

Multiphase image segmentation and modulation recovery based on shape and topological sensitivity

M. Hintermüller* and A. Laurain†

Abstract

Topological sensitivity analysis is performed for the piecewise constant Mumford-Shah functional. Topological and shape derivatives are combined in order to derive an algorithm for image segmentation with fully automatized initialization. Segmentation of 2D and 3D data is presented. Further, a generalized Mumford-Shah functional is proposed and numerically investigated for the segmentation of images modulated due to, e.g., coil sensitivities.

Keywords. Image processing, k -means clustering, modulation recovery, Mumford-Shah functional, piecewise constant reconstruction, segmentation, shape and topological sensitivity.

1 Introduction

Mathematical image segmentation is concerned with the task of partitioning a given image into disjoint (homogeneous) regions [13]. Among the many available paradigms, the approach due to Mumford and Shah [14] turned out to be particularly useful, as it simultaneously denoises and segments the image. It consists in minimizing the functional

$$J_\nu(u, \Gamma) = \int_{\Omega} (f - u)^2 + \mu \int_{\Omega \setminus \Gamma} |\nabla u|^2 + \nu \mathcal{H}^1(\Gamma), \quad (1)$$

where $f : \Omega \mapsto [0, 1]$ denotes given image data (intensity map) on the image domain $\Omega \subset \mathbb{R}^d$ and $\mathcal{H}^1(\cdot)$ is the one-dimensional Hausdorff-measure. The function u represents the reconstructed image and Γ denotes the reconstructed contours (edge set) of the image. The parameters μ and ν penalize deviations from homogeneity on the pieces and the length of the perimeter of the pieces, respectively.

*Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany, and University of Graz, Department of Mathematics and Scientific Computing, Graz, Austria.

†University of Graz, Department of Mathematics and Scientific Computing, Graz, Austria.

While the original approach due to Mumford and Shah admits some variation of the reconstructed image on the disjoint pieces, in [1, 21] Chan and Vese considered a limitation to piecewise constant functions (Chan-Vese model). In this case, the original Mumford-Shah problem reduces to

$$\min_{u, \Gamma} J_\nu(u, \Gamma) = \int_{\Omega} (f - u)^2 + \nu \mathcal{H}^1(\Gamma). \quad (2)$$

Moreover, for given $m \in \mathbb{N}$, u is defined as

$$u = \sum_{i=1}^m c_i \chi_{\Omega_i}, \quad (3)$$

with $c_i \geq 0$ the "gray-value" (or phase) on the segment Ω_i . Here χ_{Ω_i} denotes the characteristic function of Ω_i . Hence, for the minimization of J_ν , besides the determination of c_i , the optimal location of Ω_i becomes an issue. The latter corresponds to finding a topological distribution of the Ω_i in Ω . For this purpose, in [7] a discrete topological sensitivity technique is considered for solving the piecewise constant Mumford-Shah problem.

In the present paper, we first introduce topological sensitivity analysis in the continuous setting (where the original Mumford-Shah problem is posed as well). Compared to earlier methods, this approach has several benefits such as an automatized initialization which ultimately result in a highly efficient (with respect to iteration counts and CPU-time consumption) algorithm for image segmentation; see also section 4.2 for a numerical comparison.

In a second part of the paper (see section 5) we are interested in minimizing the following, more general version of (1):

$$J_\nu(u, \Gamma, \sigma) = \int_{\Omega} (f - \sigma u)^2 + \delta \int_{\Omega} |\nabla^p \sigma|^2 + \mu \int_{\Omega \setminus \Gamma} |\nabla u|^2 + \nu \mathcal{H}^1(\Gamma), \quad (4)$$

where σ , which is assumed to be unknown, models a potential modulation of the image due to, e.g., coil sensitivities during image acquisition; see figure 10 on page 35 for an example, where the image on the right in the first row is a modulated version of the one on the left. Typically, σ is a smooth function which we do assume to be independent of u . The term in (4) involving $\nabla^p \sigma$ acts as a regularization of σ . The choice of $p \geq 1$ regulates smoothness. Further we have $0 \leq \sigma \leq \bar{\sigma} < \infty$ a.e. in Ω . In the piecewise constant Mumford-Shah context we solve

$$\min_{u, \Gamma, \sigma} J_\nu(u, \Gamma, \sigma) = \int_{\Omega} (f - \sigma u)^2 + \delta \int_{\Omega} |\nabla^p \sigma|^2 + \nu \mathcal{H}^1(\Gamma). \quad (5)$$

This formulation opens up new possibilities, but also new challenges in Mumford-Shah based image segmentation. To the best of our knowledge, very little research has been devoted to this version of the functional.

The rest of the paper is organized as follows. In sections 2-4 we start our investigations by considering problem (2). In fact, in section 2 we introduce the topological derivative to the Mumford-Shah functional in the piecewise constant setting and without the perimeter term. We further design a solution algorithm and study its convergence properties. In section 3 we consider shape sensitivity in order to also include the perimeter term. In section 4 we report on numerical results for our approach. Then, in section 5 the generalized Mumford-Shah functional (5) is considered. For its numerical treatment, we introduce a modified version which, among others, takes care of the pointwise constraints on σ . The paper ends by a report on numerical results for simultaneous modulation recovery and image segmentation.

2 Piecewise constant Mumford-Shah functional and its topological sensitivity

Let $\Omega \subset \mathbb{R}^2$ be a bounded image domain, and let $f : \Omega \mapsto [0, 1]$ be the intensity map associated with a gray-level image. Further, $m > 1$ is an integer corresponding to the desired number of phases. The different phases are denoted by c_i and the corresponding domains by $\Omega_i, i \in \{1, \dots, m\}$. Thus, we have the partitioning of Ω :

$$\Omega = \cup_{i=1}^m \bar{\Omega}_i, \quad \Omega_i \cap \Omega_j = \emptyset \quad \forall i \neq j;$$

see figure 1. When minimizing $J_\nu(u, \Gamma)$ with respect to u , we get for $i = 1, \dots, m$

$$c_i = c_i(\Omega_i) = |\Omega_i|^{-1} \int_{\Omega_i} f(x) dx \quad \text{if } \Omega_i \neq \emptyset \quad \text{or } c_i = 0 \text{ otherwise.}$$

Since $f : \Omega \rightarrow [0, 1]$ we have $c_i \geq 0$ for all i . Let u be a bounded piecewise constant function defined by

$$u(x) = c_i \quad \forall x \in \Omega_i,$$

and denote by Γ the union of the boundaries Γ_i of the sets Ω_i

$$\Gamma = \cup_{i=1}^m \Gamma_i = \cup_{i=1}^m \partial\Omega_i. \quad (6)$$

Note that the sets Γ_i might have non-empty intersections. We can rewrite (2) in terms of Ω_i only. For this purpose we introduce a new functional \mathcal{J}_ν :

$$\mathcal{J}_\nu(\{\Omega_i\}_{i \in \{1, \dots, m\}}) = \sum_{i=1}^m \int_{\Omega_i} (f(x) - c_i)^2 dx + \frac{\nu}{2} \sum_{i=1}^m \mathcal{H}^1(\Gamma_i) + \frac{\nu}{2} \mathcal{H}^1(\partial\Omega). \quad (7)$$

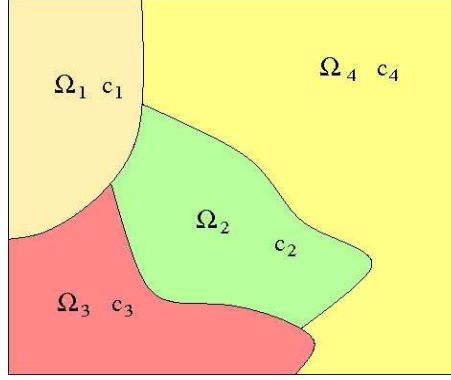


Figure 1: Partition of the domain Ω .

