A Combined Shape-Newton and Topology Optimization Technique in Real-Time Image Segmentation *

M. Hintermüller†

1 Introduction and Motivation

Recent computational challenges in *image segmentation* are due to image guided surgery, in particular brain surgery. In a preoperative phase high resolution images, *e.g.*, magnet-resonance images (MRIs), of a patient are taken, enhanced (denoised, deblurred,...) and then used for a 3D rendering of the region of interest which includes a segmentation phase for giving a precise account of tumor boundaries. Based on this reconstruction the surgeon plans the operative intervention. In brain surgery, where high precision is required, any intervention changes the local situation, *e.g.*, the tumor location. Therefore, in modern regimes [8, 19, 20] a correction of the segmented image is attempted intraoperatively by taking new MRI scans and re-doing the image processing part. This should ideally reduce the operation time and accurately guide the surgeon. Besides certain modelling aspects, on the computational side this poses several challenges such as the fast and reliable segmentation.

In image segmentation, which is the task of identifying boundary curves (contours) of regions of approximately homogeneous features, there are several paradigm models such as edge detectors, Mumford-Shah model, shape-based approaches, discrete approaches ... which address different aspects in the segmentation. In this

*Partially support by the Austrian Sciences Fund FWF under SFB03 "Optimierung und Kontrolle".

†Rice University, Department of Computational and Applied Mathematics, Houston, Texas, USA.

paper we consider a variational approach based on edge detectors. In this case, a widely used model problem is given by

$$\text{minimize } J(\Gamma) = \int_\Gamma g\, dS + \nu \int_\Omega g\, dx, \tag{1}$$

where $\Gamma \in \mathcal{D}$ denotes the contour and $\mathcal{D}$ a set of shapes, g is a given edge detector defined on the image domain $D \subset \mathbb{R}^2$, $\Omega \subseteq D$ is the set of homogeneous features with boundary Γ, and $\nu \in \mathbb{R}$ acts as a regularization parameter. In Figure 1, for the contrast agent based kidney image (left plot) we provide a standard Gaussian-type edge detector (right plot). In general, at ideal edges we have $g = 0$; otherwise $g > 0$. However, due to rest noise in the image, we can usually only expect $g \approx 0$. Let us mention that more sophisticated edge detectors compared to the one presented here are available.

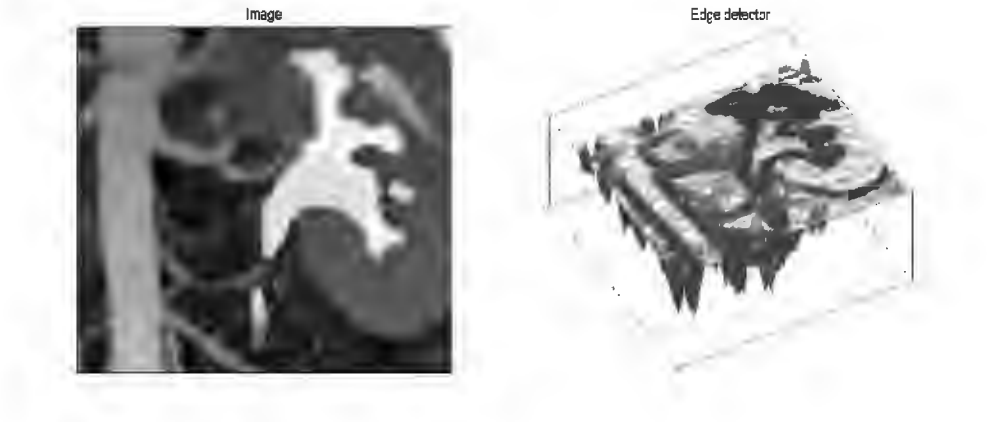

Figure 1. *Image (left) and corresponding edge detector (right).*

Active contour approaches in image segmentation consider Γ as the optimization variable which is adjusted iteratively. Earlier accounts for solving (1) were based on parameterizations of Γ, like in [10]. A particular choice was given by the arc length parameterization; see, *e.g.*, [21]. These techniques, however, have several drawbacks including the need of expensive re-parameterizations in case of topological changes. Borrowing a concept due to Osher and Sethian [13] and with the aim of devising a geometrically intrinsic, *i.e.*, parameter independent approach, Caselles et al. [4] proposed the representation of Γ as the zero-level set of a function ϕ and the approximation of the desired contour by means of an iterative time-marching process. Based on some initial estimate Γ_0, the evolution towards the solution contour is embedded in the propagation of the continuous level-set function $\phi : D \times [0, T] \to \mathbb{R}$ as

$$\Gamma(t) = \{x \in D : \phi(x, t) = 0\}.$$

For this reason ϕ is sometimes called geometrical implicit function, since it implicitly contains the information of interest [12]. Let us assume that the components of $\Gamma(t)$

are closed curves, and let $\Omega(t)$ denote the domain with boundary $\Gamma(t)$. Then, in order to have a well-conditioned representation of $\Gamma(t)$ one usually requires

$$\Omega(t) = \{x \in D : \phi(x,t) < 0\} \quad \text{and} \quad D \setminus \Omega(t) = \{x \in D : \phi(x,t) \geq 0\}. \tag{2}$$

In this case, the outward unit normal $n(x,t)$ to $\Gamma(t)$ at $x \in \Gamma(t)$ is given by

$$n(x,t) = \frac{\nabla\phi(x,t)}{\|\nabla\phi(x,t)\|_2}$$

whenever ϕ is sufficiently smooth. A popular choice satisfying (2) is given by signed distance functions. In this case we have $\|\nabla\phi(x,t)\|_2 = 1$ almost everywhere and, hence, $n(x,t) = \nabla\phi(x,t)$. Since ϕ is usually defined on D, these definitions and properties naturally extend to D.

Now, let us assume that the velocity of the above mentioned evolution in normal direction to $\Gamma(t)$ is given by $F(x(t),t)$ with F a scalar-valued function defined on $\Gamma(t)$. Then it is well-known that the transport of ϕ is governed by the *level-set equation*

$$\phi_t + F_{\text{ext}}\|\nabla\phi\|_2 = 0 \text{ on } D, \tag{3}$$

see [12, 15]. Here F_{ext} denotes some suitable extension of F to D.

The *level-set method* is celebrated for its numerical robustness, flexibility w.r.t. topological changes, wide applicability and many more. In connection with velocity fields F coming from shape sensitivity analysis [5, 18], as for instance F being the negative shape gradient of J or the corresponding Newton direction [9], it is a versatile tool for shape optimization including image segmentation via (1). While the merging or splitting of existing contours can be handled easily by level-set based shape optimization algorithms [2, 9], the creation of new components is in general hard (if not impossible) to accomplish by using classical shape sensitivity concepts only. As a remedy we propose the blending of shape sensitivity information with topological sensitivity. The latter concept goes back to [7, 17] and allows to create holes in a domain, rather than perturbing existing boundaries. For a given function $\mathcal{J} : \mathcal{D} \to \mathbb{R}$, where $\mathcal{D}$ is an appropriate set of shapes, the *topological derivative* (or *topological gradient*) at $x \in D$ is defined as

$$\mathcal{T}(x) := \lim_{\rho \downarrow 0} \frac{\mathcal{J}(D \setminus \overline{B_\rho(x)}) - \mathcal{J}(D)}{|\overline{B_\rho(x)}|}, \tag{4}$$

where $B_\rho(x) := \{y \in \mathbb{R}^n : \|y - x\| < \rho\} \cap D$, $\overline{B_\rho(x)}$ denotes the closure of $B_\rho(x)$, and $|\overline{B_\rho(x)}|$ the volume of $\overline{B_\rho(x)}$. In the case of 2D-image segmentation we have $n = 2$.

In this paper, we study edge-detector based image segmentation algorithms which utilize both shape as well as topological sensitivity information. Our technique is a phase-I–phase-II-type approach, which means that the shape sensitivity phase and the phase using the topological derivative are employed in a serial fashion, *i.e.*, no synchronous blending of information takes place. This is similar to [3]. Typically, the algorithm is started by the topological phase for computing initial

guesses. Then, if necessary, shape sensitivity is used to drive the contour. After this phase stopped, one may again enter the topology optimization phase, and then continue altering the phases until some stopping rule is satisfied.

An additional benefit of the new approach is given by the fact that the segmentation becomes fully automatic as opposed to state-of-the-art techniques which require a (more or less) manual selection of an initial contour.

