersection point. Therefore it is stored in a list for boundary intersection points such that we can handle several topological changes at once.
- (ii) In a second loop, we again consider for $i = 1, \dots, N_C^m$ and $j = 0, \dots, N_i^m$ the corresponding square of the grid. If the square is marked with $(-1, -1)$, the square marking is overwritten with (i, j) . If the square has already been marked with (i_1, j_1) , a topological change is likely to occur close to the nodes. If $i = i_1$ and if the two nodes are direct neighbor points or have a common neighbor, no topological change is detected. Otherwise, a set of a limited number n of neighbor nodes around each $\vec{X}_{i,j}^m$ and \vec{X}_{i_1,j_1}^m is considered. As the two sets contain only n nodes, the pair (i, j^*) and (i_1, j_1^*) with the smallest distance can be quickly found. In practice, $n = 5$ is a good choice. If $i = i_1$, a splitting of the curve Γ_i^m occurs and the pair (i, j^*) and (i_1, j_1^*) is stored in a list for splitting points. If $i \neq i_1$, we consider $k^+(i), k^-(i), k^+(i_1)$ and $k^-(i_1)$, i.e. the indices of the regions Ω_k , $k \in \{1, \dots, N_R\}$, separated by Γ_i^m and $\Gamma_{i_1}^m$, respectively. Recall, that the normal \vec{v}_i^m at the curve Γ_i^m points from $k^-(i)$ to $k^+(i)$. If $k^+(i) = k^+(i_1)$ and $k^-(i) = k^-(i_1)$, a merging of two curves occurs. The pair is stored in a list for merging points. We exclude the case $k^+(i) = k^-(i_1)$ and $k^-(i) = k^+(i_1)$, as this situation can never occur when taking care of the orientation of the initial curves. In case of $k^+(i) \neq k^+(i_1)$ or $k^-(i) \neq k^-(i_1)$, triple junctions occur and the pair of nodes is stored in a list for triple junctions.

Topological changes can thus be detected efficiently performing only two loops over all mesh points. Consequently, the computational effort for detecting topological changes remains small, more precisely it is $\mathcal{O}(N)$. The algorithm can handle several topological changes at once. The lists of split, merge, triple and boundary points can be sequentially considered. In case of splitting and merging of curves, the neighbor relation is modified by connecting node j^* with $j_1^* + 1$ and j_1^* with $j^* + 1$.

Figure 3 presents an example where a curve splits up in two single curves: The node $(1, 23)$ lies in a square which has been previously marked with $(1, 7)$. Among the mesh points in a neighborhood of $(1, 7)$ and in a neighborhood of $(1, 23)$, the pair $(1, 6)$ and $(1, 24)$ is identified as the pair of nodes with smallest distance. The neighbor relationships at $(1, 6)$ and $(1, 24)$ need to be modified as shown in the right sub-figure. To prevent another splitting and merging near the same point in Ω , a square can be blocked for new topological changes for some time-steps after a topological change has taken place.

If a boundary point occurs, the point is first projected orthogonal to the boundary $\partial\Omega$ and is then duplicated. If the previous curve is closed, the curve becomes an open curve. If the curve has already been an open curve, it is separated in two single curves. The two boundary nodes, the detected node and its duplication, are the end point of one curve and the first point of the second curve. They are allowed to move away from each other along $\partial\Omega$ in the next time steps. Similarly, the complementary situation can be handled. An open curve can become a closed curve, if the two boundary points are too close to each other, i.e. lie in the same square of the background grid. Two curves can merge if their boundary points are too close to each other. Again, the square where a boundary intersection has taken place can be shortly blocked.

In case of triple junctions, the topological change is handled as follows: First, as an intermediate step, a quadruple junction emerges at the point where Γ_i^m and $\Gamma_{i_1}^m$ touch. The quadruple junction is replaced by two triple junctions connected with a small curve, see Figure 4. The small curve consists of only two initial points, namely $0.5(\vec{X}_{i,j^*}^m + \vec{X}_{i_1,j_1^*}^m)$ and $0.5(\vec{X}_{i,j^*\pm 1}^m + \vec{X}_{i_1,j_1^*\pm 1}^m)$, where the sign \pm is chosen according to the orientation of the curves. Within the next time steps, the new curve can grow and new nodes are inserted by some global refinements.

Another group of topological changes occur when a curve's length falls below a minimum length. If the curve is closed or if its two boundary nodes lie both on $\partial\Omega$, it is directly deleted. If both end nodes belong to triple junctions, the curve is deleted and an intermediate quadruple junction occurs. Dependent on the phase indices of the adjacent regions Ω_k , $k \in \{1, \dots, N_R\}$, and of the previous curve network, there are different ways how to continue, see Figure 5. If a curve is marked for deletion with one boundary belonging to a triple junction and one boundary located at $\partial\Omega$, the curve and the triple junction are deleted. Further, the boundary points of the remaining two curves which have met at the triple junction are projected to the boundary $\partial\Omega$. In the next time steps, they are allowed to move away from each other.

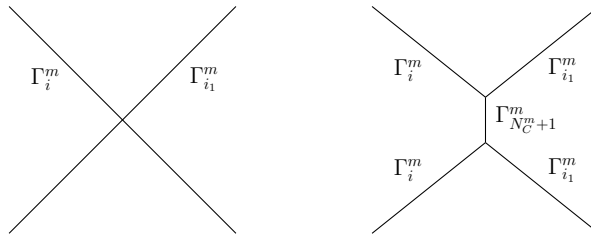


Figure 4: Conversion of a quadruple junction to a network with two triple junctions