The coefficient $\frac{1}{2}$ and the term $\frac{\nu}{2}\mathcal{H}^1(\partial\Omega)$ in (7) come from the fact that each boundary Γ_i is counted twice in the sum (6). If we choose $\nu = 0$ in (7), then we obtain

$$\mathcal{J}_0(\{\Omega_i\}_{i \in \{1, \dots, m\}}) = J_0(u, \Gamma) = \sum_{i=1}^m \int_{\Omega_i} (f(x) - c_i)^2 dx. \quad (8)$$

2.1 Existence of an optimal solution

We show now that there exists a solution of the minimization problem for $\nu > 0$

$$\begin{aligned} \text{Minimize} \quad & \mathcal{J}_\nu(\{\Omega_i\}_{i \in \{1, \dots, m\}}) \\ \text{s.t.} \quad & \Omega = \cup_{i=1}^m \overline{\Omega}_i, \\ & \Omega_i \cap \Omega_j = \emptyset \quad \forall i \neq j, \\ & \Omega_i \text{ measurable } \forall i \in \{1, \dots, m\}. \end{aligned} \quad (9)$$

Theorem 1. *Problem (9) admits a solution $\{\Omega_i^*\}_{i \in \{1, \dots, m\}}$.*

Proof. Let $\{\Omega_i^{(n)}\}_{i \in \{1, \dots, m\}}$ be a minimizing sequence for problem (9). Then we have for $\nu > 0$ that $\mathcal{H}^1(\Gamma_i^{(n)}) < M_1$ where M_1 is a constant which does not depend on i or n . Since Ω is bounded, we have

$$|\Omega_i^{(n)}| + \mathcal{H}^1(\Gamma_i^{(n)}) < M_2,$$

for some constant $M_2 > 0$. Thus, [8, Thm 2.3.10 (p. 61)] (see also [2, Thm 5.3 (p. 128)] and [24]) yields the existence of $\{\Omega_i^*\}_{i \in \{1, \dots, m\}}$ such that for all $i \in \{1, \dots, m\}$,

$$\chi_{\Omega_i^{(n)}} \longrightarrow \chi_{\Omega_i^*} \text{ in } L^1(\Omega) \quad (10)$$

up to a subsequence. We immediately infer that

$$|\Omega_i^*| = \lim_{n \rightarrow \infty} |\Omega_i^{(n)}|.$$

From (10) we get

$$\int_{\Omega} f \chi_{\Omega_i^{(n)}} \rightarrow \int_{\Omega} f \chi_{\Omega_i^*},$$

and as a consequence

$$c_i^{(n)} \rightarrow c_i^*.$$

Thus, we infer

$$\sum_{i=1}^m \int_{\Omega_i^{(n)}} (f(x) - c_i^{(n)})^2 dx \rightarrow \sum_{i=1}^m \int_{\Omega_i^*} (f(x) - c_i^*)^2 dx.$$

From [8, Prop 2.3.6 (p. 60)] (see also [2, 24]) we also have

$$\mathcal{H}^1(\Gamma_i^*) \leq \liminf \mathcal{H}^1(\Gamma_i^{(n)}).$$

Gathering the previous results we obtain

$$\mathcal{J}_{\nu}(\{\Omega_i^*\}_{i \in \{1, \dots, m\}}) \leq \liminf \mathcal{J}_{\nu}(\{\Omega_i^{(n)}\}_{i \in \{1, \dots, m\}}) = \min \mathcal{J}_{\nu}(\{\Omega_i\}_{i \in \{1, \dots, m\}}),$$

and hence

$$\mathcal{J}_{\nu}(\{\Omega_i^*\}_{i \in \{1, \dots, m\}}) = \min \mathcal{J}_{\nu}(\{\Omega_i\}_{i \in \{1, \dots, m\}}).$$

This proves the assertion. ■

2.2 Topological derivative

In this section we set $\nu = 0$ and write \mathcal{J} instead of \mathcal{J}_0 . This is motivated by the fact that the perimeter term corresponds to lower dimensional sets. Keeping this term would induce a singularity in the topological asymptotic analysis. Rather it is dealt with by shape sensitivity in a subsequent section. We note that our treatment of the perimeter term follows a standard procedure in shape optimization.

Let $\rho > 0$ be a positive scalar and let x_0 be a point in Ω_i . Then $B_{\rho} := B(x_0, \rho)$ denotes the ball of radius ρ and center x_0 . The topological derivative $\mathcal{T}_{i,j}$ of \mathcal{J} at x_0 is defined in the following way:

$$\begin{aligned} & \mathcal{J}(\Omega_1, \dots, \Omega_i \setminus B_{\rho}, \dots, \Omega_j \cup B_{\rho}, \dots, \Omega_m) \\ &= \mathcal{J}(\{\Omega_i\}_{i \in \{1, \dots, m\}}) + \pi \rho^2 \mathcal{T}_{i,j}(x_0) + o(\rho^2); \end{aligned} \quad (11)$$

see [19] for a definition in a more general shape and topology optimization context. It provides a criterion for removing a small part from Ω_i and adding it to Ω_j . In fact, if $\mathcal{T}_{i,j}(x_0)$ is negative, then the functional \mathcal{J} is decreasing

for ρ sufficiently small. Therefore we define a matrix-valued function \mathcal{T} containing the different topological derivatives corresponding to all possible cases:

$$\mathcal{T} := \{\mathcal{T}_{i,j}\}_{(i,j) \in \{1,\dots,m\}^2}. \quad (12)$$

From definition (11) we see that $\mathcal{T}_{i,i} \equiv 0$ for all $i \in \{1,\dots,m\}$. Next we calculate the topological derivative $\mathcal{T}_{i,j}$ for $i \neq j$. Assuming $|\Omega_i| > 0$, we first compute the expansion of c_i in the case where we "remove material" from Ω_i :

$$\begin{aligned} c_i(\Omega_i \setminus B_\rho) - c_i(\Omega_i) &= \frac{1}{|\Omega_i \setminus B_\rho|} \int_{\Omega_i \setminus B_\rho} f(x) dx - \frac{1}{|\Omega_i|} \int_{\Omega_i} f(x) dx \\ &= \frac{|B_\rho|}{|\Omega_i|} \left(c_i(\Omega_i) - \frac{1}{|B_\rho|} \int_{B_\rho} f(x) dx \right) \\ &\quad + \frac{|B_\rho|^2}{|\Omega_i| |\Omega_i \setminus B_\rho|} \left(c_i(\Omega_i) - \frac{1}{|B_\rho|} \int_{B_\rho} f(x) dx \right). \end{aligned}$$

The expansion of c_j when we "add material" to Ω_j depends on whether $|\Omega_j| = 0$ or not. If $|\Omega_j| \neq 0$, then we write

$$\begin{aligned} c_j(\Omega_j \cup B_\rho) - c_j(\Omega_j) &= -\frac{|B_\rho|}{|\Omega_j|} \left(c_j(\Omega_j) - \frac{1}{|B_\rho|} \int_{B_\rho} f(x) dx \right) \\ &\quad + \frac{|B_\rho|^2}{|\Omega_j| |\Omega_j \cup B_\rho|} \left(c_j(\Omega_j) - \frac{1}{|B_\rho|} \int_{B_\rho} f(x) dx \right), \end{aligned}$$

otherwise, if $|\Omega_j| = 0$, we have

$$c_j(\Omega_j \cup B_\rho) = |B_\rho|^{-1} \int_{B_\rho} f(x) dx.$$

Now, if $|\Omega_j| \neq 0$, then we obtain

$$\begin{aligned} &\mathcal{J}(\Omega_1, \dots, \Omega_i \setminus B_\rho, \dots, \Omega_j \cup B_\rho, \dots, \Omega_m) - \mathcal{J}(\{\Omega_i\}_{i \in \{1,\dots,m\}}) \\ &= \int_{\Omega_j \cup B_\rho} (f(x) - c_j(\Omega_j \cup B_\rho))^2 dx + \int_{\Omega_i \setminus B_\rho} (f(x) - c_i(\Omega_i \setminus B_\rho))^2 dx \\ &\quad - \int_{\Omega_j} (f(x) - c_j(\Omega_j))^2 dx - \int_{\Omega_i} (f(x) - c_i(\Omega_i))^2 dx. \end{aligned}$$

Taking into account the expansions of $c_i(\Omega_i \setminus B_\rho)$ and $c_j(\Omega_j \cup B_\rho)$ results in

$$\begin{aligned}
& \mathcal{J}(\Omega_1, \dots, \Omega_i \setminus B_\rho, \dots, \Omega_j \cup B_\rho, \dots, \Omega_m) - \mathcal{J}(\{\Omega_i\}_{i \in \{1, \dots, m\}}) \\
&= \int_{B_\rho} (f(x) - c_j(\Omega_j))^2 - (f(x) - c_i(\Omega_i))^2 dx \\
&\quad + 2 \frac{|B_\rho|}{|\Omega_j|} \left(c_j(\Omega_j) - \frac{1}{|B_\rho|} \int_{B_\rho} f(x) dx \right) \int_{\Omega_j} (f(x) - c_j(\Omega_j)) dx \\
&\quad - 2 \frac{|B_\rho|}{|\Omega_i|} \left(c_i(\Omega_i) - \frac{1}{|B_\rho|} \int_{B_\rho} f(x) dx \right) \int_{\Omega_i} (f(x) - c_i(\Omega_i)) dx + \mathcal{O}(|B_\rho|)
\end{aligned}$$

as $|B_\rho| \rightarrow 0$ for $\rho \rightarrow 0$. From Lebesgue's differentiation theorem [3] and $\int_{\Omega_i} (f(x) - c_i(\Omega_i)) dx = \int_{\Omega_j} (f(x) - c_j(\Omega_j)) dx = 0$ we infer

$$\mathcal{T}_{i,j}(x_0) = (f(x_0) - c_j(\Omega_j))^2 - (f(x_0) - c_i(\Omega_i))^2 \text{ for almost all } x_0 \in \Omega. \quad (13)$$