The rest of the paper is organized as follows: In section 2 we derive a model for reconstructing topological information in a given image. The corresponding minimization problem involves a geometrically regularized least-squares objective function subject to a linear elliptic partial differential equation. Further, the topological gradient for the objective in the model problem is computed. The section finishes with the phase-I algorithm. First and second order shape sensitivity analysis and the phase-II algorithm, which is a shape Newton method, are the subjects of section 3. In the final section 4 we address aspects of an efficient numerical realization and report on results obtained by the new method. It turns out that the topology optimization phase-I can act as a segmentation algorithm in its own right.

Notation: We use the standard notation $L^p(S)$ and $H^1(S)$, $H^1_0(S)$ for Lebesgue and Sobolv spaces, where $S \subset \mathbb{R}^n$ is some measurable set. By $(\cdot,\cdot)_S$ the standard $L^2(S)$ inner-product is denoted. In a slight misuse of notation we also write $(\cdot,\cdot)_S$ for vector-valued arguments. The norm on $H^1_0(S)$ is given by $\|\nabla v\|_{H^1_0(S)}$ for $v \in H^1_0(S)$. Moreover, $\|w\|_{L^2(S)}$, $w \in L^2(S)$, denotes the standard $L^2(S)$-norm; analogously for $\|w\|_{H^1(S)}$. Further, the measure of a set S is denoted by $|S|$.

2 Model and its topological sensitivity

In this section we study the segmentation of an image by means of the topological derivative. Here we assume that images are given in terms of their intensity map I, which is also at the core of computing suitable edge detectors. We start by establishing an appropriate minimization problem and compute the topological derivative of its objective functional afterwards. We conclude this section by specifying the phase-I algorithm for topology optimization.

2.1 A simple model

Let us assume that $u_d \in H^1_0(D)$ is given data related to the intensity map $I : D \to [0,1]$ of a gray-scale image. For simplicity we assume that the image intensities are scaled to [0,1], where features of interest have intensity 1, ideally. Since I might contain some rest noise, one option for computing u_d is given by solving

$$-\Delta u_d = I \text{ in } D, \quad u_d = 0 \text{ on } \partial D, \tag{5}$$

which can be interpreted as the convolution of the intensity map I with a Gaussian kernel. Our aim is now to find a function $u^* \in H^1_0(D)$ which is close to u_d. Then $-\Delta u^*$ is an approximation to the intensity map I. A natural choice realizing this objective is given my

$$\text{minimize } \mathcal{J}_0(u,\Omega) := \frac{\beta_1}{2}\|u-u_d\|^2_{L^2(D)} + \frac{\beta_2}{2}\|u-u_d\|^2_{H^1_0(D)} \tag{6a}$$

$$\text{subject to } -\Delta u = f_\Omega \text{ in } D, \quad u = 0 \text{ on } \partial D, \tag{6b}$$

where $\mathcal{J}_0 : \mathcal{D} \to \mathbb{R}$, and $\beta_1, \beta_2 \in \mathbb{R}_0^+$, with $\beta_1 + \beta_2 > 0$. The right hand side of the governing equation is defined as

$$f_\Omega(x) = \begin{cases} 1 & \text{if } x \in \Omega, \\ 0 & \text{else.} \end{cases}$$

Since the presence of noise in I may adversely affect the minimization of (6), we propose to add a regularization term in form of a volume penalty to the objective function. Then the resulting model for recovering u^* is as follows:

$$\text{minimize } \mathcal{J}_\mu(u, \Omega) := \mathcal{J}_0(u, \Omega) + \mu \int_\Omega 1 dx \tag{7a}$$

$$\text{subject to } -\Delta u = f_\Omega \text{ in } D, \quad u = 0 \text{ on } \partial D, \tag{7b}$$

where $\mu \in \mathbb{R}_0^+$ is fixed.

The state equation (6b) admits a unique solution $u = u(\Omega)$. This fact can be used in problem (7) to derive a corresponding reduced problem:

$$\text{minimize } \mathcal{J}(\Omega) = \mathcal{J}_\mu(u(\Omega), \Omega) \quad \text{over} \quad \Omega \in \mathcal{D}. \tag{8}$$

We consider (8) as our model problem for the topology optimization phase.

Before addressing topological sensitivity issues of $\mathcal{J}(\Omega)$, let us point out that different, more sophisticated approaches and models are possible.

2.2 Topological derivative

Now we compute the topological derivative of $\mathcal{J}$. For this purpose note that we are not only interested in creating holes in Ω but also in adding components in $D \setminus \overline{\Omega}$. This is accomplished by considering for $x \in D \setminus \overline{\Omega}$

$$\mathcal{T}(x) = \lim_{\rho \downarrow 0} \frac{\mathcal{J}(\Omega \cup \overline{B_\rho(x)}) - \mathcal{J}(\Omega)}{|\overline{B_\rho(x)}|}. \tag{9}$$

Notice the difference to (4). In (9) we do not "subtract material", rather we "add material" with the aim of changing the topology in $D \setminus \overline{\Omega}$. Strictly speaking, we have to consider $\mathcal{J}(D \cap (\Omega \cup \overline{B_\rho(x)}))$ and $|D \cap (\Omega \cup \overline{B_\rho(x)})|$ in (9).

Let us first recall a result from [3, Proposition 2.2] which is useful in the proof of our main result, Theorem 2.

Proposition 1. *Let $\Omega \subset D$ be some measurable domain with positive measure, and let $u_{\rho,x} \in H_0^1(D)$ and $u \in H_0^1(D)$ be the solutions of*

$$-\Delta u_{\rho,x} = f_{\Omega \setminus \overline{B_\rho(x)}} \quad \text{and} \quad -\Delta u = f_\Omega,$$

respectively. Then there exists a constant $C > 0$ such that

$$\|u_{\rho,x} - u\|_{H^1(\Omega)} \leq C |B_\rho(x) \cap \Omega|^{1/s} \tag{10}$$

for $s \in (\frac{2n}{n+2}, +\infty]$.

Now we are prepared for proving the main result of this section.

Theorem 2. *Let* $\Omega \subset D$ *be some measurable domain with positive measure. Then the topological derivative of* $\mathcal{J}$ *in (8) is given by*

$$\mathcal{T}(x) = \begin{cases} -p(x) - \mu & \text{for a.a. } x \in \overline{\Omega}, \\ p(x) + \mu & \text{for a.a. } x \in D \setminus \overline{\Omega}, \end{cases} \tag{11}$$

where $p \in H_0^1(\Omega)$ *solves the adjoint problem*

$$-\Delta p = \beta_1(u - u_d) - \beta_2 \Delta(u - u_d) \text{ in } D, \quad p = 0 \text{ on } \partial D. \tag{12}$$

Proof. We consider first $x \in \Omega$ and (4). Then

$$\begin{aligned} \mathcal{J}(\Omega \setminus \overline{B_\rho(x)}) - \mathcal{J}(\Omega) = \frac{\beta_1}{2}\|u_{\rho,x} - u\|^2_{L^2(D)} + \beta_1(u_{\rho,x} - u, u - u_d)_D + \\ \frac{\beta_2}{2}\|u_{\rho,x} - u\|^2_{H_0^1(D)} + \\ \beta_2(\nabla(u_{\rho,x} - u), \nabla(u - u_d))_D - \mu|\overline{B_\rho(x)}| \\ = \frac{\beta_1}{2}\|u_{\rho,x} - u\|^2_{L^2(D)} + (\nabla p, \nabla(u_{\rho,x} - u))_D - \mu|\overline{B_\rho(x)}| \\ = \frac{\beta_1}{2}\|u_{\rho,x} - u\|^2_{L^2(D)} + (f_{\Omega\setminus\overline{B_\rho(x)}} - f_\Omega, p)_D - \mu|\overline{B_\rho(x)}|, \end{aligned}$$

where we used (12) in the next to the last inequality and the governing equation for the last identity. Using Proposition 1, we obtain

$$\lim_{\rho\downarrow 0} \frac{\mathcal{J}(\Omega \setminus \overline{B_\rho(x)}) - \mathcal{J}(\Omega)}{|\overline{B_\rho(x)}|} = -\lim_{\rho\downarrow 0} \frac{1}{|\overline{B_\rho(x)}|} \int_{\overline{B_\rho(x)}\cap D} p(z)\, dz - \mu.$$

From the localization theorem we infer

$$-\lim_{\rho\downarrow 0} \frac{1}{|\overline{B_\rho(x)}|} \int_{\overline{B_\rho(x)}\cap D} p(z)\, dz = -p(x)$$

and thus

$$\lim_{\rho\downarrow 0} \frac{\mathcal{J}(\Omega \setminus \overline{B_\rho(x)}) - \mathcal{J}(\Omega)}{|\overline{B_\rho(x)}|} = -p(x) - \mu.$$