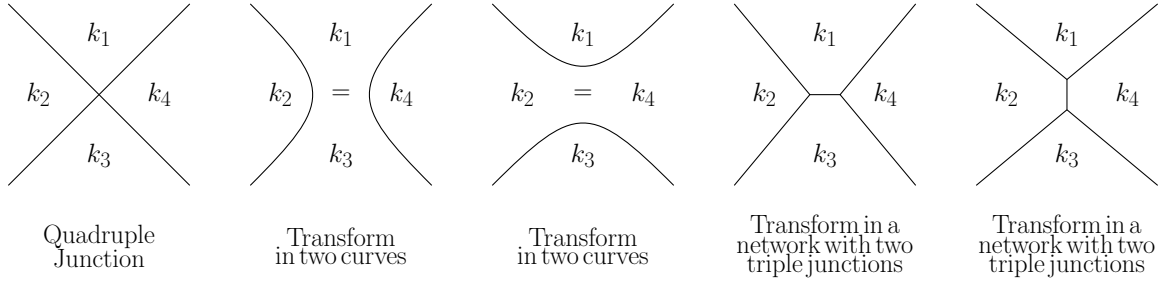


Figure 5: Quadruple junction and several possibilities for continuation

3.4 Computations of regions and coefficients

As the curves evolve in time, an approximation Ω_k^m of the regions $\Omega_k(t_m)$ and an approximation c_k^m of the coefficients $c_k(t_m)$ (cf. (2.5)), $k = 1, \dots, N_R$, has to be computed. As the image function $u_0 : \Omega \rightarrow \mathbb{R}$ is locally constant at the pixels, we assign each pixel to a phase Ω_k^m . If a pixel is truncated by a curve, it is assigned to the phase to which the largest part belongs. Let S_k^m be the set of n_k^m pixels belonging to Ω_k^m . Then the approximation c_k^m is set to

$$c_k^m := \frac{C_k^m}{n_k^m}, \quad C_k^m := \sum_{pix \in S_k^m} u_0|_{pix}. \quad (3.16)$$

The entire image domain needs to be considered only for $m = 0$. As the curves move only slightly from one time step to the next, the regions Ω_k^m and the coefficients c_k^m need to be updated only in a small environment of the curves. This results in a very efficient computation of the regions and the coefficients. As the normal $\bar{\nu}_i^m$ points from $\Omega_{k^-(i)}^m$ to $\Omega_{k^+(i)}^m$, the pixels close to the curve Γ_i^m are assigned to the phase $k^+(i)$ or $k^-(i)$, respectively.

As initialization we set $n_k^m = n_k^{m-1}$ and $C_k^m = C_k^{m-1}$ for $k = 1, \dots, N_R$. For $i = 1, \dots, N_C$, all pixels in a environment of Γ_i^m are subsequently considered. Let a pixel pix be assigned to phase $k \in \{k^+(i), k^-(i)\}$ and let $l \neq k$ be the former phase index of the pixel. We set

$$n_k^m = n_k^{m-1} + 1, \quad n_l^m = n_l^{m-1} - 1, \quad C_k^m = C_k^{m-1} + u_0|_{pix}, \quad C_l^m = C_l^{m-1} - u_0|_{pix}. \quad (3.17)$$

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

Similarly, the coefficients in case of colored images can be computed. The coefficients are all means or normalized means of the corresponding component of the image function. In case of CB or HSV color space, the value $u_0|_{pix}$ needs to be first transformed to the CB and HSV color. If image components lie on S^1 or S^2 , the computation needs to be slightly modified: For example, in case of the HSV space, the hue coefficients are computed by $c_k^m = C_k^m/\|C_k^m\|$ with $C_k^m = \sum_{pix \in S_k^m} h_0|_{pix}$, where $h_0 : \Omega \rightarrow S^2$ is the hue component of the image.

3.5 Solution of the Image Denoising Problem

Besides the segmentation of the image, the algorithm provides a piecewise constant approximation $u = \sum_{k=1}^{N_R} c_k^m \chi_{\Omega_k^m}$ of u_0 . A piecewise smooth approximation can be obtained by solving a discretization of the boundary value problem (2.12). For the bulk equations, a finite difference approximation can be used.

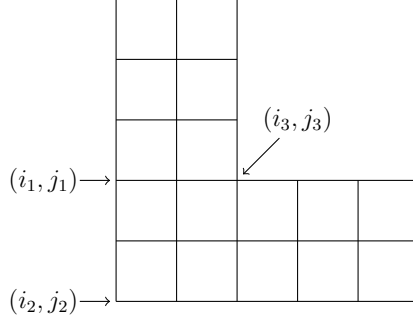


Figure 6: A possible region Ω_k^m

Therefore, we consider a spatial discretization $\Omega^h := \{(ih, jh) : i = 0, \dots, N_x, j = 0, \dots, N_y\}$, $N_x, N_y \in \mathbb{N}$, of the rectangular image domain Ω . As we have to solve an equation on each phase separately, we define the discrete set $\Omega_k^h := \Omega^h \cap \overline{\Omega_k^m}$. In the following, we consider k as fixed. We aim at finding a function u^h minimizing a discrete analogue of the energy (2.11).

We define for $i = 1, \dots, N_x, j = 1, \dots, N_y$

$$\begin{aligned} A_x(i, j) &= \text{area} \left(\left([(i-1)h, ih] \times [(j-\frac{1}{2})h, (j+\frac{1}{2})h] \right) \cap \Omega_k^m \right), \\ A_y(i, j) &= \text{area} \left(\left([(i-\frac{1}{2})h, (i+\frac{1}{2})h] \times [(j-1)h, jh] \right) \cap \Omega_k^m \right), \end{aligned}$$

and for $i = 0, 1, \dots, N_x, j = 0, 1, \dots, N_y$

$$A(i, j) = \text{area} \left(\left([(i-\frac{1}{2})h, (i+\frac{1}{2})h] \times [(j-\frac{1}{2})h, (j+\frac{1}{2})h] \right) \cap \Omega_k^m \right),$$

where area denotes the two-dimensional Lebesgue measure.

We define the following discrete energy

$$\begin{aligned} E_{\text{discr},k}(u^h) &= \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \left(A_x(i, j) \left(\frac{u_{i,j}^h - u_{i-1,j}^h}{h} \right)^2 + A_y(i, j) \left(\frac{u_{i,j}^h - u_{i,j-1}^h}{h} \right)^2 \right) \\ &\quad + \sum_{i=0}^{N_x} \sum_{j=0}^{N_y} \lambda_k A(i, j) \left(u_{i,j}^h - u_0(ih, jh) \right)^2 \end{aligned} \quad (3.18)$$

which is a discrete analogue of $\int_{\Omega_k} ((u_x^2 + u_y^2) + \lambda_k(u - u_0)^2) dx$. For each $(ih, jh) \in \Omega_k^h$ we take the derivative with respect to $u_{i,j}^h$ and set the resulting term to zero. This leads to a linear system. Recall that Ω_k^m need not be rectangular. Geometries like the one shown in Figure 6 can occur.