In the case where $|\Omega_j| = 0$ we have

$$\begin{aligned}
& \mathcal{J}(\Omega_1, \dots, \Omega_i \setminus B_\rho, \dots, \Omega_j \cup B_\rho, \dots, \Omega_m) - \mathcal{J}(\{\Omega_i\}_{i \in \{1, \dots, m\}}) \\
&= \int_{B_\rho} (f(x) - c_j(\Omega_j \cup B_\rho))^2 dx + \int_{\Omega_i \setminus B_\rho} (f(x) - c_i(\Omega_i \setminus B_\rho))^2 dx \\
&\quad - \int_{\Omega_i} (f(x) - c_i(\Omega_i))^2 dx \\
&= \int_{B_\rho} (f(x) - c_j(\Omega_j \cup B_\rho))^2 - (f(x) - c_i(\Omega_i))^2 dx \\
&\quad - 2 \frac{|B_\rho|}{|\Omega_i|} \left(c_i(\Omega_i) - \frac{1}{|B_\rho|} \int_{B_\rho} f(x) dx \right) \int_{\Omega_i} (f(x) - c_i(\Omega_i)) dx \\
&\quad + \mathcal{O}(|B_\rho|)
\end{aligned}$$

as $|B_\rho| \rightarrow 0$ for $\rho \rightarrow 0$. Since $c_j(\Omega_j \cup B_\rho) = |B_\rho|^{-1} \int_{B_\rho} f(x) dx \rightarrow f(x_0)$ for almost all $x_0 \in \Omega$, we get

$$\mathcal{T}_{i,j}(x_0) = -(f(x_0) - c_i(\Omega_i))^2 \text{ for almost all } x_0 \in \Omega. \quad (14)$$

2.3 Algorithm for topological derivative

The initialization of our algorithmic procedure for solving (9) with $\nu = 0$ uses $\Omega_1 = \Omega$ and $\Omega_i = \emptyset$ for all $i > 1$. Then we determine Ω_2 by computing $\mathcal{T}_{1,2}$. Following equation (11), we may choose

$$\Omega_2 = \{x_0 \in \Omega_1 \mid \mathcal{T}_{1,2}(x_0) < 0\}.$$

Actually, since the topological derivative is only a local criterion, we prefer to change the domain Ω_1 only when the topological derivative is sufficiently

negative. In order to do that, we fix a prescribed tolerance $0 \leq \gamma < 1$ and set

$$\Omega_2 = \left\{ x_0 \in \Omega_1 \mid \mathcal{T}_{1,2}(x_0) < \gamma \min_{y \in \Omega_1} \mathcal{T}_{1,2}(y) \right\}.$$

Once we have initialized the sets Ω_k , $k = 1, \dots, i < m$, we set

$$\Omega_{i+1} = \bigcup_{k=1}^i \left\{ x_0 \in \Omega_k \mid \mathcal{T}_{k,i+1}(x_0) < \gamma \min_{y \in \Omega_k} \mathcal{T}_{k,i+1}(y) \right\}. \quad (15)$$

Therefore the topological algorithm operates in two steps. In the first step we initialize all the domains Ω_i , and in the second step we modify the domains according to the topological derivatives between all the m existing domains. Ideally, convergence is reached when all the topological derivatives are zero. However, for numerical purposes, we stop the algorithm as soon as

$$\sum_{i,j=1}^m \|\mathcal{T}_{i,j}^{(l)}\|_2 \leq \mu_t \left(1 + \sum_{i,j=1}^m \|\mathcal{T}_{i,j}^{(0)}\|_2 \right),$$

where $\|\mathcal{T}_{i,j}^{(l)}\|_2$ is the L_2 -norm of $\mathcal{T}_{i,j}^{(l)}$ and $0 < \mu_t \ll 1$ denotes a user-specified stopping tolerance.

In what follows we propose two different algorithms. The first algorithm is based on the previous idea, which is a standard technique for using the topological derivative. The second one turns out to be much faster in practice. Below, theorem 2 provides a relation between the two methods.

Algorithm 1: Initialization

Input: $\Omega, f, m, \gamma, \mu_t$

Output: $\Omega_i, c_i, i = 1, \dots, m$.

Initialization: Set $\Omega_1^{(0)} := \Omega$ and, thus, $c_1^{(0)} = |\Omega|^{-1} \int_{\Omega} f(x) dx$. Set $l := 0$.

for $i = 1, \dots, m - 1$ **do**

for $k = 1, \dots, i$ **do**

 Compute $c_k^{(0)} = |\Omega_k^{(0)}|^{-1} \int_{\Omega_k^{(0)}} f(x) dx$.

 Compute $\mathcal{T}_{k,i+1}^{(0)}(x_0) = -(f(x_0) - c_k^{(0)})^2$.

 Initialize $\Omega_{i+1}^{(0)}$ by using (15), and update $\Omega_k^{(0)}$ by setting

$$\Omega_k^{(0)} = \Omega_k^{(0)} \setminus \left\{ x_0 \in \Omega_k^{(0)} \mid \mathcal{T}_{k,i+1}^{(0)}(x_0) < \gamma \min_{y \in \Omega_k^{(0)}} \mathcal{T}_{k,i+1}^{(0)}(y) \right\}.$$

end

end

while $\sum_{i,j=1}^m \|\mathcal{T}_{i,j}^{(l)}\|_2 > \mu_t \left(1 + \sum_{i,j=1}^m \|\mathcal{T}_{i,j}^{(0)}\|_2 \right)$ **do**

 Set $\Omega_k^{(l+1)} = \Omega_k^{(l)} \quad \forall k \in \{1, \dots, m\}$.

for $i = 1, \dots, m$ **do**

 Compute $\mathcal{T}_{i,j}^{(l+1)}$ for all $x_0 \in \Omega_i^{(l)}$ and all $j \in \{1, \dots, m\}$. Define

$$T_i^{(l+1)}(x_0) = \min_{j \in \{1, \dots, m\}} \mathcal{T}_{i,j}^{(l+1)}(x_0).$$

 Determine the sets

$$A_i^{(l+1)} = \left\{ x_0 \in \Omega_i^{(l)} \mid T_i^{(l+1)}(x_0) < \gamma \min_{y \in \Omega_i^{(l)}} T_i^{(l+1)}(y) \right\},$$

$$\Omega_i^{(l+1)} = \Omega_i^{(l+1)} \setminus A_i^{(l+1)}.$$

for $j = 1, \dots, m, j \neq i$ **do**

$$\Omega_j^{(l+1)} = \Omega_j^{(l+1)} \cup \left\{ x_0 \in A_i^{(l+1)} \mid \mathcal{T}_{i,j}^{(l+1)}(x_0) = T_i^{(l+1)}(x_0) \right\},$$

$$A_i^{(l+1)} = A_i^{(l+1)} \setminus \left\{ x_0 \in A_i^{(l+1)} \mid \mathcal{T}_{i,j}^{(l+1)}(x_0) = T_i^{(l+1)}(x_0) \right\}$$

end

end

for $i = 1, \dots, m$ **do**

 If $|\Omega_i^{(l+1)}| \neq 0$ set $c_i^{(l+1)} = |\Omega_i^{(l+1)}|^{-1} \int_{\Omega_i^{(l+1)}} f(x) dx$.

 Else set $c_i^{(l+1)} = 0$.

end

$l = l + 1$

end

In our implementation, in the above stopping rule $\|\cdot\|_2$ and μ_t are replaced by a discrete L_2 -norm and $\mu_t h$, respectively, with $h = \max(1/n_{x_i} : i = 1, \dots, d_f)$, where n_{x_i} denotes the number of pixels in the x_i -direction.

Algorithm 1 can be straightforwardly extended to more general situations, such as the modulation recovery in section 5. In the specific context of piecewise constant approximations, however, the following variant of the previous algorithm is much faster and streamlines computations. Concerning the latter note that the computation of the $\mathcal{T}_{i,j}$ -values is replaced by a direct computation of $\Omega_i^{(l+1)}$ according to the structure of the topological sensitivities. For the definition of this algorithm we need the symmetric difference of two sets A and B , i.e.,

$$A\Delta B = (A \cap B^c) \cup (A^c \cap B),$$

where superscript 'c' refers to the complement of the respective set in Ω . We further define the arithmetic mean of two subsequent phases by

$$d_i = \frac{c_{i-1} + c_i}{2} \quad \forall i \in \{2, \dots, m\}. \quad (16)$$

Algorithm 2:

Input: Ω, f, m .

Output: $\Omega_i, c_i, i = 1, \dots, m$.

Initialization: Initialize $c_i^{(0)}, i \in \{1, \dots, m\}$ such that $\min(f) \leq c_1^{(0)} < \dots < c_i^{(0)} < \dots < c_m^{(0)} \leq \max(f)$. Set $l = 0$ and $\Omega_i^{(0)} = \emptyset \forall i \in \{1, \dots, m\}$.

while [$l > 0$ **and** $|\Omega_i^{(l)} \Delta \Omega_i^{(l-1)}| > 0 \forall i$] **or** $l = 0$] **do**

Compute $d_i^{(l)}, i \in \{2, \dots, m\}$, set $d_1^{(l)} < 0, d_{m+1}^{(l)} = \max(f)$.