For $x \in D \setminus \overline{\Omega}$ and (9) we obtain in a similar way

$$\mathcal{J}(\Omega \cup \overline{B_\rho(x)}) - \mathcal{J}(\Omega) = \frac{\beta_1}{2}\|u_{\rho,x} - u\|^2_{L^2(D)} + (f_{\Omega\cup\overline{B_\rho(x)}} - f_\Omega, p)_D + \mu|\overline{B_\rho(x)})|$$

and further

$$\begin{aligned} \lim_{\rho\downarrow 0} \frac{\mathcal{J}(\Omega \cup \overline{B_\rho(x)}) - \mathcal{J}(\Omega)}{|\overline{B_\rho(x)})|} &= \lim_{\rho\downarrow 0} \frac{1}{|\overline{B_\rho(x)})|} \int_{\overline{B_\rho(x)})\cap D} p(z)\, dz + \mu \\ &= p(x) + \mu, \end{aligned}$$

which completes the proof. □

According to (11) the topological gradient has a simple structure since–up to an additive constant–it coincides with the adjoint. In general, the computational effort is related to the solution of (12) which can be obtained efficiently by, *e.g.*, multigrid methods or FFT-techniques. In the special case $\beta_1 = 0$ and $\beta_2 > 0$ we have

$$\mathcal{T}(x) = \begin{cases} \beta_1(u_d(x) - u(x)) - \mu & \text{for a.a. } x \in \overline{\Omega}, \\ \beta_1(u(x) - u_d(x)) + \mu & \text{for a.a. } x \in D \setminus \overline{\Omega}, \end{cases}$$

which shows that for computing $\mathcal{T}$ only the state equation has to be solved.

As it is the case for classical gradient techniques, the objective function $\mathcal{J}$ can be decreased if there exists a search (update) direction $\delta^{\mathcal{T}}$ such that for some sufficiently small $\delta_t > 0$

$$\mathcal{J}(\Omega(t + \delta_t)) - \mathcal{J}(\Omega(t)) < 0 \tag{13}$$

where the new domain $\Omega(t + \delta_t)$ is given by

$$\Omega(t + \delta_t) = \{x \in D : \phi(x, t + \delta_t) < 0\}$$

with ϕ satisfying

$$\phi_t(x, t) = \delta^{\mathcal{T}}(x, t). \tag{14}$$

It remains to address how (11)–(12) can be used if the geometry Γ is represented by a geometrical implicit function ϕ and such that (13) is realized. Throughout we assume that ϕ satisfies the sign convention (2). We further point out that the evolution of $\Gamma(t)$, resp. $\Omega(t)$, is related to a time-dependent process via (3) resp. (14). As a consequence we write $\mathcal{T}(x, t)$ for the topological derivative of $\mathcal{J}(\Omega(t))$ at $x \in D$. Following a reasoning similar to the one in [3], we find:

- For $x \in \Omega(t)$, *i.e.*, for $\phi(x, t) < 0$ a reduction of $\mathcal{J}(\Omega(t))(x)$ is expected if $\mathcal{T}(x, t) < 0$ which means that the topology should change, *i.e.*, a hole should be created. Since $\phi(x, t) < 0$, creating a hole can be accomplished by adding a positive quantity to $\phi(x, t)$. Since $\mathcal{T}(x, t) < 0$, the search (update) direction can be chosen as

$$\delta^{\mathcal{T}}(x, t) = -\mathcal{T}(x, t) > 0 \quad \text{for } x \in \Omega(t). \tag{15}$$

- For $x \in D \setminus \overline{\Omega(t)}$, *i.e.*, for $\phi(x, t) > 0$ we again expect a reduction if $\mathcal{T}(x, t) < 0$. Changing the topology can now be achieved we subtracting a positive quantity from ϕ. This suggests

$$\delta^{\mathcal{T}}(x, t) = \mathcal{T}(x, t) < 0 \quad \text{for } x \in D \setminus \overline{\Omega(t)}. \tag{16}$$

From (11), (15), and (16) we infer

$$\delta^{\mathcal{T}}(x, t) = p(x) + \mu \quad \text{for a.a. } x \in D,$$

where p solves the adjoint problem (12) with $u = (-\Delta)^{-1} f_{\Omega(t)} \in H_0^1(D)$. Note that in general we do not expect that $\|\delta^{\mathcal{T}}\|_{H^1(D)} = 0$ at the solution Ω^*, since the solution set Ω^* is invariant w.r.t. certain update directions. In fact, Ω^* remains unchanged if $\delta^{\mathcal{T}} \leq 0$ on Ω^* and $\delta^{\mathcal{T}} \geq 0$ on $D \setminus \overline{\Omega^*}$.

2.3 Phase-I algorithm for topology optimization

In sections 2.1 and 2.2 we gathered all the tools needed for devising the algorithm for phase-I, *i.e.*, the optimization w.r.t. the topology. Before we state the method we briefly discuss a strategy for adjusting the step length δ_t in (13). We propose a technique very similar to classical line search methods [11].

As a descent measure we propose

$$m(t) = -\|\mathfrak{m}(t)\|^2_{L^2(D)}, \tag{17}$$

with

$$\mathfrak{m}(x,t) = \chi_{\Omega(t)}(x,t)(\delta^{\mathcal{T}}(x,t))_+ - \chi_{D\setminus\overline{\Omega(t)}}(x,t)(\delta^{\mathcal{T}}(x,t))_-, \tag{18}$$

where χ_S denotes the characteristic function of a set S, and $(\cdot)_+ = \max(0,\cdot)$, $(\cdot)_- = \min(0,\cdot)$. Then starting with $\delta_t^0 > 0$ and setting $l = 0$ we check whether

$$\mathcal{J}(\Omega(t+\delta_t^l)) - \mathcal{J}(\Omega(t)) \leq \omega\delta_t^l m(t) < 0, \tag{19}$$

where $0 < \omega < 1$ is fixed. If (19) holds true, then we set $\delta_t = \delta_t^l$; otherwise δ_t^l is reduced, *e.g.* by $\delta_t^{l+1} = q\delta_t^l$, with $0 < q < 1$ fixed, and (19) is checked again. For this procedure to work well, the initial δ_t^0 has to be chosen sufficiently large in order to achieve a topological change.

The algorithm is as follows.

Algorithm 2.1 Phase-I algorithm for topology optimization.

(i) Choose an initial contour Γ_0, with associated domain Ω_0, and the corresponding $\phi(x,0)$, $x \in D$; set $t_0 = 0$ and $k = 0$.

(ii) Solve for $u_k \in H_0^1(\Omega)$: $-\Delta u_k = f_{\Omega(t_k)}$.

(iii) Solve the adjoint problem (12) with $u = u_k$, and let p_k denote the solution. Determine $\delta_k^{\mathcal{T}}$ according to (15) and (16).

(iv) Choose $\delta_t^0 > 0$, and compute the first element of the sequence $\{\delta_t^l\}_{l=0}^{\infty} = \{\delta_t^0, q\delta_t^0, q^2\delta_t^0, \ldots\}$ such that (19) is satisfied. Denote this step-size by $\delta_{t,k}$.

(v) Update ϕ according to (14) with time step $\delta_{t,k}$; set $t_{k+1} := t_k + \delta_{t,k}$, $k := k+1$, and go to (ii).

Several remarks concerning the algorithm are in order. To ease the notation we subsequently use $\mathcal{J}_k = \mathcal{J}(\Omega(t_k))$ and $\Omega_k = \Omega(t_k)$.

- Numerically we stop the algorithm if $\delta_{t,k} \leq \epsilon_1$, or if the difference of two successive function values is small: $|\mathcal{J}_{k+1} - \mathcal{J}_k| \leq \epsilon_2$. Here ϵ_1, ϵ_2 are user specified tolerances which typically depend on the mesh size of the discretization.
- In our numerical tests we found that a stopping rule solely based on $m(t)$ is not adequate.

- There is no need for sophisticated initializations. Step (i) merely serves the purpose of computing a reasonably scaled ϕ which contains geometrical information and can be used in the updating process. One may choose ϕ to be a signed distance function initially. However, we point out that different choices satisfying (2) are possible.
- In contrast to level-set based shape optimization methods there is no need to maintain ϕ to be a signed distance function in all iterations.
- In our tests we choose the line search parameters $\delta_t^0 = \max(1, \|\delta^{\mathcal{T}}\|^{-1}_{H^1(D)})$, $q = 0.5$ and $\omega = 0.001$.

3 Shape sensitivity

Now we collect some results concerning the shape sensitivity of the objective functional of (1), and devise a level-set based shape Newton algorithm for solving (1). In connection with Algorithm 2.1 this provides the phase-II method for shape optimization.