At interior nodes (ih, jh) , we get the following equation considering the derivative with respect to $u_{i,j}^h$:

$$(4 + \lambda_k h^2) u_{i,j}^h - u_{i-1,j}^h - u_{i+1,j}^h - u_{i,j-1}^h - u_{i,j+1}^h = \lambda_k h^2 u_0(ih, jh).$$

At boundary points like $(i_1, j_1), (i_2, j_2), (i_3, j_3)$, we get the following equations from (3.18):

$$\begin{aligned} \left(2 + \frac{1}{2}\lambda_k h^2\right) u_{i_1, j_1}^h - u_{i_1+1, j_1}^h - \frac{1}{2}u_{i_1, j_1-1}^h - \frac{1}{2}u_{i_1, j_1+1}^h &= \frac{1}{2}\lambda_k h^2 u_0(i_1 h, j_1 h), \\ \left(1 + \frac{1}{4}\lambda_k h^2\right) u_{i_2, j_2}^h - \frac{1}{2}u_{i_2+1, j_2}^h - \frac{1}{2}u_{i_2, j_2+1}^h &= \frac{1}{4}\lambda_k h^2 u_0(i_2 h, j_2 h), \\ \left(3 + \frac{3}{4}\lambda_k h^2\right) u_{i_3, j_3}^h - u_{i_3-1, j_3}^h - \frac{1}{2}u_{i_3+1, j_3}^h - u_{i_3, j_3-1}^h - \frac{1}{2}u_{i_3, j_3+1}^h &= \frac{3}{4}\lambda_k h^2 u_0(i_3 h, j_3 h). \end{aligned} \quad (3.19)$$

At boundary points we can derive the Neumann boundary conditions from the linear equations considering the limit $h \rightarrow 0$. For example, rewriting (3.19) as follows

$$-\frac{u_{i_1+1, j_1}^h - u_{i_1, j_1}^h}{h} + \frac{u_{i_1, j_1}^h - u_{i_1, j_1-1}^h}{2h} - \frac{u_{i_1, j_1+1}^h - u_{i_1, j_1}^h}{2h} = \frac{1}{2}\lambda_k h \left(u_0(i_1 h, j_1 h) - u_{i_1, j_1}^h\right).$$

and considering $h \rightarrow 0$ yields the Neumann boundary condition $u_x = 0$. In summary, we have a linear system whose system matrix is sparse and strictly diagonally dominant as $\lambda_k > 0$. Thus, the linear system has a unique solution.

The pixel grid ($h = 1$) serves as natural grid for a spatial discretization. The numerical solution u^h is determined for all corner points of pixels. Consider a pixel in Ω_k^m with corner points (i, j) , $(i-1, j)$, $(i-1, j-1)$ and $(i, j-1)$. Let $p_{i,j}$ be the center of the pixel. Then a denoised image function is defined by

$$u^h(p_{i,j}) = \frac{1}{4} \left(u_{i,j}^h + u_{i-1,j}^h + u_{i-1,j-1}^h + u_{i,j-1}^h\right). \quad (3.20)$$

Color images are denoised component-wise, i.e. each channel is treated like a scalar image.

It is sufficient to perform the image smoothing for $m = M$, i.e. as a post-processing step, when the region interfaces Γ_i^M match approximately with the edges of the objects in the image.

3.6 Summary of the algorithm

In summing up, we propose the following algorithm for image segmentation. Given a set of polygonal curves $\Gamma^0 = (\Gamma_1^0, \dots, \Gamma_{N_C}^0)$ and $\vec{X}^0 = (\vec{X}_1^0, \dots, \vec{X}_{N_C}^0)$ with $\vec{X}_i^0(I_i) = \Gamma_i^0$, perform the following steps for $m = 0, 1, \dots, M-1$:

- (i) Compute the regions Ω_k^m and the coefficients c_k^m , $k = 1, \dots, N_R$, as described in Section 3.4.
- (ii) Compute b^m as defined in (3.8) by using the coefficients c_k^m of step (i). Compute $\vec{X}^{m+1} = \vec{X}^m + \delta \vec{X}^{m+1}$ by solving the linear equation (3.14b), see Section 3.1.
- (iii) Check if topological changes occur, see Section 3.3. In case of a topological change, repeat the steps (i) and (ii) n_{sub} -times with a step size of τ_m/n_{sub} and execute the topological change when it occurs in a sub-step.
- (iv) Compute the length of the polygonal curves Γ_i^m . If $|\Gamma_i^m|/N_i^m < l_{\min}$, perform a global coarsening, and if $|\Gamma_i^m|/N_i^m > l_{\max}$, perform a global refinement of the curve discretization.

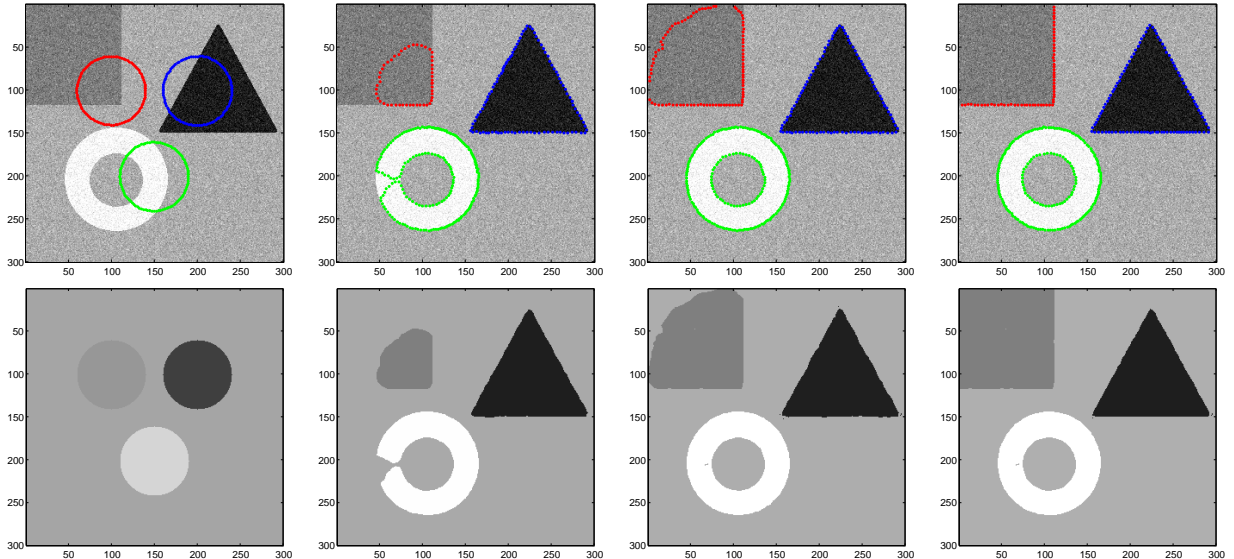


Figure 7: Multi-phase image segmentation of a gray-scaled image with multiple objects to be detected, $m = 1, 560, 2150, 3000$, $\Delta t = 0.1$, $\lambda = 20$, σ -factor 20%