Set $\Omega_i^{(l+1)} = \{x \in \Omega \mid d_i^{(l)} < f(x) \leq d_{i+1}^{(l)}\} \quad \forall i \in \{1, \dots, m\}$.

for $i = 1, \dots, m$ **do**

if $|\Omega_i^{(l+1)}| > 0$ **then**

Update $c_i^{(l+1)} = |\Omega_i^{(l+1)}|^{-1} \int_{\Omega_i^{(l+1)}} f(x) dx$.

else

Choose arbitrary $c_i^{(l+1)}$ outside the interval $[d_{i-1}^{(l)}, d_i^{(l)}]$.

end

end

set $l = l + 1$

end

A few words on the algorithm are in order. First note that algorithm 2 for minimizing the piecewise constant Mumford-Shah functional (Chan-Vese model) without perimeter term is closely related to the k -means clustering

algorithm; see, e.g., [5, 6, 11] for details on the latter. Secondly, in the else-branch of the if-statement alternative choices of $c_i^{(l+1)}$ even within $[d_{i-1}^{(l)}, d_i^{(l)}]$ are possible. Keeping the same value than in the previous step, however, is only meaningful if at least one of the other c_i -values changes. Our suggestion in the else-branch above is motivated by the fact that $c_i^{(l)} \in [d_{i-1}^{(l)}, d_i^{(l)}]$ produced $\Omega_i^{l+1} = \emptyset$. Further, below we show that algorithm 2 is similar to algorithm 1 for $\gamma = 0$. We also prove a monotonicity property of algorithm 2; see Proposition 1 below. For the initialization of the phases one may use

$$c_i^{(0)} = \min(f) + i \frac{(\max(f) - \min(f))}{m+1} \quad \text{for } i = 1, \dots, m,$$

for instance, but other choices are possible.

We start our investigation by proving several auxiliary results. In what follows, we assume

$$\min(f) \leq c_1^{(0)} < \dots < c_i^{(0)} < \dots < c_m^{(0)} \leq \max(f). \quad (17)$$

Lemma 1. *Define*

$$\mathcal{T}_{i,j}(x) := (f(x) - c_j)^2 - (f(x) - c_i)^2 \quad \forall i, j \in \{1, \dots, m\}, \quad \forall x \in \Omega. \quad (18)$$

Then

$$\mathcal{T}_{i,j}(x) = \mathcal{T}_{i,k}(x) + \mathcal{T}_{k,j}(x).$$

Proof: The proof follows immediately from the definition of $\mathcal{T}_{i,j}$. ■

Note that $\mathcal{T}_{i,j}$ in (18) is an extended version of the topological derivative in (11) as x is arbitrary in Ω (and not just Ω_i).

Lemma 2. *Let $\mathcal{T}_{i,j}$ be defined as in (18). Then*

$$\mathcal{T}_{i,j}(x) < 0 \iff \begin{cases} \text{either} & c_i < c_j & \text{and} & f(x) > \frac{c_i + c_j}{2}, \\ \text{or} & c_i > c_j & \text{and} & f(x) < \frac{c_i + c_j}{2}. \end{cases}$$

Proof: Note that $\mathcal{T}_{i,j}$ can be written in the form

$$\mathcal{T}_{i,j}(x) = (c_i - c_j)(2f(x) - c_i - c_j).$$

Then the assertion follows immediately. ■

Lemma 3. *Let $x \in \Omega_k$, and let p be the smallest integer such that*

$$\mathcal{T}_{k,p}(x) = \min_{l \in \{1, \dots, m\}} \mathcal{T}_{k,l}(x). \quad (19)$$

Moreover, assume that the c_i 's satisfy (17). Then

$$d_p < f(x) \leq d_{p+1}. \quad (20)$$

Conversely, if $f(x)$ satisfies (20) for some \hat{p} then

$$\mathcal{T}_{k,\hat{p}}(x) = \min_{l \in \{1, \dots, m\}} \mathcal{T}_{k,l}(x).$$

Proof: If $p = m$, then $f(x) \leq d_{p+1} = \max(f)$ is true. Next suppose $p < m$. We assume that $f(x) > d_{p+1}$. Then, according to lemma 2, we get

$$\mathcal{T}_{p,p+1}(x) < 0,$$

and, according to lemma 1,

$$\mathcal{T}_{k,p}(x) + \mathcal{T}_{p,p+1}(x) = \mathcal{T}_{k,p+1}(x) < \mathcal{T}_{k,p}(x).$$

This, however, contradicts our assumption (19). Thus $f(x) \leq d_{p+1}$.

Now we study the second inequality. If $p = 1$, the inequality is fulfilled since $d_1 < 0$ and $f(x) \geq 0$. Now assume that $f(x) \leq d_p$ and $p > 1$. First, if $f(x) < d_p$, we get $\mathcal{T}_{p,p-1}(x) < 0$ and

$$\mathcal{T}_{k,p}(x) + \mathcal{T}_{p,p-1}(x) = \mathcal{T}_{k,p-1}(x) < \mathcal{T}_{k,p}(x),$$

which is impossible due to our assumption (19). Thus, we have $f(x) \geq d_p$. Further, if $f(x) = d_p$, it is easy to see that $\mathcal{T}_{p,p-1}(x) = 0$. Hence we have

$$\mathcal{T}_{k,p}(x) + \mathcal{T}_{p,p-1}(x) = \mathcal{T}_{k,p-1}(x) = \mathcal{T}_{k,p}(x) = \min_{l \in \{1, \dots, m\}} \mathcal{T}_{k,l}(x).$$

Again, this contradicts (19). Therefore, $f(x) > d_p$ holds true.

Conversely, assume that $f(x)$ satisfies (20) for some $1 < \hat{p} < m$, and let $q \geq \hat{p} + 1$. Then

$$f(x) \leq d_{\hat{p}+1} \leq \frac{c_{\hat{p}} + c_q}{2},$$

and, according to lemma 2, we get $\mathcal{T}_{q,\hat{p}}(x) \leq 0$ and, thus, $\mathcal{T}_{\hat{p},q}(x) \geq 0$. Further

$$\mathcal{T}_{k,\hat{p}}(x) + \mathcal{T}_{\hat{p},q}(x) = \mathcal{T}_{k,q}(x) \geq \mathcal{T}_{k,\hat{p}}(x).$$

In a similar way, if $q \leq \hat{p} - 1$, we have

$$f(x) > d_{\hat{p}} \geq \frac{c_{\hat{p}} + c_q}{2},$$

and, according to lemma 2, we get $\mathcal{T}_{q,\hat{p}}(x) \leq 0$ and, thus, $\mathcal{T}_{\hat{p},q}(x) \geq 0$. Further

$$\mathcal{T}_{k,\hat{p}}(x) + \mathcal{T}_{\hat{p},q}(x) = \mathcal{T}_{k,q}(x) > \mathcal{T}_{k,\hat{p}}(x).$$

Finally we get

$$\mathcal{T}_{k,\hat{p}}(x) = \min_{l \in \{1, \dots, m\}} \mathcal{T}_{k,l}(x),$$

which proves the converse statement. ■

Now we are going to prove a theorem which shows that algorithm 2 and algorithm 1 have a similar behavior.

Theorem 2. *Let $l \geq 1$. Assume that $\Omega_i^{(l)}, i \in \{1, \dots, m\}$, and the corresponding $\min(f) \leq c_1^{(l)} < \dots < c_i^{(l)} < \dots < c_m^{(l)} \leq \max(f)$ are given. Assume that*

$$\hat{l}_k(x) = \operatorname{argmin}_{l \in \{1, \dots, m\}} \mathcal{T}_{k,l}(x) \quad (21)$$

is unique for all x in Ω and $k \in \{1, \dots, m\}$. Then, after one iteration of the main loop of algorithm 1 with $\gamma = 0$, we get:

$$\begin{aligned} \Omega_k^{(l+1)} = & \left(\left\{ x \in \Omega \mid d_k^{(l)} < f(x) \leq d_{k+1}^{(l)} \right\} \right. \\ & \left. \cup \left\{ x \in \Omega_k^{(l)} \mid f(x) = d_k^{(l)} \right\} \right) \setminus \left\{ x \in \Omega_{k+1}^{(l)} \mid f(x) = d_{k+1}^{(l)} \right\}. \end{aligned}$$

Proof: In algorithm 1, $A_k^{(l+1)}$ is defined in the following way:

$$A_k^{(l+1)} = \left\{ x \in \Omega_k^{(l)} \mid T_k^{(l+1)}(x) < \gamma \min_{y \in \Omega_i^{(l)}} T_k^{(l+1)}(y) \right\}.$$

If $\gamma = 0$ we have

$$A_k^{(l+1)} = \left\{ x \in \Omega_k^{(l)} \mid T_k^{(l+1)}(x) < 0 \right\}.$$

which can be written as

$$A_k^{(l+1)} = \bigcup_{j=1}^m B_{j,k}^{(l+1)}, \quad (22)$$

with

$$B_{j,k}^{(l+1)} = \left\{ x \in \Omega_k^{(l)} \mid d_j^{(l)} < f(x) \leq d_{j+1}^{(l)} \text{ and } T_k^{(l+1)}(x) < 0 \right\}.$$

Now we analyse different cases for $B_{j,k}$:

First case: If $k = j$, then $B_{j,k}^{(l+1)} = \emptyset$. Indeed, if $d_k^{(l)} < f(x) \leq d_{k+1}^{(l)}$, then we get from lemma 3

$$0 = \mathcal{T}_{k,k}^{(l+1)}(x) = \min_{l \in \{1, \dots, m\}} \mathcal{T}_{k,l}^{(l+1)}(x)$$

and, thus, $T_k^{(l+1)}(x) = 0$.