3.1 Shape gradient and shape Hessian

Suppose that we are given an open set $\Omega \subset \mathbb{R}^n$ with boundary $\Gamma = \partial\Omega$, and we would like to compute the sensitivity of

$$J(\Gamma) := J_1(\Gamma) + J_2(\Omega) = \int_\Gamma g \, dS + \nu \int_\Omega g \, dx \tag{20}$$

w.r.t. small perturbations of Ω and, hence, Γ. Further we assume that these perturbations are due to a sufficiently smooth vector-field $V : \mathbb{R}^2 \to \mathbb{R}^2$ with compact support in $\mathbb{R}^2$. In a Lagrangian frame, the perturbations are given by

$$X'(\tau) = V(X(\tau)), \quad X(0) = x \in \mathbb{R}^2. \tag{21}$$

From this we define $P_V : \mathbb{R}^2 \times [0, +\infty) \to \mathbb{R}$ with

$$P_V(x, \tau) = X(\tau), \quad \text{and} \quad \Gamma(\tau) = P_V(\Gamma, \tau), \quad \Omega(\tau) = P_V(\Omega, \tau).$$

We have $\Gamma(0) = \Gamma$ and $\Omega(0) = \Omega$. Then the *Eulerian semi-derivative* of J at Γ in direction V is defined by

$$dJ_1(\Gamma; V) = \lim_{\tau \downarrow 0} \frac{J(\Gamma(\tau)) - J(\Gamma)}{\tau}; \tag{22}$$

analogously one defines $dJ_2(\Omega; V)$. If $dJ_1(\Gamma; V)$ and $dJ_2(\Omega; V)$ exist for all $V \in B$, with B a Banach space of feasible velocity fields, and is linear and continuous in V, then J is called *shape differentiable*. By the Zolesio-Hadamard structure theorem [18, Thm. 2.27] we have that

$$dJ_2(\Omega; V) = \langle j_2(\Omega)n, V \rangle_{\mathcal{C}'(\Gamma), \mathcal{C}(\Gamma)},$$

where $\langle \cdot, \cdot \rangle_{\mathcal{C}'(\Gamma),\mathcal{C}(\Gamma)}$ denotes an appropriate duality pairing. Then the quantity $j_2(\Omega)n$ is called the *shape gradient* of J_2. Note that the structure theorem asserts that the shape gradient always admits a representation as a distribution on Γ. Since this is generically true for J_1, the shape gradient of J can be written as

$$j(\Gamma) = j_1(\Gamma) + j_2(\Omega)$$

with $j_1(\Gamma)$ the shape gradient of J_1.

In a similar fashion one defines the *second order Eulerian semi-derivative* of J at Γ in direction $W \in B$:

$$d^2J(\Gamma; V; W) = \lim_{\tau \downarrow 0} \frac{dJ(\Gamma(\tau); V) - dJ(\Gamma; V)}{\tau},$$

where $\Gamma(\tau) = P_W(\Gamma; \tau)$.

In [9] the first and second order Eulerian semi-derivatives of the function in (20) were computed. We only provide the formulas and refer to [9] for details. The first order Eulerian derivative is given by

$$dJ(\Gamma; V) = \int_\Gamma \left(\frac{\partial g}{\partial n} + g(\kappa + \nu) \right) n \cdot V \, dS, \tag{23}$$

where κ denotes the mean curvature of Γ; see [5]. Thus, we identify

$$j(\Gamma) = \frac{\partial g}{\partial n} + g(\kappa + \nu).$$

The second order Eulerian derivative, or shape Hessian, is found to be

$$d^2J(\Gamma; V_F; V_G) = \int_\Gamma \left(\frac{\partial^2 g}{\partial n^2} + (2\kappa + \nu)\frac{\partial g}{\partial n} + \nu\kappa g \right) FG + g(\nabla_\Gamma F \cdot \nabla_\Gamma G) \, dS. \tag{24}$$

Above we used velocity fields $V_F, V_G \in B$ satisfying

$$V_F(x,t) = F_{\text{ext}}(x,t)\frac{\nabla\phi(x,t)}{\|\nabla\phi(x,t)\|_2}, \tag{25}$$

with the properties

$$\nabla F_{\text{ext}}(x,t) \cdot \frac{\nabla\phi(x,t)}{\|\nabla\phi(x,t)\|_2} = 0, \quad F_{\text{ext}}|_\Gamma = F; \tag{26}$$

analogously for V_G. Above, ϕ denotes a geometrical implicit function satisfying (2). Therefore,

$$(V_F \cdot n)_{|\Gamma} = F \quad \text{and} \quad (V_G \cdot n)_{|\Gamma} = G. \tag{27}$$

This choice is motivated by two facts: (i) Analytically it makes non-symmetric terms in the shape Hessian vanish; for details see [9, p.454]. (ii) Extensions of velocities originally defined only on Γ to a neighborhood are frequently chosen to be constant in normal direction; see [1, 9].

Further, $\nabla_\Gamma v$ denotes the tangential gradient of v and is defined by

$$\nabla_\Gamma v_{|\Gamma} = (\nabla v)_{|\Gamma} - \frac{\partial v}{\partial n} n.$$

The term $(\nabla_\Gamma F, \nabla_\Gamma G)_\Gamma$ for all G denotes the weak form of the Laplace-Beltrami operator applied to F which is sometimes written as $-\Delta_\Gamma F$.

3.2 Newton-type flow and descent properties

Using the shape gradient and the shape Hessian at $\Gamma = \Gamma(t)$ we can specify the system satisfied by a *Newton-type velocity field* $N(\Gamma(t))$. Observe that, for instance, for computing the shape sensitivity information required for obtaining the Newton direction at $\Gamma(t)$, in (22) the set $\Gamma(\tau)$ has to be replaced by $\Gamma(\tau + t)$. The Newton equation is given by

$$d^2J(\Gamma(t); V_F; V_{N(\Gamma(t))}) = -dJ(\Gamma(t); V_F) \quad \text{for all } V_F \in B. \tag{28}$$

In general the bilinear form related to d^2J need not be coercive. In our particular example (20) we have that the Laplace-Beltrami operator implies only positive semi-definiteness properties, and the first term in (24) may be negative on subsets of $\Gamma(t)$, which is true especially far away from the optimal shape. Therefore, it is of interest to consider coercive modifications $S(\Gamma(t); V_F, V_G)$ of $d^2J(\Gamma(t); V_F; V_G)$. Here we consider

$$S(\Gamma(t); V_F; V_G) = \int_\Gamma \mathfrak{s}(\Gamma(t))FG + g_{\epsilon_g}(\nabla_\Gamma F \cdot \nabla_\Gamma G)\, dS. \tag{29}$$

with

$$\mathfrak{s}(\Gamma(t))(x) = \max\left(\left(\frac{\partial^2 g}{\partial n^2} + (2\kappa + \nu)\frac{\partial g}{\partial n} + \nu\kappa g\right)(x), \epsilon_{\mathfrak{s}}\right)$$

whenever the first term in the max-expression is smooth enough; otherwise we set $\mathfrak{s}(\Gamma(t)) = \epsilon_{\mathfrak{s}}$ for small $\epsilon_{\mathfrak{s}} > 0$. Notice that we replaced g by g_{ϵ_g} with $g_{\epsilon_g} \geq \epsilon_g$ for small $\epsilon_g > 0$.

As a result there exists $\epsilon > 0$ such that

$$S(\Gamma(t); V_F; V_F) \geq \epsilon \|F\|^2_{H^1(\Gamma)} \text{ for all } V_F \in B. \tag{30}$$

Let $N^S(\Gamma(t))$ denote the solution of

$$S(\Gamma(t); V_F; V_{N^S(\Gamma(t))}) = -dJ(\Gamma(t); V_F) \quad \text{for all } V_F \in B. \tag{31}$$

Further assume that $N^S(\Gamma(t))$ is sufficiently smooth and that $N^S_{\text{ext}}(\Gamma(t))$ is an extension satisfying (26). Further we assume that $V_{N^S(\Gamma(t))} \in B$ satisfies

$$V_{N^S(\Gamma(t))} \cdot \nabla\phi = N^S_{\text{ext}}(\Gamma(t))\|\nabla\phi\|_2.$$

Plugging $V_{N^S(\Gamma(t))}$ in (21) (replacing $V(X(t))$), and applying the chain rule to $\phi(X(t), t) = 0$ (recall that the $\Gamma(t)$ is required to be the zero-level set of the geometrical implicit function ϕ at all times t), we get

$$\begin{aligned} 0 &= (\phi_t(X(\tau), \tau) + \nabla\phi(X(\tau), \tau) \cdot X'(\tau))_{\tau=t} \\ &= \phi_t(x,t) + \|\nabla\phi(x,t)\|_2 \left(\frac{\nabla\phi(x,t)}{\|\nabla\phi(x,t)\|_2}\right) \cdot V_{N^S(\Gamma(t))}(x,t) \\ &= \phi_t(x,t) + N^S(\Gamma(t))_{\text{ext}}(x,t)\|\nabla\phi(x,t)\|_2 \end{aligned} \tag{32}$$

which is the level set equation (3) with $F_{\text{ext}} = N^S_{\text{ext}}$.