We can use a constant parameter σ which weights the curvature term in (2.8a). Alternatively, the parameter can be set adaptively: For that, the internal energy $\sigma|\Gamma|$ in the energy functional (2.4) is set to a certain percentage of the external energy $E(\Gamma, c_1, \dots, c_{N_R}) - \sigma|\Gamma|$. The percentage is shortly called σ -factor. The energies and the new value for σ need not be updated at each time step. In the results presented in the next section, the factor σ is computed either every 10th or every 50th time step.

Having found a final segmentation, a smooth approximation of the image is computed as described in Section 3.5.

4 Results

We apply the image segmentation method of Section 2 and 3 on artificial and real, gray-scaled and color images. The UMFPACK algorithm as MATLAB built-in routine is used to solve the linear equation (3.14b), see also Davis (2004).

In a first experiment, a gray-scaled image with three objects is segmented, see Figure 7. Typically, the initial contours are closed curves which evolve with time according to the evolution equation (2.8a). The first row in Figure 7 shows the given image u_0 and the contours for four different iteration steps. The second row shows the piecewise constant, approximating image given by (2.3). The gray value in the region Ω_k , $k = 1, \dots, N_R$ (here $N_R = 4$), is the mean of u_0 in Ω_k , recall (2.5). Having detected the objects, the contours match with the edges of the objects and the approximating image is a smooth, piecewise constant version of u_0 .

This sample image also demonstrates two kinds of topological changes: splitting of a curve and creation of boundary intersection points. The green curve splits up into two single sub-curves. At time step $m = 560$, two parts of the curve nearly touch. The splitting is detected as described in Section 3.3. The red contour in Figure 7 intersects the image boundary $\partial\Omega$ at two positions. Two open curves with each two boundary intersection points exist at time step $m = 2150$. The length of the smaller sub-curve decreases continuously in the following time steps and the curve is deleted

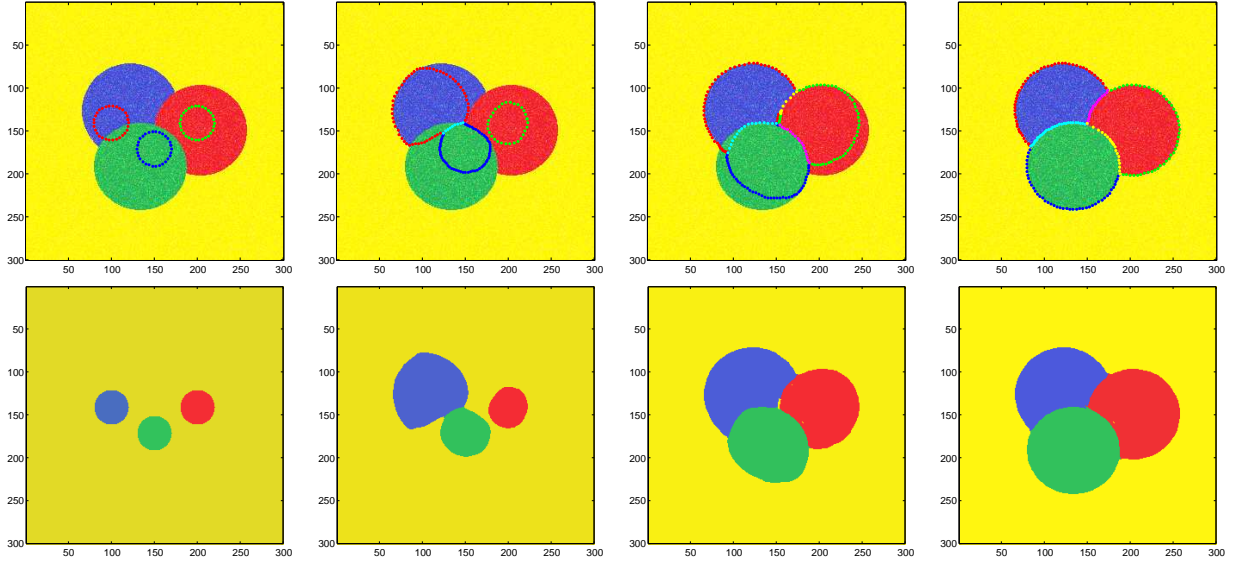


Figure 8: Multi-phase image segmentation of a color image using RGB space, $m = 1, 100, 200, 300$, $\Delta t = 0.1$, $\lambda_1 = \lambda_2 = \lambda_3 = 5$, σ -factor 25%

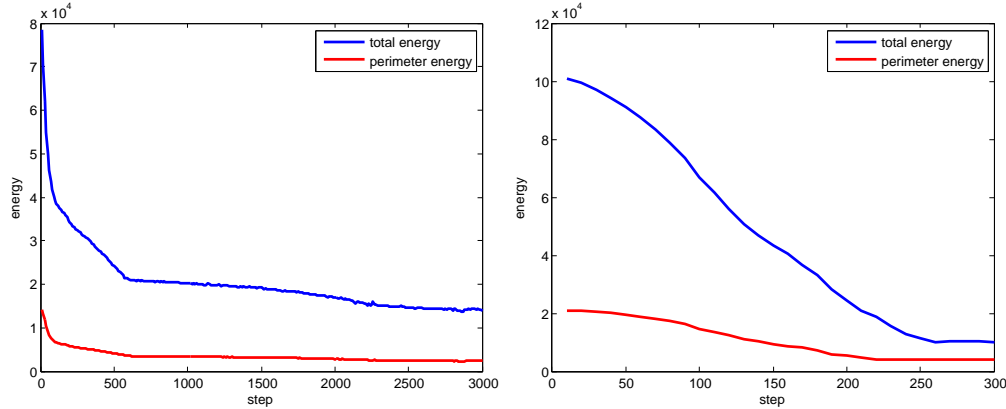


Figure 9: Energy decrease for test images of Figure 7 (left) and Figure 8 (right)

close to the upper left corner of the image. The remaining curve is attracted to the boundary of the upper left gray square.