Second case: If $k > j + 1$, then

$$B_{j,k}^{(l+1)} = \left\{ x \in \Omega_k^{(l)} \mid d_j^{(l)} < f(x) \leq d_{j+1}^{(l)} \right\}.$$

Indeed, we have

$$f(x) \leq d_{j+1}^{(l)} = \frac{c_j^{(l)} + c_{j+1}^{(l)}}{2} < \frac{c_j^{(l)} + c_k^{(l)}}{2}.$$

Thus, $\mathcal{T}_{k,j}^{(l+1)}(x) < 0$ according to lemma 2 and, hence, $T_k^{(l+1)}(x) < 0$.

Third case: If $k = j + 1$, we either have

$$f(x) < d_{j+1}^{(l)} \implies \mathcal{T}_{j+1,j}^{(l+1)}(x) < 0$$

or

$$f(x) = d_{j+1}^{(l)} \implies \mathcal{T}_{j+1,j}^{(l+1)}(x) = 0.$$

In the second case, according to lemma 3 we get

$$0 = \mathcal{T}_{j+1,j}^{(l+1)}(x) = \min_{l \in \{1, \dots, m\}} \mathcal{T}_{j+1,l}^{(l+1)}(x)$$

which implies $T_{j+1}^{(l+1)}(x) = 0$. This allows us to write $B_{j,k}^{(l+1)}$ as

$$B_{j,k}^{(l+1)} = \left\{ x \in \Omega_k^{(l)} \mid d_j^{(l)} < f(x) < d_{j+1}^{(l)} \right\}.$$

Fourth case: If $k < j$, then we have

$$f(x) > d_j^{(l)} = \frac{c_{j-1}^{(l)} + c_j^{(l)}}{2} \geq \frac{c_k^{(l)} + c_j^{(l)}}{2}$$

which implies $\mathcal{T}_{k,j}^{(l+1)}(x) < 0$ according to lemma 2. Thus $T_k^{(l+1)}(x) < 0$. Finally, in this case we get

$$B_{j,k}^{(l+1)} = \left\{ x \in \Omega_k^{(l)} \mid d_j^{(l)} < f(x) \leq d_{j+1}^{(l)} \right\}.$$

In view of algorithm 1 and assumption (21), after one iteration, we obtain

$$\begin{aligned} \Omega_k^{(l+1)} &= (\Omega_k^{(l)} \setminus A_k^{(l+1)}) \cup \left(\bigcup_{i=1, i \neq k}^m \left\{ x \in A_i^{(l+1)} \mid \mathcal{T}_{i,k}^{(l+1)}(x) = T_i^{(l+1)}(x) \right\} \right) \\ &= (\Omega_k^{(l)} \setminus A_k^{(l+1)}) \cup \left(\bigcup_{i=1, i \neq k}^m A_i^{(l+1)} \right) \cap \left(\bigcup_{i=1, i \neq k}^m \left\{ x \in \Omega_i^{(l)} \mid \mathcal{T}_{i,k}^{(l+1)}(x) = T_i^{(l+1)}(x) \right\} \right) \end{aligned}$$

Now according to (22) and in view of lemma 3 we have

$$\begin{aligned}\Omega_k^{(l+1)} &= (\Omega_k^{(l)} \setminus A_k^{(l+1)}) \cup \left(\bigcup_{i=1, i \neq k}^m \bigcup_{j=1}^m B_{j,i}^{(l+1)} \right) \cap \left(\bigcup_{i=1, i \neq k}^m \left\{ x \in \Omega_i^{(l)} \mid \mathcal{T}_{i,k}^{(l+1)}(x) = T_i^{(l+1)}(x) \right\} \right) \\ &= (\Omega_k^{(l)} \setminus A_k^{(l+1)}) \cup \left(\bigcup_{j=1}^m \bigcup_{i=1, i \neq k}^m \left(B_{j,i}^{(l+1)} \cap \left\{ x \in \Omega_i^{(l)} \mid d_k^{(l)} < f(x) \leq d_{k+1}^{(l)} \right\} \right) \right).\end{aligned}$$

According to our above characterization of $B_{j,k}^{(l+1)}$ we get

$$B_{j,i}^{(l+1)} \cap \left\{ x \in \Omega_i^{(l)} \mid d_k^{(l)} < f(x) \leq d_{k+1}^{(l)} \right\} = \emptyset \text{ if } j \neq k.$$

As a consequence

$$\begin{aligned}\Omega_k^{(l+1)} &= (\Omega_k^{(l)} \setminus A_k^{(l+1)}) \cup \left(\bigcup_{i=1, i \neq k, k+1}^m \left\{ x \in \Omega_i^{(l)} \mid d_k^{(l)} < f(x) \leq d_{k+1}^{(l)} \right\} \right) \\ &\cup \left\{ x \in \Omega_{k+1}^{(l)} \mid d_k^{(l)} < f(x) < d_{k+1}^{(l)} \right\}.\end{aligned}$$

In addition we have

$$\begin{aligned}\Omega_k^{(l)} \setminus A_k^{(l+1)} &= \left\{ x \in \Omega_k^{(l)} \mid T_k^{(l+1)}(x) \geq 0 \right\} \\ &= \left\{ x \in \Omega_k^{(l)} \mid d_k^{(l)} \leq f(x) \leq d_{k+1}^{(l)} \right\}\end{aligned}$$

which leads to

$$\begin{aligned}\Omega_k^{(l+1)} &= \left\{ x \in \Omega_k^{(l)} \mid d_k^{(l)} \leq f(x) \leq d_{k+1}^{(l)} \right\} \cup \left\{ x \in \Omega_{k+1}^{(l)} \mid d_k^{(l)} < f(x) < d_{k+1}^{(l)} \right\} \\ &\cup \left(\bigcup_{i=1, i \neq k, k+1}^m \left\{ x \in \Omega_i^{(l)} \mid d_k^{(l)} < f(x) \leq d_{k+1}^{(l)} \right\} \right),\end{aligned}$$

and finally

$$\begin{aligned}\Omega_k^{(l+1)} &= \left(\left\{ x \in \Omega \mid d_k^{(l)} < f(x) \leq d_{k+1}^{(l)} \right\} \right. \\ &\quad \left. \cup \left\{ x \in \Omega_k^{(l)} \mid f(x) = d_k^{(l)} \right\} \right) \setminus \left\{ x \in \Omega_{k+1}^{(l)} \mid f(x) = d_{k+1}^{(l)} \right\}.\end{aligned}$$

This concludes the proof. ■

Remark 1. *Theorem 2 states that the updated domain $\Omega_k^{(l+1)}$ of algorithm 1 differs from the updated domain $\Omega_k^{(l+1)}$ of algorithm 2 by the set*

$$\left\{ x \in \Omega_k^{(l)} \mid f(x) = d_k^{(l)} \right\} \setminus \left\{ x \in \Omega_{k+1}^{(l)} \mid f(x) = d_{k+1}^{(l)} \right\},$$

which will be negligible for most "real-life" images. Moreover, whenever the index $\hat{l}_k(x)$ in (21) is not unique, x may be moved into any of the sets yielding the minimum. In our implementation the set with the smallest index is chosen. In practice such a non-uniqueness typically occurs only for a few pixels during the segmentation except maybe in certain cases of piecewise constant data.