Now suppose that $\Gamma(t+\delta_t)$, with $\delta_t > 0$ sufficiently small, is obtained by evolving ϕ via (32). From our shape sensitivity analysis and (31) we infer

$$\begin{aligned} J(\Gamma(t+\delta_t)) &= J(\Gamma(t)) + \delta_t dJ(\Gamma(t); V_{N^S(\Gamma(t))}) + \mathcal{O}(\delta_t) \\ &= J(\Gamma(t)) - \delta_t S(\Gamma(t); V_{N^S(\Gamma(t))}; V_{N^S(\Gamma(t))}) + \mathcal{O}(\delta_t) \\ &\leq J(\Gamma(t)) - \epsilon\delta_t \|N^S(\Gamma(t))\|^2_{H^1(\Gamma)} + \mathcal{O}(\delta_t), \end{aligned}$$

where we used (30) for the last inequality. Thus, if $\delta_t > 0$ is sufficiently small we have

$$J(\Gamma(t+\delta_t)) \leq J(\Gamma(t)) - \underline{\epsilon}\delta_t \|N^S(\Gamma(t))\|^2_{H^1(\Gamma)} \tag{33}$$

for some fixed $\underline{\epsilon}$ satisfying $0 < \underline{\epsilon} < \epsilon$. This proves that the Newton-type flow is a descent flow, *i.e.*, if $\Gamma(t)$ (resp. $\Omega(t)$) is not optimal, then there exists a small positive time step in (32) such that the objective value is sufficiently reduced. We summarize this result in the following proposition.

Proposition 3. *Let $N_S(\Gamma(t))$ be the solution of (31) with extension $N^S_{ext}(\Gamma(t))$ satisfying (25)–(26). Then there exist $\underline{\epsilon} > 0$ and $\delta_t > 0$ such that (33) is satisfied.*

Proposition 3 and (33) are the basis for a strategy for selecting δ_t. In the spirit of classical backtracking line search techniques an initial step length $\delta_t^0 >$ is selected and, if necessary reduced, until (33) is satisfied. This can be done by determining the first element of the sequence $\{\delta_t^l\}_{l=0}^{\infty} = (\delta_t^0, q\delta_t^0, q^2\delta_t^0, \ldots)$, with $0 < q < 1$, which satisfies (33).

3.3 Phase-II algorithm for shape optimization

Now we specify the shape-Newton algorithm for solving (1). Here the method is presented on the continuous level. Details for its discrete counterpart are subject of section 4. For ease of notation, below subscript k refers to quantities at time t_k.

Algorithm 3.1 Phase-II algorithm for shape optimization.

(i) Choose an initial contour Γ_0 with associated domain Ω_0. Compute the corresponding geometrical implicit function ϕ_0 which satisfies the sign convention (2); set $t_0 := 0$ and $k := 0$.

(ii) Solve the Newton-type equation

$$S_k(\Gamma_k; V_F; V_{N_k}) = -dJ(\Gamma_k; V_F) \quad \text{for all } V_F \in B.$$

(iii) Extend N_k to a neighborhood of Γ_k to obtain $N_{k,\text{ext}}$.

(iv) Compute ϕ_{k+1} by using (32) with velocity $N_{k,\text{ext}}$ and a time step $\delta_{t,k} > 0$ such that (33) is satisfied.

(v) If necessary reinitialize ϕ_{k+1}; set $t_{k+1} := t_k + \delta_{t,k}$ and $k := k+1$. Go to (ii).

A few words on details and the realization of the steps of the algorithm are in order.

- Compared to our phase-I Algorithm 2.1 the initialization of Γ is more delicate. In fact, depending on the sign of the parameter ν, see (1), the initial contour has to be a closed curve which is either completely contained in the region of interest ($\nu < 0$), or the region of interest is a subset of Ω_0 ($\nu > 0$). A poor initial choice results in a segmentation failure.
- Choosing ϕ_k to be a signed distance function has several implications: - Flatness in the implicit function is avoided, which allows a numerically stable identification of $\Omega(t)$. - Geometric information can be easily computed; *e.g.*, $n(\Gamma(t),t) = \nabla\phi(\Gamma(t),t)$, $\kappa(\Gamma(t),t) = \operatorname{div}(n(\Gamma(t),t)) = \Delta\phi(\Gamma(t),t)$. - If the aforementioned relations are used to simplify geometrical expression, then it is essential that ϕ remains a signed distance function for all times t; otherwise the geometrical terms are inaccurate. - Numerically, the latter aspect requires a reinitialization, *i.e.*, if ϕ deviates sufficiently from being a signed distance function, then it has to be "reshaped" to a signed distance function.
- There are several options for the reinitialization process; see [12, 15]. We focus on solving the *Eikonal* equation

$$\|\nabla\phi\| = 1 \text{ in } D \text{ with } \phi_{|\Gamma_{k+1}} = \phi_{k+1|\Gamma_{k+1}} = 0. \tag{34}$$

  The solution of (34) is called ϕ_{k+1} again.
- We propose the following extension:

$$\nabla N_{k,\text{ext}} \cdot n_k = 0 \quad \text{and} \quad (N_{k,\text{ext}})_{|\Gamma_k} = N_k.$$

  Then $V_{N_k} = N_{k,\text{ext}} n_k$ satisfies (27).

4 Numerical realization and results

In this section we discuss several aspects of the implementation of our phase-I–phase-II algorithm. Especially phase-II requires some care in its numerical realization. This is necessitated by the discretization of the shape gradient and shape Hessian. We close the section by a report on extensive test runs.

4.1 Numerical realization

The Laplacian operators in the state equation (7b) and the adjoint equation (12) are discretized by the standard five-point stencil. The solution of these Poission problems is achieved by a multigrid technique with a Gauss-Seidel-type smoother.

Phase-I. Compared to the phase-II method, Algorithm 2.1 is rather easy to implement.

- We initialize the method by some easy to generate shape Γ_0 like a circle, ellipse, or $\{x \in D : \|x - x_m\|_q^q = r\}$, $1 \le q < +\infty$, $r > 0$, and $x_m \in \mathbb{R}^2$.

In our experience the method behaves robustly w.r.t. the initialization of the geometry.

- As described earlier, we stop the method if $\delta_{t,k}$ or $|\mathcal{J}_{k+1} - \mathcal{J}_k|$ are smaller than some prescribed tolerances $\epsilon_1, \epsilon_2 > 0$. In our test runs we typically choose $\epsilon_1 = \sqrt{\epsilon_M}$, with ϵ_M the machine precision, and $\epsilon_2 = (\tau^0/(n_{x_1} n_{x_2}))\mathcal{J}_0$, where $\tau^0 > 0$ (typically $\tau^0 = 0.01$ in our tests) and n_{x_1}, n_{x_2} the numbers of pixels in x_1- and x_2-direction.

Phase-II. The challenges in the discretization of Algorithm 3.1 for shape optimization are primarily related to the shape gradient and the shape Hessian. In level-set based front propagation one typically avoids the resolution of the interface Γ_k. While in many cases this may lead to satisfactory evolution (see [12] and the references therein), in the case of optimization techniques one encounters the following troubles: An inaccurate shape Hessian approximation, as long as it induces a coercive bilinear form, may yield no improved convergence speed compared to a pure shape gradient flow. This is in contrast to one of the aims for employing higher order derivatives in optimization. Inaccurate shape gradients typically cause the algorithm to get stuck away from the (discrete) solution. Then the distance to the optimal solution depends on the condition of the problem and the size of the error.

One way to overcome these difficulties is to resolve the interface which allows to compute more accurate approximations to normals, curvatures, function values, and other geometry dependent information and, thus, yields rather accurate shape gradients and shape Hessians. Below we provide some details of the numerical realization of Algorithm 3.1.