Figure 8 presents the segmentation results applying the algorithm on a color image. For this experiment, the RGB space is used and for all color channels $i \in \{1, 2, 3\}$ the parameter $\lambda_i = 5$ is used. When two different curves touch, an intermediate quadruple junction occurs. A new small curve is created and the quadruple junction is replaced by two triple junctions connected by the new contour as illustrated in Section 3.3, Figure 4.

The Mumford-Shah energy (2.4) for piecewise constant approximations u is shown in Figure 9. Additional to the total energy, the perimeter energy $\sigma|\Gamma|$ is plotted to visualize the proportion of the length term which ensures smoothness of the curve. The plots show the energy decrease when segmenting the gray-scaled test image of Figure 7 and the color image of Figure 8. The energy plotted on the left decreases fast at the beginning. After approximately 600 steps, the ring and the triangle are both detected. The difference between the gray value of the square and the gray value of the background is small. Therefore, the segmentation of the square is much slower compared to the segmentation of the black triangle and the white ring. The energy decreases only slightly after the two objects have already been detected. The segmentation of the color image of Figure 8 is

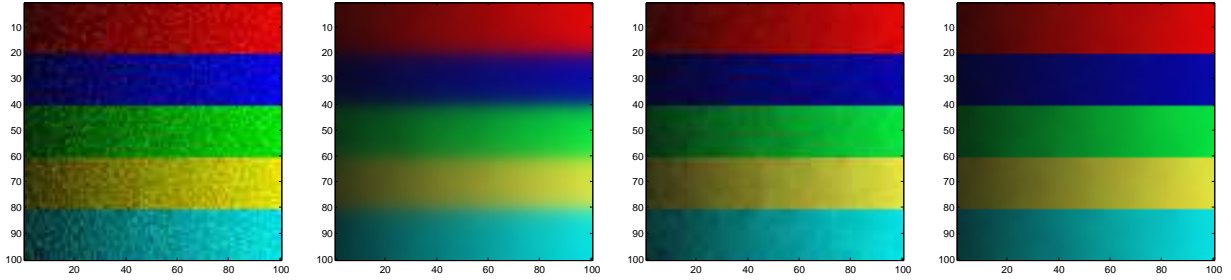


Figure 10: Test image denoising: original image, image denoising without edge enhancement with $\lambda = 0.1$, image denoising result with edge enhancement with $\lambda = 1$ and $\lambda = 0.1$

finished after 300 time steps. In this test image the color of the different objects significantly differ. Therefore, the regions are easy to detect.

In Section 2.3 we proposed a method for image smoothing. The partial differential equation $-\frac{1}{\lambda}\Delta u + u = u_0$ with Neumann boundary conditions is solved on each phase separately. Having previously detected the regions, the image smoothing is performed as a post-processing step. Figure 10 shows a noisy image (1st sub-figure) with five differently colored stripes. The brightness increases from left to right. The second sub-figure to the left shows the result solving the diffusion equation on Ω neglecting the previously detected regions. The edges are strongly smoothed out.

This motivates to search for an approximation of the image which enhances the edges. It is necessary to use a piecewise smooth approximation since a piecewise constant approximation would be a too strong simplification and the brightness change of the original image would get lost. The smoothing effect severely depends on the parameter λ . If λ is large, $\frac{1}{\lambda}$ is small and the approximation u is close to the original image u_0 . The smaller λ on the contrary, the bigger is the smoothing effect. Therefore, choosing $\lambda = 1$ the noise is not completely smoothed out, cf. third sub-figure in Figure 10. Choosing $\lambda = 0.1$ results in a smooth approximation u of u_0 , cf. forth sub-figure. The change in the brightness from left to right is still conserved. By solving the diffusion equation with Neumann boundary conditions separately in each phase, the edges remain sharp.

Additional to artificial test images, the segmentation technique is applied on real images. Figure 11 presents the segmentation of a medical image. It shows the original image and the segmentation for $m = 1, 100, 500$ and 1500 (1st row) and the piecewise constant approximation (2nd row). The parameter λ which weights the external forcing term is chosen high ($\lambda = 400$), as the brightness differences of some objects and the background is small. The σ -factor is set to 5%. Additionally, a lower limit of $\sigma_{\min} = 15$ for $m < 400$ and $\sigma_{\min} = 5$ for $m \geq 400$ is applied. A magnification of the final segmentation demonstrates the ability of region based methods to handle weak edges (3rd row, left). The Figure additionally presents the final piecewise smooth approximation for two different parameters λ (3rd row, sub-figure 2-3). This example also demonstrates the creation of new interfaces and triple junctions.

Figure 12 shows the result of segmenting a colored image from the Caltech image database (Fei-Fei et al., 2004). For this experiment, the chromaticity-brightness color space is used. The weighting parameters for the external forcing terms are set to $\lambda_C = 80$ and $\lambda_B = 20$ such that the chromaticity has a higher influence on the region based segmentation. The ability to handle multiple phases, topological changes (i.e. boundary intersection and creation of triple junctions) and vector-valued image data is demonstrated in this example.