Next, we establish a monotonicity result for Algorithm 2. For its proof we need the following auxiliary result, which requires the mean value \bar{f}_D of a given function $f : A \rightarrow \mathbb{R}$, where A is nonempty and bounded and $D \subset A$ with positive Lebesgue measure $|D|$, defined by

$$\bar{f}_D = |D|^{-1} \int_D f(x) dx.$$

Lemma 4. *Let $f \in L^\infty(A, \mathbb{R})$, $A \subset \mathbb{R}^N$ bounded, $N \geq 1$, and define*

$$\begin{aligned} A_1 &= \{x \in A \mid a < f(x) \leq b\}, \\ A_2 &= \{x \in A \mid c < f(x) \leq d\} \end{aligned}$$

with $a \leq c$ and $b \leq d$. We assume that $|A_1| \neq 0$ and $|A_2| \neq 0$. Then we have

$$\bar{f}_{A_1} \leq \bar{f}_{A_2}.$$

Proof: First of all, if $c \geq b$ we immediately find

$$\bar{f}_{A_2} > c \geq b \geq \bar{f}_{A_1}.$$

On the other hand, if $c < b$ we define the sets

$$\begin{aligned} B_1 &= \{x \in A \mid c < f(x) \leq b\}, \\ C_1 &= \{x \in A \mid a < f(x) \leq c\}. \end{aligned}$$

Note that $|B_1| = 0$ and $|C_1| = 0$ cannot happen simultaneously since $A_1 = B_1 \cup C_1$ and $|A_1| \neq 0$. If $|B_1| = 0$, then we have

$$\bar{f}_{A_2} > c \geq \bar{f}_{A_1}.$$

Therefore, in what follows, we assume that $|B_1| \neq 0$. If $|C_1| = 0$, then $A_1 = B_1$ and, consequently,

$$\bar{f}_{A_1} = \bar{f}_{B_1};$$

otherwise, if $|C_1| \neq 0$, then there holds

$$\bar{f}_{A_1} < \bar{f}_{B_1}. \tag{23}$$

Indeed, first note that $\bar{f}_{C_1} < \bar{f}_{B_1}$ and further

$$\begin{aligned}
\bar{f}_{A_1} - \bar{f}_{B_1} &= |A_1|^{-1} \left(\int_{B_1} f(x) dx + \int_{C_1} f(x) dx \right) - \bar{f}_{B_1} \\
&= \frac{|B_1|}{|A_1|} \bar{f}_{B_1} + \frac{|C_1|}{|A_1|} \bar{f}_{C_1} - \bar{f}_{B_1} \\
&< \frac{|B_1|}{|A_1|} \bar{f}_{B_1} + \frac{|C_1|}{|A_1|} \bar{f}_{B_1} - \bar{f}_{B_1} \\
&< \bar{f}_{B_1} - \bar{f}_{B_1} \\
&< 0.
\end{aligned}$$

Similarly, if $|C_2| \neq 0$, with $C_2 = \{x \in A \mid b < f(x) \leq d\}$, then we have

$$\bar{f}_{B_1} < \bar{f}_{A_2}.$$

On the other hand, if $|C_2| = 0$, then

$$\bar{f}_{B_1} = \bar{f}_{A_2}.$$

Finally we get

$$\bar{f}_{A_1} \leq \bar{f}_{A_2}.$$

The equality in the previous inequality occurs only if $|C_1| = 0$ and $|C_2| = 0$. ■

The monotonicity of the phases is addressed next.

Proposition 1. *Using Algorithm 2, if there exists an integer n_0 such that*

$$c_k^{(n_0+1)} \geq c_k^{(n_0)} \quad \forall k \in \{1, \dots, m\},$$

then we have

$$c_k^{(n+1)} \geq c_k^{(n)} \quad \forall k \in \{1, \dots, m\} \quad \forall n \geq n_0. \quad (24)$$

In addition, if there exists an integer n_1 such that

$$c_k^{(n_1)} \leq c_k^* \quad \forall k \in \{1, \dots, m\},$$

then we have

$$c_k^{(n+1)} \leq c_k^* \quad \forall k \in \{1, \dots, m\} \quad \forall n \geq n_1, \quad (25)$$

where

$$c_k^* = |\Omega_k^*|^{-1} \int_{\Omega_k^*} f(x) dx$$

and Ω_k^ is a solution of problem (9).* ■

Proof: We prove (24) by induction. Assume (24) is true for some $n > n_0$, then we can write for all $k \in \{1, \dots, m-1\}$:

$$\frac{c_k^{(n+1)} + c_{k+1}^{(n+1)}}{2} \geq \frac{c_k^{(n)} + c_{k+1}^{(n)}}{2}$$

and hence

$$d_{k+1}^{(n+1)} \geq d_{k+1}^{(n)}.$$

Since

$$\Omega_k^{(n+1)} = \left\{ x \in \Omega \mid d_k^{(n)} < f(x) \leq d_{k+1}^{(n)} \right\} \quad \forall k \in \{1, \dots, m\}$$

and

$$\Omega_k^{(n+2)} = \left\{ x \in \Omega \mid d_k^{(n+1)} < f(x) \leq d_{k+1}^{(n+1)} \right\} \quad \forall k \in \{1, \dots, m\}$$

we apply lemma 4 with $A_1 = \Omega_k^{(n+1)}$ and $A_2 = \Omega_k^{(n+2)}$ for $k \in \{2, \dots, m-1\}$ and get

$$c_k^{(n+2)} \geq c_k^{(n+1)}.$$

For $k = 1$ and $k = m$ we obtain a similar result. Concerning the second part of the proposition, we note that if we assume that (25) is true for some $n \geq n_1$, then we can write for all $k \in \{1, \dots, m-1\}$:

$$\frac{c_k^* + c_{k+1}^*}{2} \geq \frac{c_k^{(n+1)} + c_{k+1}^{(n+1)}}{2}$$

and further

$$d_{k+1}^* \geq d_{k+1}^{(n+1)}.$$

Since

$$\Omega_k^* = \left\{ x \in \Omega \mid d_k^* < f(x) \leq d_{k+1}^* \right\} \quad \forall k \in \{1, \dots, m\}$$

and

$$\Omega_k^{(n+2)} = \left\{ x \in \Omega \mid d_k^{(n+1)} < f(x) \leq d_{k+1}^{(n+1)} \right\} \quad \forall k \in \{1, \dots, m\}$$

we apply lemma 4 with $A_1 = \Omega_k^{(n+2)}$ and $A_2 = \Omega_k^*$ for $k \in \{2, \dots, m-1\}$ and get

$$c_k^{(n+2)} \leq c_k^*.$$

For $k = 1$ and $k = m$ we obtain a similar result. ■

In our numerical experience the prerequisites of Proposition 1 are typically met after a few iterations. In fact, we find an iteration \bar{n} such that all c_k -values increase (compared to their respective value in the previous iteration). In such a situation Proposition 1 guarantees that the sequences $\{c_k^{(n)}\}$, $k = 1, \dots, m$, are monotonically increasing for $n \geq \bar{n}$. In addition, if $c_k^{(\bar{n})} \leq c_k^*$ for all $k = 1, \dots, m$ then $c_k^{(n)}$, $k = 1, \dots, m$, approaches the optimal value from below as stated in the second part of the proposition.

3 Shape derivatives of the Mumford-Shah functional and the level set framework

Next we recall some theoretical aspects of shape optimization; see [20]. The analysis here is greatly simplified by the fact that we do not deal with partial differential equations. Let Γ be defined as in (6) and let n_i be the outward unit normal vector to Ω_i . In what follows $V(t, x)$ is a smooth vector field defined on $[0, T] \times \Omega$ with $V(t, x) \cdot n_i(x) = 0$ for almost every $x \in \partial\Omega$ all $t \in [0, T]$ and all $i \in \{1, \dots, m\}$. If the unit exterior normal vector n_i is not defined at a singular $x \in \Gamma$ we assume that $V(t, x) = 0$. The vector field V is said to be admissible if it satisfies these conditions. Let $x = x(t, \mathbf{x})$ denote the solution of the initial value problem

$$\frac{d}{dt}x(t, \mathbf{x}) = V(t, x(t, \mathbf{x})), \quad (26)$$

$$x(0, \mathbf{x}) = \mathbf{x}. \quad (27)$$

with $\mathbf{x} \in \Omega$ and $t \in [0, T]$, and we denote by $T^t : \Omega \rightarrow \Omega$ the time- t map with respect to (26)-(27), i.e. $T^t(\mathbf{x}) = x(t, \mathbf{x})$. We set $\Gamma^t = T^t(\Gamma)$. The Eulerian semi-derivative of J at Γ in direction of the vector field V is defined as the limit

$$dJ(\Gamma; V) = \lim_{t \rightarrow 0} \frac{1}{t} (J(\Gamma_t) - J(\Gamma)),$$

if it exists. A classical result on the structure of the shape derivative for smooth domains states (see [20]) that there exists a distribution ∇J on Γ such that

$$dJ(\Gamma; V) = \langle \nabla J, v_n \rangle_{\Gamma},$$

where $v_n(x) = V(0, x) \cdot n(x)$ and $\langle \cdot, \cdot \rangle_{\Gamma}$ denotes an appropriate duality pairing. If this duality pairing can be realized as an integral over Γ we have

$$dJ(\Gamma; V) = \int_{\Gamma} \nabla J v_n d\Gamma, \quad (28)$$

and we are able to use a gradient method by choosing $v_n = -\nabla J$; otherwise some presmoothing is necessary.