- The level set equation (3) is discretized on a fixed Cartesian grid. Since it is a PDE of Hamilton-Jacobi-type, one has to be careful in selecting spatial and time discretizations. With respect to time we use a first order explicit Euler scheme (in our tests it appears that spatial accuracy is more important than accuracy in time). For the spatial discretization we use a second order essentially non-oscillatory scheme (ENO-scheme); see [12, 15, 16].
- The interface is resolved in the following way. In a first sweep we detect interior and exterior interface neighbors among the grid points. An interior neighbor is considered to be in the discrete analogue of Ω_k, and an exterior neighbor is an element of the discrete version of $D \setminus \overline{\Omega_k}$; see Figure 2 where the black nodes correspond to interior and the the white nodes to exterior interface neighbors. Based on this information, then for every cell with an interface transition a local bilinear interpolation model is established. The function values of the model are given by the signed distance values at the grid points, *i.e.*, the corners of the cell. Then discrete approximations of the intersection points (crosses in Figure 2) are given by the roots along grid lines of the interpolation polynomial.
- We choose a locally linear approximation of the interface. For instance, in the case of the exemplary cell in Figure 2, the portion of an interface in a cell is approximated by connecting the interface points. From this we can easily

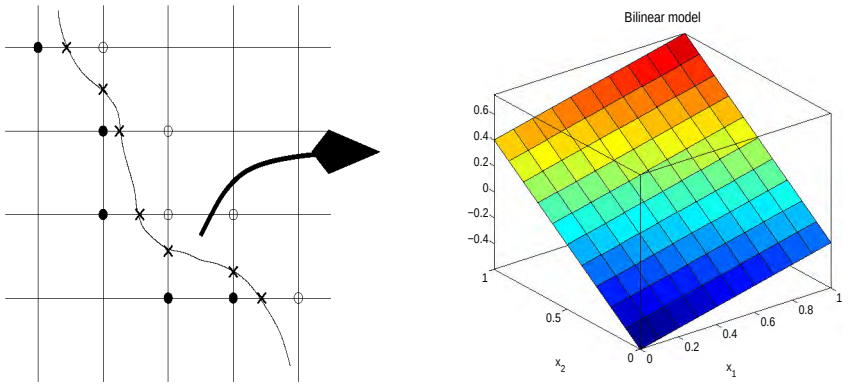

Figure 2. *Left plot: Interior (black circles), exterior (white circles) interface neighbors, and intersection points (crosses). Right plot: Bilinear interpolation model.*

compute the outward normal direction of the interface model. Since we are interested in the outward unit normal at an intersection point, which is not unique due to the kinky structure of the piecewise linear model, we use a linear combination of the normals of neighboring linear pieces. The weights are given by the lengthes of the interface in the respective cell. Normal derivatives and curvature values are computed as follows: Standard finite difference models are used for computing the gradient of a function on the fixed grid points. At the intersection point the gradient of a function is computed by weighted averages. Then the normal derivative is obtain as the inner product of the gradient times the normal. The curvature at regular grid points is computed by discretizing $\Delta\phi$. At intersection points again weighted averages are used.

- For the discretization of the Laplace-Beltrami operator we refer to [6, 9]. In the case of a 1D interface in form of a closed curve, it is closely related to a non-uniform discretization of the Laplacian operator on an interval with periodic boundary conditions.
- The (re)initialization of ϕ is based on (34). On the numerical level we employ a fast marching technique for solving the *Eikonal* equation; see [14] for details.
- In order to reduce the computational effort a narrow band approach is used, *i.e.*, ϕ and all related quantities and equations are only considered on a band around the actual interface (contour).

4.2 Numerical results

In this section we report on numerical results obtained from applying our algorithm. Phase-I is initialized by $\phi_0 = (x_1 - 0.5)^4 + (x_2 - 0.5)^2 - 0.06$. Our initialization of phase-II uses the contour obtained from applying phase-I and computes the corresponding signed distance function. The intensity map I is obtained by a simple

thresholding technique. In fact, let $\tilde{I} \in [0,1]$ denote the original image intensity map. Then, for a grid point x_i we set

$$I(x_i) = \begin{cases} 1 & \text{if } \tilde{I}(x_i) \geq \theta, \\ 0 & \text{if } \tilde{I}(x_i) < \theta. \end{cases} \tag{35}$$

with $\theta > 0$ a user-defined threshold. An alternative is to use $I = \tilde{I}$, but, of course, more sophisticated choices are possible.

Phase-I as an initialization procedure

We first focus on results where phase-I is used as an (re)initialization procedure for phase-II. To this end note that the second integral in (1), depending on the sign of ν, either minimizes or maximizes the area of Ω. If $\nu > 0$, then Ω is intended to be small; otherwise, if $\nu < 0$, then Ω should be large. Thus, depending on the sign of ν the initialization of phase-II is either a contour within the feature of interest ($\nu < 0$), or Ω_0 contains the feature of interest ($\nu > 0$). Here we focus on the first case. In this situation phase-I should provide a geometry which is contained in the region of interest. Then phase-II should drive the contour to its optimal location w.r.t. (1). This requires $\nu > 0$, and $\mu > 0$ relatively large. Note that if μ is large, then $|\Omega|$ is supposed to be small.

In Figure 3 we show the result of phase-I (right plot) when applied to the image in the left plot. For this run the parameters had values $\beta_1 = 1$, $\beta_2 = 1.0E4$, $\mu = 50$, $\theta = 0.45$ and the algorithm terminated after 9 iterations. The initialization of the contour is displayed in the right plot of Figure 7. The corresponding solutions

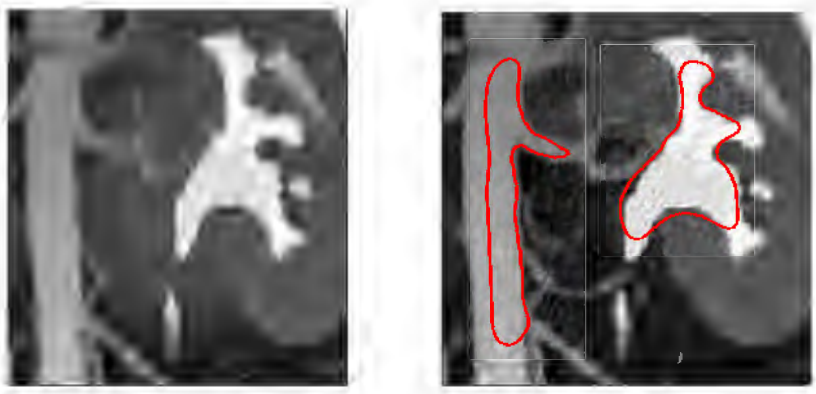

Figure 3. *Image (left) and the result obtained form phase-I (right);* $\beta_1 = 1$, $\beta_2 = 1.0E4$, $\mu = 50$, $\theta = 0.45$.

u of (7b) and p of (12) are depicted in the left and right plot of Figure 4, respectively. One can see that the contour is far off the desired contours at places and stays within the region of interest. Consequently, the set Ω_*^h, the numerical approximation of

the solution to (1), upon termination of phase-I is smaller than the optimal set. In Figure 5 we present the result for a smaller parameter $\mu = 25$ as compared to the previous run. Now the algorithm took 14 iterations. Due to the reduction in μ the area of Ω_*^h is larger. Still, Ω_*^h is smaller than the optimal solution. However, the contour in Figure 5 is an excellent initial guess for phase-II with $\nu < 0$.

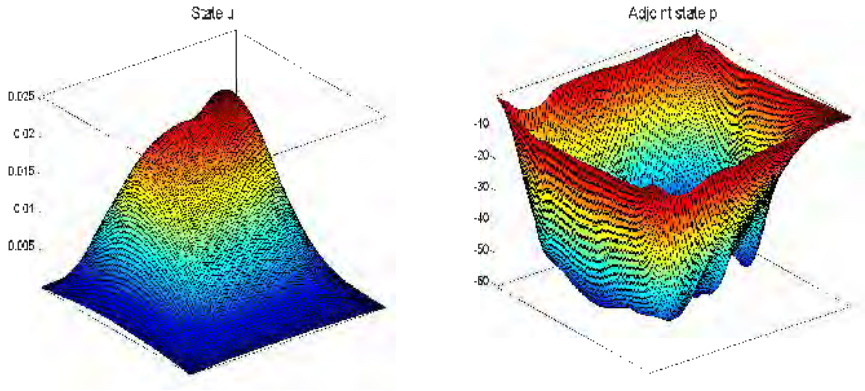

Figure 4. *State u (left) and adjoint state (right) for phase-I; $\beta_1 = 1$, $\beta_2 = 1.0E4$, $\mu = 50$, $\theta = 0.45$.*