A second real colored image from the Berkeley image database (Arbelaez et al., 2011)) showing two flowers is used to demonstrate the algorithm and to compare the color spaces HSV and CB. Figure 13 and 14 present the results, i.e. the contours and the piecewise constant segmentation at

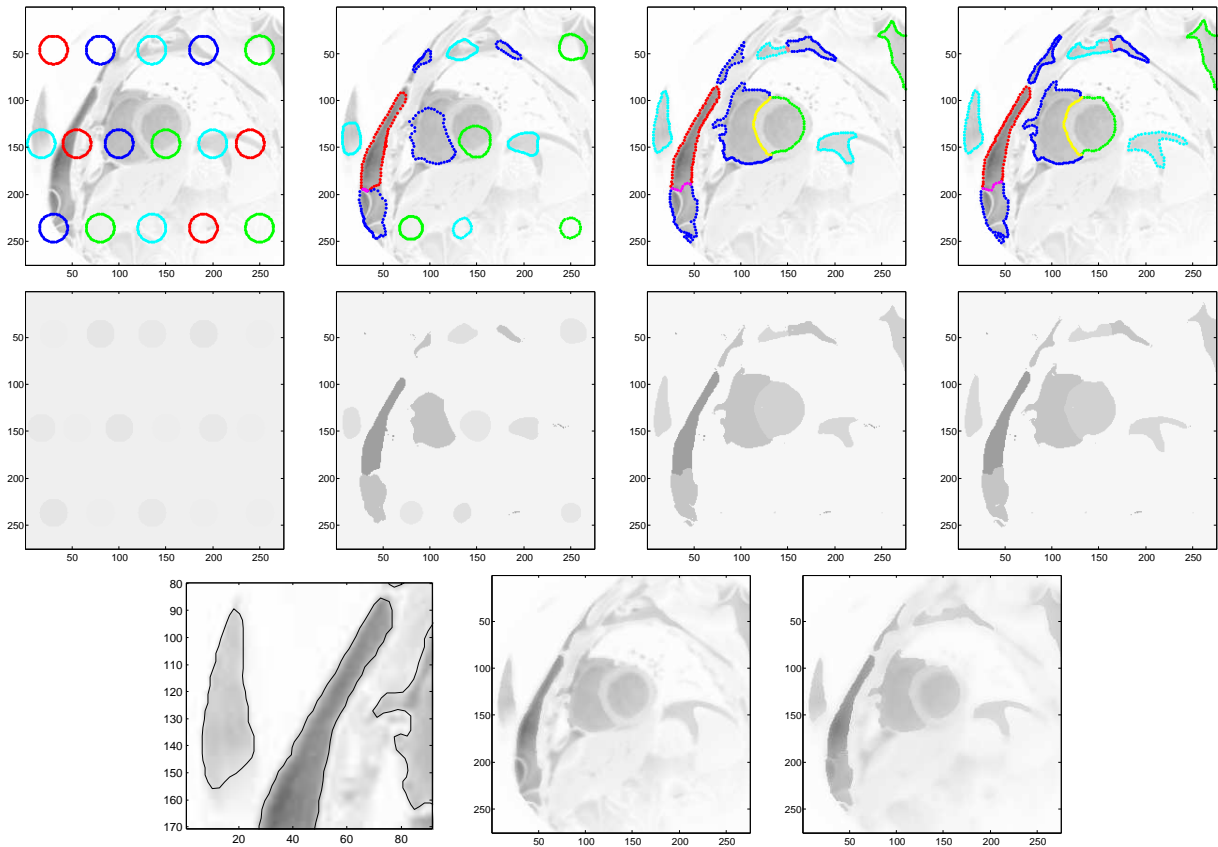


Figure 11: Multi-phase image segmentation of a medical image, $m = 1, 100, 500, 1500$, $\Delta t = 0.02$, $\lambda = 400$, σ -factor 5%, last row: magnification of final segmentation to demonstrate a weak edge, approximation u computed in the post-processing step with $\lambda_k = 1$ and $\lambda_k = 0.1$, $k = 1, \dots, N_R$

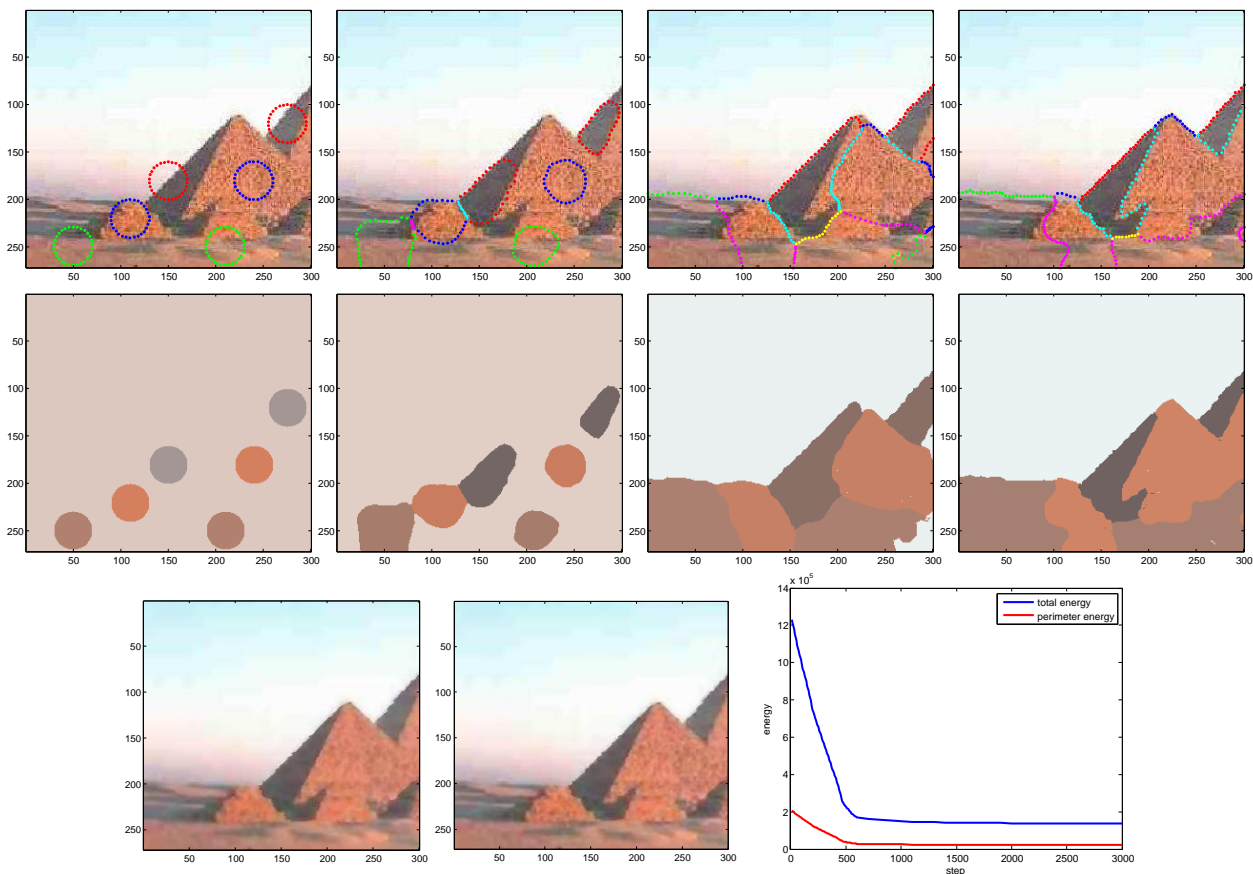


Figure 12: Multi-phase image segmentation of a color image using CB space, image from Caltech database (Fei-Fei et al., 2004), $m = 1, 100, 500, 3000$, $\Delta t = 0.005$, $\lambda_C = 80$, $\lambda_B = 20$, σ -factor 20%, last row: approximation u computed in the post-processing step with $\lambda_k = 0.1$ and $\lambda_k = 1$, $k = 1, \dots, N_R$, and plot of the Mumford Shah energy