In the level set framework [15, 16, 18], the two-dimensional domain Ω_i is represented as the set of points where a three dimensional function ϕ defined over Ω is negative. It is also assumed that $\phi(\Omega \setminus \overline{\Omega}_i) > 0$ with ϕ Lipschitz continuous. Since we have not only one but a collection of domains $\Omega_i, i \in \{1, \dots, m\}$, it is straightforward to see that k level set functions allow for 2^k domains. Therefore k can be chosen as the closest integer value greater than or equal to $\frac{\ln m}{\ln 2}$. For instance, with $m = 4$ we get $k = 2$. For the sake of simplicity, we restrict ourselves to the two-dimensional case and $m = 4$ in what follows. Therefore, we define two level sets functions ϕ_1 and ϕ_2 and

the sets D_1 and D_2 such that

$$D_1 := \Omega_1 \cup \Omega_2 = \{x \in \Omega \mid \phi_1(x) < 0\}, \quad (29)$$

$$D_2 := \Omega_1 \cup \Omega_3 = \{x \in \Omega \mid \phi_2(x) < 0\}. \quad (30)$$

Then, for instance, the set Ω_1 can be deduced by

$$\Omega_1 = \{x \in \Omega \mid \phi_1(x) < 0 \text{ and } \phi_2(x) < 0\}.$$

There exists a simple relation between the evolution of a level set function ϕ and the vector field $V(t, x)$, which corresponds to the moving boundary Γ^t . Actually, the level set function ϕ is the solution of the Hamilton-Jacobi equation

$$\phi_t(t, x) + v_n^{ext}(t, x)|\nabla\phi(t, x)| = 0 \quad \text{for } (t, x) \in [0, T] \times \Omega, \quad (31)$$

$$\phi(0, x) = \phi_0(x) \quad \text{for } x \in \Omega \quad (32)$$

where ϕ_t stands for the time derivative of ϕ and ϕ_0 is given initial data. For example, ϕ_0 may correspond to the signed distance function of the initial contour Γ^0 . Note that v_n is defined only on Γ , therefore it is necessary to define an extension v_n^{ext} to the entire domain. This extension is used in (31). There are several ways to achieve such an extension; see, for instance, [9].

3.1 Shape sensitivity analysis

Now we return to our initial multiphase formulation (2). For convenience, we use the notation

$$\mathcal{J}_\nu(\Gamma) = \mathcal{J}_\nu(\{\Omega_i\}_{i \in [1, m]}).$$

Using the calculus developed in [20] and assuming that $\Gamma_i \cap \Gamma_j$, $i \neq j$, is sufficiently smooth, the shape derivative of $\mathcal{J}_\nu(\Gamma)$ in the direction of the vector field V is given by

$$\begin{aligned} d\mathcal{J}_\nu(\Gamma; V) = & \sum_{i=1}^m 2 \int_{\Omega_i} (f(x) - c_i) c'_i(\Gamma; V) dx \\ & + \sum_{i=1}^m \sum_{j \neq i}^m \left(\int_{\Gamma_i \cap \Gamma_j} ((f(x) - c_i)^2 + \frac{\nu}{2} \kappa_i(x)) v_{n_i}(x) dx \right), \end{aligned}$$

where $\Gamma_i := \partial\Omega_i$ and κ_i is the curvature of Γ_i , n_i is the outer unit normal vector to Ω_i , and $c'_i(\Gamma; V)$ denotes the shape derivative of c_i at Γ in the direction V . Actually, since $c'_i(\Gamma; V)$ is a scalar, we have

$$\sum_{i=1}^m \int_{\Omega_i} (f(x) - c_i) c'_i(\Gamma; V) dx = \sum_{i=1}^m c'_i(\Gamma; V) \int_{\Omega_i} (f(x) - c_i) dx = 0,$$

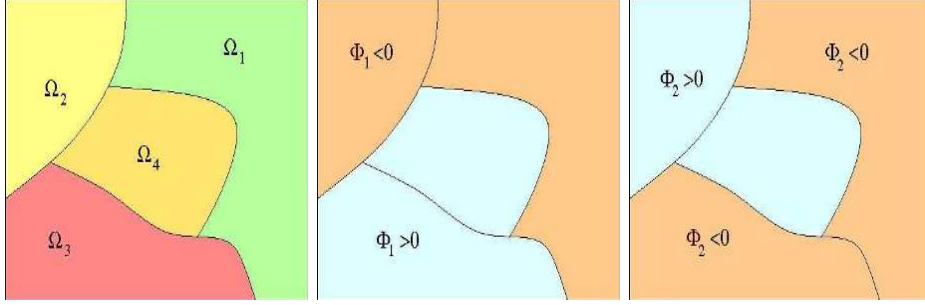


Figure 2: Partition of Ω and level set functions ϕ_1 and ϕ_2 .

as c_i is the mean value of f over the domain Ω_i . Thus, we obtain

$$d\mathcal{J}_\nu(\Gamma; V) = \frac{1}{2} \sum_{i=1}^m \sum_{j \neq i}^m \int_{\Gamma_i \cap \Gamma_j} ((f(x) - c_i)^2 - (f(x) - c_j)^2 + \nu \kappa_i(x)) v_{n_i}(x) dx. \quad (33)$$

Therefore, we identify the shape gradient as in (28) by

$$\nabla \mathcal{J}_\nu(x) = (f(x) - c_i)^2 - (f(x) - c_j)^2 + \frac{\nu}{2} \kappa_i(x) \text{ a.e. on } \Gamma_i \cap \Gamma_j$$

for $i, j = 1, \dots, m$ with $j \neq i$. Extending this expression to Γ_i , $i = 1, \dots, m$, we define the steepest descent flow by

$$v_{n_i}(x) = -(f(x) - c_i)^2 + (f(x) - c_j)^2 - \frac{\nu}{2} \kappa_i(x) \text{ a.e. on } \Gamma_i; \quad (34)$$

see [1] for a related expression.

The normal velocity field in (34) is defined on Γ_i , but for the purpose of solving the Hamilton-Jacobi equation, we need to define it on the boundaries of D_1 and D_2 . We must take into account the fact that the intersections of the Γ_i , $i \in \{1, \dots, m\}$, are non-empty. From the definition of D_1 we get

$$\begin{aligned} \partial D_1 \setminus \partial \Omega &= (\Gamma_1 \cup \Gamma_2) \cap (\Gamma_3 \cup \Gamma_4) \\ &= (\Gamma_1 \cap \Gamma_3) \cup (\Gamma_1 \cap \Gamma_4) \cup (\Gamma_2 \cap \Gamma_3) \cup (\Gamma_2 \cap \Gamma_4). \end{aligned}$$

From (34), and in view of $\kappa_i = -\kappa_j$ on $\Gamma_i \cap \Gamma_j$ for $i, j \in \{1, \dots, m\}$, we deduce the values of the velocity fields on the boundaries of D_1 and D_2 , respectively, as

$$v_{n_{D_1}} = -(f - c_{12})^2 + (f - c_{34})^2 - \nu \kappa_{D_1} \text{ a.e. on } \partial D_1, \quad (35)$$

$$v_{n_{D_2}} = -(f - c_{13})^2 + (f - c_{24})^2 - \nu \kappa_{D_2} \text{ a.e. on } \partial D_2, \quad (36)$$

where c_{ij} designates the piecewise constant function

$$\begin{aligned} c_{ij}(x) &= c_i \text{ if } x \in \Gamma_i, \\ c_{ij}(x) &= c_j \text{ if } x \in \Gamma_j, \end{aligned}$$

and κ_{D_1} , κ_{D_2} are the curvatures of D_1 and D_2 , respectively.

3.2 Algorithm for shape derivative

The starting point of our shape optimization algorithm is the numerical solution obtained after running algorithm 2. In what follows, the superscript (l) refers to the l -th iterate of the discrete counterpart of the respective continuous variable. In a similar way as for the algorithm for topological derivative we stop the algorithm as soon as

$$\max(\|V_1^{(l)}\|_2, \|V_2^{(l)}\|_2) \leq \mu_s^1(1 + \max(\|V_1^{(0)}\|_2, \|V_2^{(0)}\|_2)),$$

or

$$|\mathcal{J}_\nu(\Gamma^{(l+1)}) - \mathcal{J}_\nu(\Gamma^{(l)})| \leq \mu_s^2(1 + \mathcal{J}_\nu(\Gamma^{(0)}))$$

where $0 < \mu_s^1, \mu_s^2 \ll 1$ are user-specified stopping tolerances.

step 1 Initialize by choosing $\phi_1^{(0)}$ and $\phi_2^{(0)}$ as the signed distance functions to $\Omega_1^{(0)} \cup \Omega_2^{(0)}$ and $\Omega_1^{(0)} \cup \Omega_3^{(0)}$, respectively, so that (29)-(30) is satisfied. Here, the sets $\Omega_i^{(0)}, i \in \{1, \dots, m\}$, come from Algorithm 1; set $l = 0$.

step 2 Compute the normal velocities $v_{n,1}^{(l)}$ and $v_{n,2}^{(l)}$ for $\phi_1^{(l)}$ and $\phi_2^{(l)}$ according to (35)-(36). If $\|v_{n,1}^{(l)}\| = 0$ and $\|v_{n,2}^{(l)}\| = 0$, then stop; otherwise continue with step 3.

step 3 Extend the normal velocity $v_{n,1}^{(l)}$ and $v_{n,2}^{(l)}$ to the whole domain Ω as described in (57), and update the level set functions $\phi_1^{(l)}$ and $\phi_2^{(l)}$ by solving the Hamilton-Jacobi equation (31).

step 4 Update the domains $\Omega_i^{(l)}, i \in \{1, \dots, m\}$ according to (29)-(30) and put $l = l + 1$. Go to step 2.

As before, in our numerical realization of the above scheme we replace $\|\cdot\|_2$ and $\mu_s^j, j = 1, 2$, by a discrete L_2 -norm and $\mu_s^j h$, respectively.