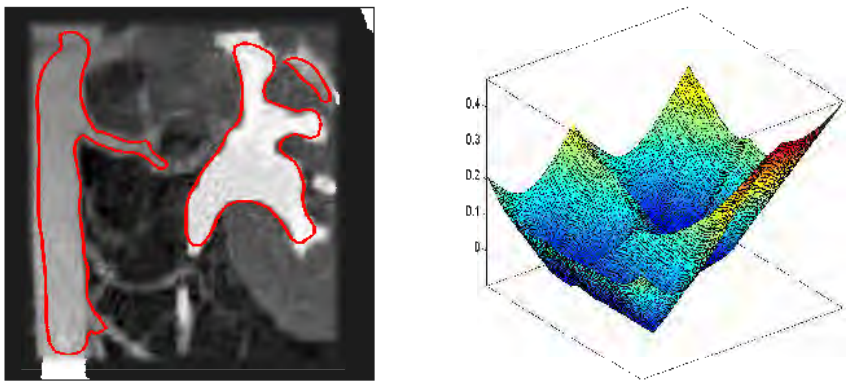

Figure 5. *Segmentation result for phase-I and the corresponding signed distance function; $\beta_1 = 1$, $\beta_2 = 1.0E4$, $\mu = 25$, $\theta = 0.45$.*

If we use the result of Figure 5 as the initial guess for phase-II with $\nu = 1.25$ (ϕ_0 for phase-II is depicted in the right plot of Figure 5), then after 14 phase-II iterations the segmentation result shown in Figure 6 is obtained. Observe that there is no change of the initial topology. This reflects the general fact that a level-set based method using shape sensitivity, except for merging and splitting existing

Figure 6. *Segmentation result obtained by phase-II with initialization by phase-I;* $\beta_1 = 1$, $\beta_2 = 1.0E4$, $\mu = 25$, $\theta = 0.45$.

components, is unable to create new topological information. On the other hand, phase-I provides an excellent automatically produced initial guess for phase-II.

Phase-I as a segmentation algorithm

In our numerical tests we found that our new phase-I algorithm for topology optimization has the potential to serve as a segmentation algorithm even without applying phase-II. Then, of course, μ has to be adjusted adequately. But let us mention that in our test runs no "sophisticated" selection strategies appeared to be necessary.

In Figure 7 (left plot) we show the segmentation result for the contrast agent based kidney image in Figure 3 (left plot) together with the initial contour (right plot in Figure 7). We chose the parameters $\beta_1 = 1$, $\beta_2 = 1.0E4$, $\mu = 1.0E-2$, and $\theta = 0.45$. The algorithm required 34 iterations. Compared to segmentation runs using the phase-II method with a simple manual initial choice (> 60 iterations) and level-set based methods using the shape gradient as a descent flow (> 100 iterations), the iteration count is in favor of our phase-II method. The optimal state and optimal adjoint state for the run documented in Figure 7 are shown in Figure 8.

In Figure 9 we demonstrate that the method is able to detect rather complicated topologies. The initial guess for our topology optimization algorithm is similar to the one shown in Figure 7. We ran phase-I with $\beta_1 = 1$, $\beta_2 = 1.0E4$, $\mu = 1.0E-2$, and $\theta = 0.55$. The algorithm stopped after 78 iterations. The difference between u_*^h, the state upon termination, and u_d^h in the pointwise ℓ_∞-norm is shown in the left plot of Figure 10. The right plot depicts the behavior of $\{J^h(\Omega_k^h)\}$, the sequence of discrete objective values. Note that the scale on the vertical axis is a logarithmic one. Due to our line search procedure $\{J^h(\Omega_k^h)\}$ is monotonically decreasing.

Our final example is shown in Figure 11. The left plot is an MRI scan of a

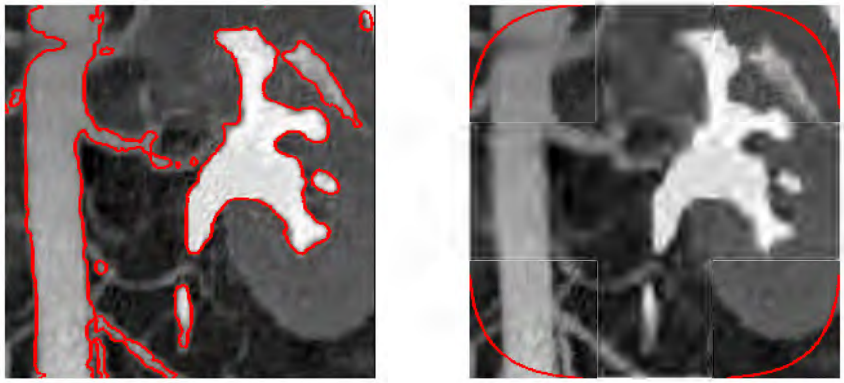

Figure 7. *Segmentation result (left) obtained by phase-I only and initial contour (right);* $\beta_1 = 1$, $\beta_2 = 1.0E4$, $\mu = 1.0E-2$, $\theta = 0.45$.

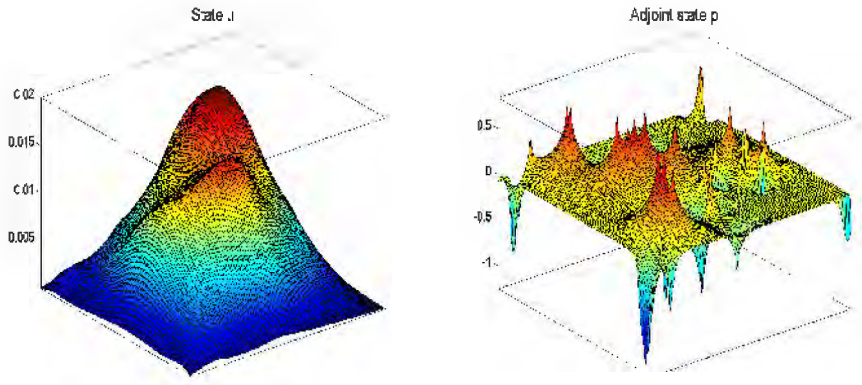

Figure 8. *State u (left) and adjoint state (right) for phase-I;* $\beta_1 = 1$, $\beta_2 = 1.0E4$, $\mu = 1.0E-2$, $\theta = 0.45$.

male chest. The image contains many small components as well as comparatively large ones. Our simple initial guess is depicted in the right plot of Figure 11. It is completely different from the segmentation result shown in the left plot of Figure 12. The parameter values were the same ones as for the results in Figure 9. The algorithm stopped after 53 iterations. The algorithm stopped after 53 iterations. In the right plot of Figure 12 we show the zero-level set without the image in the background. As can be seen, our phase-I algorithm combined with our thresholding strategy allows to detect complex topologies with components of quite different sizes. In Figure 13 we show the signed distance function whose zero-level set is displayed in the right plot of Figure 12. Also, we show the convergence behavior of $\{J^h(\Omega_k^h)\}$. Again, we can see that $\{J^h(\Omega_k^h)\}$ is monotonically decreasing due to

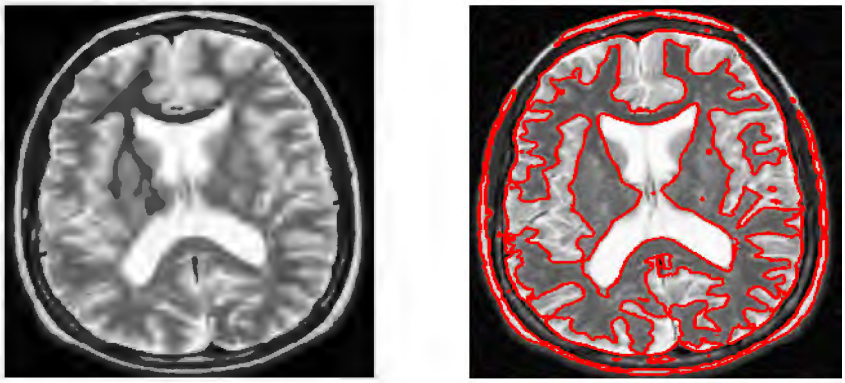

Figure 9. *Image and segmented image using phase-I only;* $\beta_1 = 1$, $\beta_2 = 1.0E4$, $\mu = 1.0E-2$, $\theta = 0.55$.

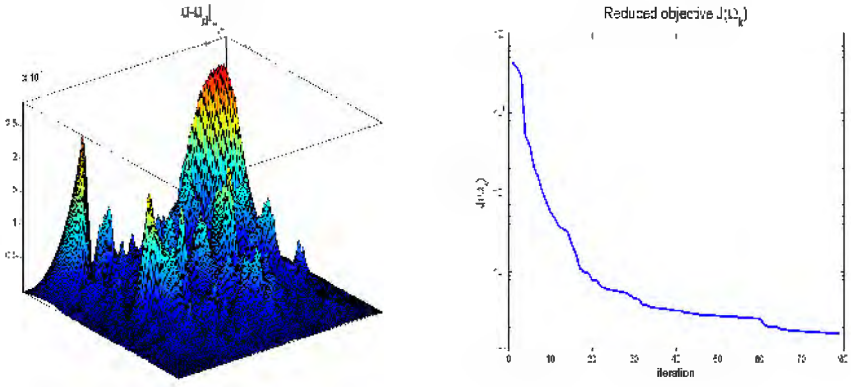

Figure 10. *Difference* $|u_*^h - u_d^h|_{\ell_\infty}$ *and behavior of* $\{J^h(\Omega_k^h)\}$; $\beta_1 = 1$, $\beta_2 = 1.0E4$, $\mu = 1.0E-2$, $\theta = 0.55$.

our time-step size selection strategy.