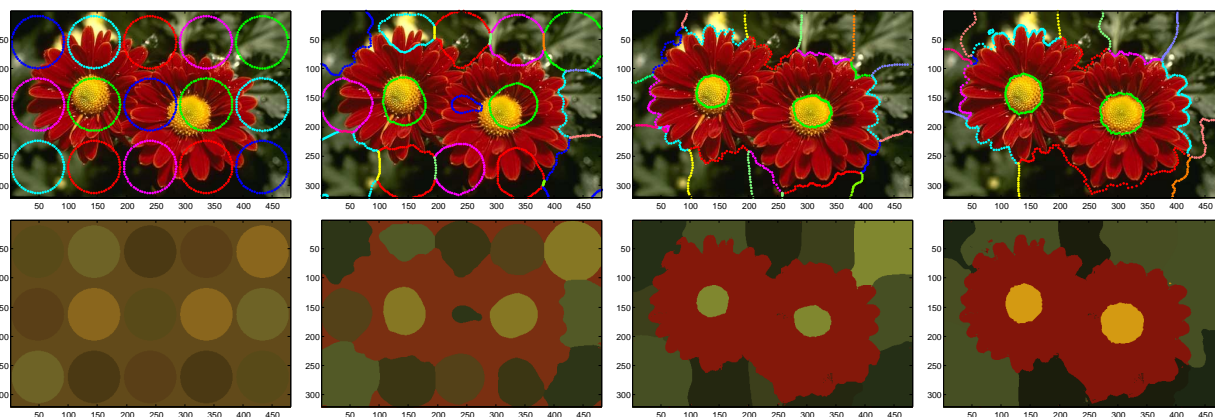


Figure 13: Multi-phase image segmentation of a color image using HSV space, image from Berkeley database (Arbelaez et al., 2011), $m = 1, 25, 100, 1900$, $\Delta t = 0.01$, $\lambda_H = 150$, $\lambda_S = 50$, $\lambda_V = 50$, σ -factor 15%

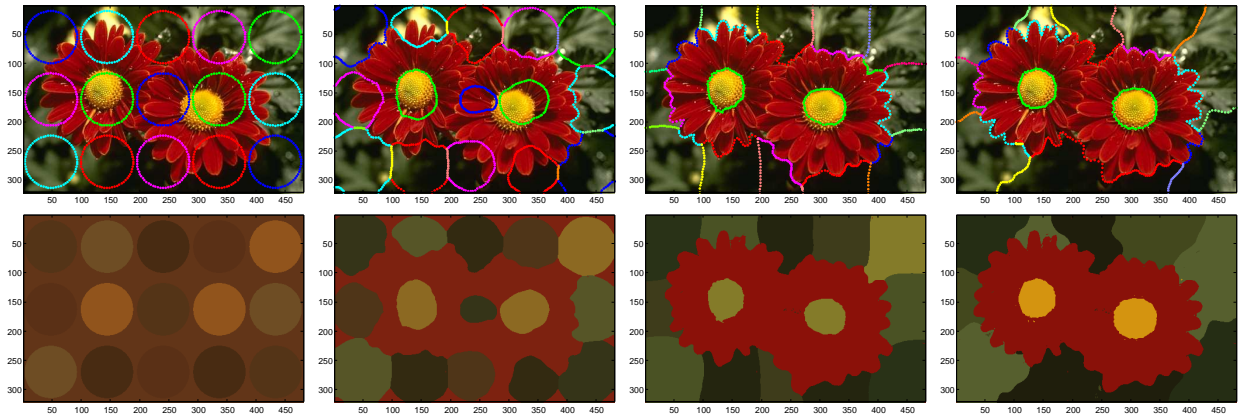


Figure 14: Multi-phase image segmentation of a color image using CB space, image from Berkeley database (Arbelaez et al., 2011), $m = 1, 50, 100, 1900$, $\Delta t = 0.01$, $\lambda_C = 180$, $\lambda_B = 40$, σ -factor 15%

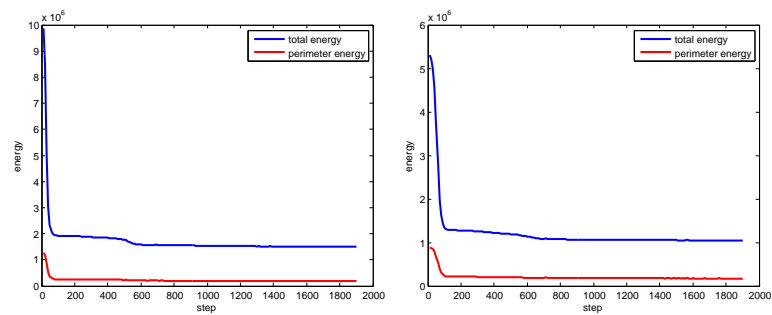


Figure 15: Energy decrease segmenting the flower image from Berkeley database, Left: HSV color space, Right: CB color space



Figure 16: Result of post-processing smoothing of the flower image from Berkeley database using $\lambda_k = 1$ (first row) and $\lambda_k = 10$ (second row), $k = 1, \dots, N_R$, Left: HSV color space, Right: CB color space

time step $m = 1, 25, 100, 1900$. Apart of the weights for external forcing terms, the same parameters are used for the two runs. The segmentation using the CB color space results in a better final segmentation. The interface between red phase and dark green phase near the right flower matches better with the edges of the flower using the CB space. Furthermore, the different green phases of the background with different brightness are segmented more accurately. Figure 15 shows the energy decrease. In each case, the fastest energy decrease can be noted at the beginning, i.e. in the first 200 steps. Then, only small improvements of the segmentation are conducted where the energy decreases only slightly. The result of the post-processing image denoising step u , i.e. the solution of $-\frac{1}{\lambda}\Delta u + u = u_0$ for $\lambda = 1$ and $\lambda = 10$ with Neumann boundary conditions, is presented in Figure 16.

5 Conclusion

We proposed a parametric approach for image segmentation using region based active contours. The evolution equations for image segmentation are based on the functional of Mumford and Shah (1989). The evolution equations consist of a curvature term and an external forcing term designed for image segmentation purposes. Similar as Chan and Vese (2001), we consider piecewise smooth approximations of the original image function. In contrast to Chan and Vese who make use of level-set techniques, the contours are represented by smooth parameterizations in this work. Following the parametric approach of Barrett et al. (2007a), we introduced a framework for multiple curves which can also meet at triple junctions and can intersect with the outer boundary. Further, an elliptic partial differential equation with Neumann boundary conditions is derived from the Mumford-Shah functional for piecewise smooth approximating image functions. Having detected the regions, the image can be denoised by solving the PDE on each phase separately. Due to the Neumann boundary conditions, the region boundaries (edges) are not smoothed out.

A numerical approximation based on finite differences has been presented where the smooth curves are approximated by polygonal lines. The numerical scheme automatically provides good mesh quality. Indeed, a semi-discrete variant of the scheme leads to meshes with an equidistribution property. As parametric methods cannot detect topological changes automatically, we used an efficient scheme to detect topological changes such as splitting and merging of curves and creation of triple junctions and boundary intersection points. The segmentation technique has been successfully applied on several artificial and real images with scalar and vector-valued image data.

Acknowledgement

The authors thank Dr. Declan O'Regan and the Robert Steiner MR Unit, MRC Clinical Sciences Centre, Imperial College London, for providing the heart slice image used in Figure 11.

References

- S. Araki, N. Yokoya, H. Iwasa, and H. Takemura. Splitting of active contour models based on crossing detection for extraction of multiple objects. *Systems and Computers in Japan*, 28(11): 34–42, 1997.
- P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011.

- G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing*. Springer, New York, 2006.
- J.-F. Aujol and S. H. Kang. Color image decomposition and restoration. *Journal of Visual Communication and Image Representation*, 17(4):916–928, 2006.
- M. Balažovjech, K. Mikula, M. Petrasova, and J. Urbán. Lagrangean method with topological changes for numerical modelling of forest fire propagation. *ALGORITMY 2012, 19th Conference on Scientific Computing, Podbanske, Slovakia, September 9-14, 2012, Proceedings of contributed papers and posters*, pages 42–52, 2012.
- J. W. Barrett, H. Garcke, and R. Nürnberg. On the variational approximation of combined second and fourth order geometric evolution equations. *SIAM Journal on Scientific Computing*, 29:1006–1041, 2007a.
- J. W. Barrett, H. Garcke, and R. Nürnberg. A parametric finite element method for fourth order geometric evolution equations. *Journal of Computational Physics*, 222:441–467, 2007b.
- V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.
- T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10:266–277, 2001.
- T. F. Chan, B. Y. Sandberg, and L. A. Vese. Active contours without edges for vector-valued images. *Journal of Visual Communication and Image Representation*, 11:130–141, 2000.
- T. F. Chan, S. H. Kang, and J. Shen. Total variation denoising and enhancement of color images based on the CB and HSV color models. *Journal of Visual Communication and Image Representation*, 12:422–435, 2001.
- D. Cremers, C. Schnörr, and J. Weickert. Diffusion-snakes: Combining statistical shape knowledge and image information in a variational framework. *IEEE Workshop on Variational and Level set Methods*, 2001.
- T.A. Davis. Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 30(2):196–199, 2004.
- K. Deckelnick, G. Dziuk, and C. M. Elliott. Computation of geometric partial differential equations and mean curvature flow. *Acta Numerica*, 14:139–232, 2005.
- G. Doğan, P. Morin, and R. H. Nochetto. A variational shape optimization approach for image segmentation with a Mumford-Shah functional. *SIAM Journal on Scientific Computing*, 30(6):3028–3049, 2008.
- L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *IEEE, CVPR Workshop on Generative-Model Based Vision*, 2004.
- L. He and S. Osher. Solving the Chan-Vese model by a multiphase level set algorithm based on the topological derivative. In: *Scale Space Variational Methods in Computer Vision, Lecture Notes in Computer Science*, pages 777–788, 2007.
- M. Hintermüller and A. Laurain. Multiphase image segmentation and modulation recovery based on shape and topological sensitivity. *Journal of Mathematical Imaging and Vision*, 35:1–22, 2009.
- M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

- S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi Jr. Conformal curvature flows: From phase transitions to active vision. *Archive for Rational Mechanics and Analysis*, 134(3): 275–301, 1996.
- R. Malladi, J. A. Sethian, and Vemuri B. C. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:158–175, 1995.
- K. Mikula and J. Urbán. New fast and stable Lagrangean method for image segmentation. In *Proceedings of the 5th International congress on image and signal processing, CISP 2012*, pages 834–842, 2012.
- K. Mikula and D. Ševčovič. Evolution of curves on a surface driven by the geodesic curvature and external force. *Applicable Analysis*, 85(4):345–362, 2006.
- D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989.
- A. Nakhmani and A. Tannenbaum. Self-crossing detection and location for parametric active contours. *IEEE Transactions on Image Processing*, 21(7):3150–3156, 2012.
- S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.
- R. Ronfard. Region-based strategies for active contour models. *International Journal of Computer Vision*, 13:229–251, 1994.
- V. Srikrishnan, S. Chaudhuri, S.D. Roy, and D. Ševčovič. On stabilisation of parametric active contours. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR '07.*, 2007.
- B. Tang, G. Sapiro, and V. Caselles. Color image enhancement via chromaticity diffusion. *IEEE Transactions on Image Processing*, 10:701–707, 2002.
- A. Tsai, A. Yezzi, and A. S. Willsky. Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation and magnification. *IEEE Transactions on Image Processing*, 10(8):1169–1186, 2001.
- L. A. Vese and T. F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50:271–293, 2002.