5 Conclusion

We have presented a phase-I–phase-II concept for real time image segmentation. Phase-I is based on topological sensitivity information in a given image, and phase-II uses shape sensitivity for computing descent flows for a shape functional. Both phases rely on geometrical implicit functions for representing the geometry and utilize line search techniques for computing adequate time step sizes for updating the geometry. In addition, phase-II utilizes a level-set framework for updating the geometry.

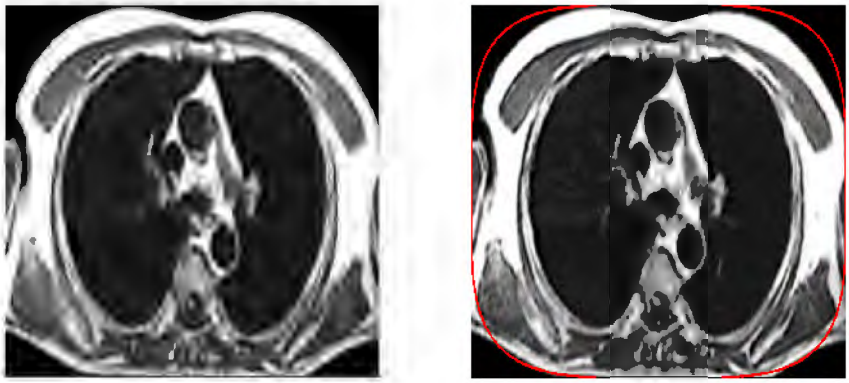

Figure 11. *Image (left) and initial guess (right) for phase-I;* $\beta_1 = 1$, $\beta_2 = 1.0E4$, $\mu = 1.0E-2$, $\theta = 0.55$.

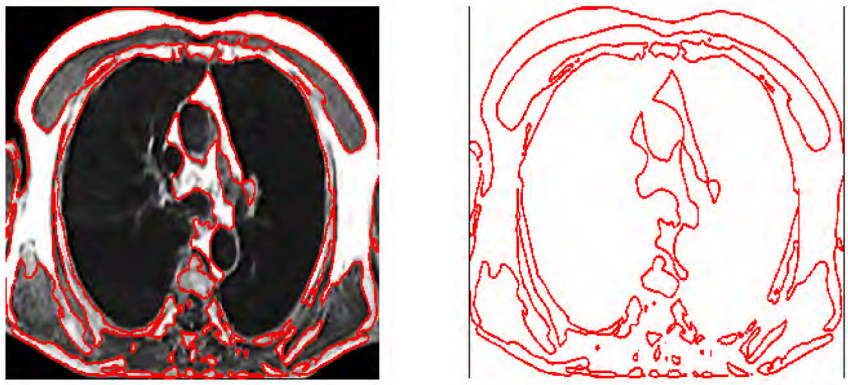

Figure 12. *Segmented image (left) and zero-level set without image background (right);* $\beta_1 = 1$, $\beta_2 = 1.0E4$, $\mu = 1.0E-2$, $\theta = 0.55$.

From our numerical tests we find that phase-I can be used as an initialization method for phase-II and, thus, allows to dispense with manual initialization techniques. This feature is attractive in intraoperative imaging where the amount of required user interaction with the algorithm should be kept as small as possible. Another important consequence of our investigation is related to the fact that phase-I may serve as a segmentation method in its own right, rather than being an initialization method only. In our experience it provides excellent segmentation results. The latter efficiency is based on an preprocessing step for the intensity map. We use a simple thresholding technique which usually follows an image enhancement phase. However, it appears that in difficult situations where tumorous tissue and surrounding tissue are hard to distinguish further investigations for preprocessing

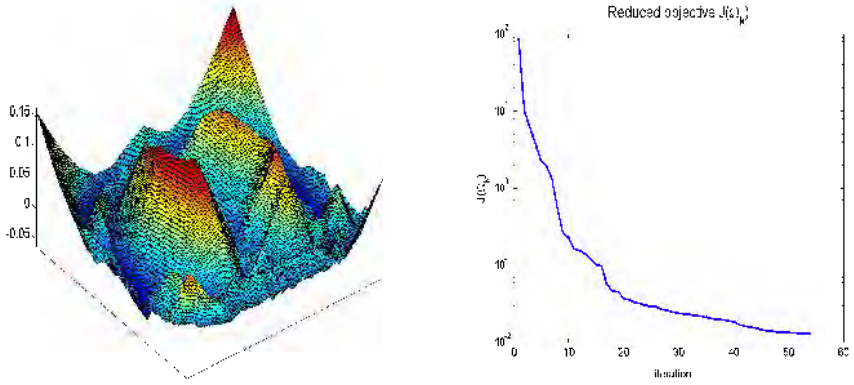

Figure 13. *Signed distance function at the solution (left) and behavior of* $\{J^h(\Omega_k^h)\}$ *(right);* $\beta_1 = 1$, $\beta_2 = 1.0E4$, $\mu = 1.0E-2$, $\theta = 0.55$.

the intensity map are necessary. Also more sophisticated techniques than convolution with a Gaussian-type kernel for obtaining a model problem for the topology optimization phase should be the subject of future investigations.

Bibliography

[1] D. Adalsteinsson and J. A. Sethian. The fast construction of extension velocities in level set methods. *J. Comput. Phys.*, 148(1):2–22, 1999.

[2] M. Burger. Levenberg-marquardt level set methods for inverse problems. *Inverse Problems*, 20:259–282, 2004.

[3] M. Burger, B. Hackl, and W. Ring. Incorporating topological derivatives into level set methods. *J. Comput. Phys.*, 194:344–362, 2004.

[4] V. Caselles, F. Catté, T. Coll, and F. Dibos. A geometric model for active contours in image processing. *Numer. Math.*, 66(1):1–31, 1993.

[5] M. C. Delfour and J.-P. Zolésio. *Shapes and geometries*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001. Analysis, differential calculus, and optimization.

[6] G. Dziuk. Finite elements for the Beltrami operator on arbitrary surfaces. In *Partial differential equations and calculus of variations*, volume 1357 of *Lecture Notes in Math.*, pages 142–155. Springer, Berlin, 1988.

[7] S. Garreau, P. Guillaume, and M. Masmoudi. The topological asymptotic for pde systems: the elasticity case. *SIAM J. Control Optim.*, 39:1756–1778, 2001.

[8] D. T. Gering et al. An integrated visualization system for surgical planning and guidance using image fusion and an open MR. *J. Magnetic Resonance Imag.*, 13:967–975, 2001.

[9] M. Hintermüller and W. Ring. A second order shape optimization approach for image segmentation. *SIAM J. Appl. Math.*, 64(2):442–467, 2003.

[10] M. Kass, A. Witkin, and D. Terzopoulos. Snakes; active contour models. *Int. J. of Computer Vision*, 1:321–331, 1987.

[11] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999.

[12] S. J. Osher and R. P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, New York, 2002.

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

[14] J. A. Sethian. Fast marching methods. *SIAM Rev.*, 41(2):199–235 (electronic), 1999.

[15] J. A. Sethian. *Level set methods and fast marching methods*. Cambridge University Press, Cambridge, second edition, 1999. Evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science.

[16] C-W. Shu and S. J. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.*, 77:439–471, 1988.

[17] J. Sokołowski and A. Żochowski. On the topological derivative in shape optimization. *SIAM J. Control Optim.*, 37(4):1251–1272, 1999.

[18] J. Sokołowski and J-P. Zolésio. *Introduction to shape optimization*. Springer-Verlag, Berlin, 1992. Shape sensitivity analysis.

[19] Y. R. Tsai et al. A shape-based approach to the segmentation of medical imagery using level sets. *IEEE Trans. Med. Imag.*, 22(2):137–154, 2003.

[20] S. Warfield, F. Jolesz, and R. Kikinis. . http://splweb.bwh.harvard.edu:8000/pages/papers/warfield/sc98/. 2004.

[21] A. Yezzi, S. Kichenassamy, A. Kumar, P. Olver, and A. Tannenbaum. A geometric snake model for segmentation of medical imagery. *IEEE Trans. Med. Imaging*, 16(3):199–209, 1997.