A line search procedure is performed to modify the time step in the discrete Hamilton-Jacobi equation; see (51). This is done by multiplying the time step by $\alpha > 0$, which is determined such that a Armijo-type descent criterion is satisfied. In the iterative procedure below, one chooses some initial $\alpha_0 > 0$ and reduces this value by multiplying by $\beta \in (0, 1)$, if necessary. For the description of the lines search, let $\lambda > 0$ be a given parameter and $0 < \alpha_s \ll 1$ denote a lower bound on α . Then the line search procedure of iteration l is as follows:

step 1 If $\mathcal{J}_\nu(\Gamma^{(l+1)}) - \mathcal{J}_\nu(\Gamma^{(l)}) \leq -\lambda\alpha \max(\|V_1^{(l)}\|_2, \|V_2^{(l)}\|_2)$ then stop the line search and set $\alpha = \min(2\alpha, \alpha_0)$; else go to step 2.

step 2 Set $\alpha := \beta\alpha$. If $\alpha < \alpha_s$ then stop the algorithm.

step 3 Compute the new boundary $\Gamma^{(l+1)}$.

step 4 Compute the new value $\mathcal{J}_\nu(\Gamma^{(l+1)})$. Go to step 1.

Finally we mention that in our numerics we use a narrow-band version of our level-set based shape optimization algorithm.

4 Numerics

Now, we present different segmentation results. For topological sensitivity all results were obtained by algorithm 2. The number m of levels is fixed to $m = 4$ in all of our examples. The domain Ω is a square containing n^2 pixels. For each example, the number of pixels and iterations, respectively, and the CPU-time in seconds are presented. In the line search procedure discussed above we use the parameter values $\beta = 0.5$ and $\lambda = 10^{-2}$. In our tests, the initial step size α_0 is chosen as $\alpha_0 = 10^{-4}$. The constants for the stopping criteria described in the algorithms are chosen as $\mu_s^1 = 5.10^2$, $\mu_s^2 = 10$, $\alpha_s = 10^{-8}$. The regularization parameter for the length of the contour is $\nu = 200$ in all examples. All computations were performed on a standard desktop PC (Intel 3.20 GHz CPU with 2GB of RAM).

4.1 Numerical results

Example 1. The first example is an image of a plane; see figure 3. The topology optimization is extremely fast. It produces an excellent segmentation, but the boundaries of the Ω_i 's still require the application of the shape sensitivity step in order to account for the perimeter term in the Mumford-Shah functional. We point out that the shape optimization step usually is slower than the topology optimization phase. This is due to the additional work in the time-stepping of the Hamilton-Jacobi equation for advancing ϕ and the redistancing; see Table 1.

Table 1

n^2	Iterations topo	Time topo	Iterations shape	Time shape
390^2	11	0.11s	6	11.28s

Example 2. The second example is an image of a lamp; see figure 4. In this case, conclusions as for the first example can be drawn; see table 2 for the results.

Table 2

n^2	Iterations topo	Time topo	Iterations shape	Time shape
315^2	14	0.09s	3	4.05s

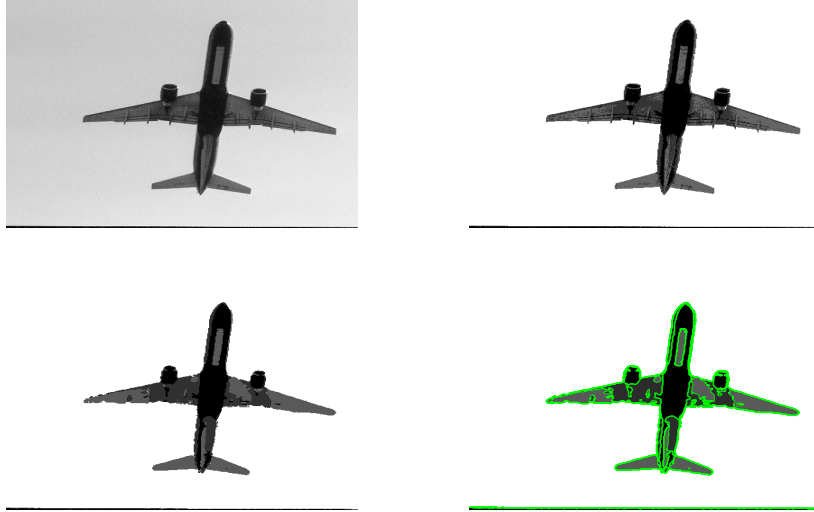


Figure 3: Example 1. Original image (upper left), image after topology step (upper right), segmentation without contour (lower left), segmentation with contour in green (lower right).



Figure 4: Example 2. Original image (upper left), image after topology step (upper right), segmentation without contour (lower left), segmentation with contour in green (lower right).

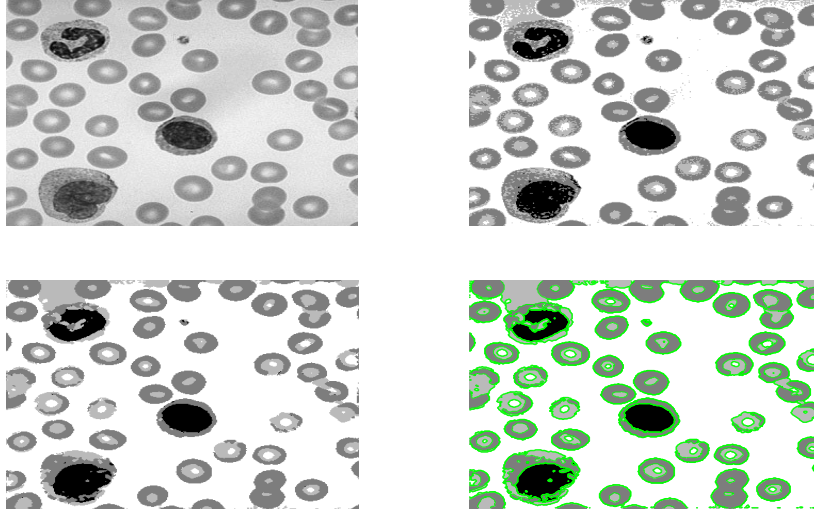


Figure 5: Example 3. Original image (upper left), image after topology step (upper right), segmentation without contour (lower left), segmentation with contour in green (lower right).

Example 3. As a third example we segment an image of blood cells; see figure 5. Again, the conclusions are similar as for example 1.

Table 3

n^2	Iterations topo	Time topo	Iterations shape	Time shape
331^2	35	0.26s	6	17.83s

Note that compared to the previous two examples the number of iterations increases which we may attribute to the higher image complexity.

4.2 Comparison with the discrete topological algorithm [7].

Next we compare our results with the algorithm presented in [7], which relies on a discrete notion of topological sensitivity. The method proceeds as follows: First, a sequence of pixels is chosen, e.g., from the upper left corner to the lower right corner of the image. Then, according to the discrete topological sensitivity at a pixel of the sequence, this pixel might get moved to a different segment. If the pixel was moved, then the level set function and the value of the phases are updated. In order to deal with the perimeter term in the Mumford-Shah functional, it is necessary to include a preprocessing or postprocessing step in the algorithm. However, we are interested here in the part of the algorithm which deals with the topological sensitivity. Hence, we only compare this part with ours.

In figure 6, we show the original image (127×127 pixels), which has to be segmented, on the left and the result obtained by our algorithm on the right. In figure 7, one can see two results for the algorithm of [7] obtained from two different initializations. These initializations are plotted in the left column, and in the right column the corresponding final segmentation is shown. For the first initialization we get the correct segmentation. But for the second initialization we find that the disk inside the square on the upper left of the image was not detected.

Apparently the algorithm in [7] is very sensitive to its initialization. If the initialization gives a good guess of the position of the different objects in the image, then the algorithm is able to find the correct segmentation; otherwise it may miss important features of the images. In contrast to this, our method utilizes an automated initialization which produces the correct segmentation of the image. But even if we force our algorithm to use different initializations (such as the two tested above), it always produces the correct segmentation.

Moreover, the CPU-time consumption of our method is significantly less compared to the one of the method in [7]. Counting floating point operations (FLOPs) per iteration (and ignoring arithmetic logic units–ALUs), we find that our algorithm requires $2N + 3m - 2$ FLOPs, where $N = n^2$ denotes the number of image pixels. Since, typically, $m \ll N$ (e.g., $m = 4$ for two-dimensional image data), the overall complexity is approximately $2N$. On the other hand, the method in [7] requires approximately $28N$ FLOPs per iteration. This yields a ratio of 14 in favor of our method. We remark that, in addition, [7] requires slightly more ALUs and floating point multiplications per iteration when compared with our method. This difference in complexity could be observed in our numerics, as well. For instance, for example 2 (see figure 4) our method requires 14 iterations and the algorithm of [7] 12 to reach the same segmentation result. In this case our method was approximately 15 times faster.

4.3 Coarse to fine grid technique

The speed of our topological derivative method can be increased by using a very simple coarse to fine grid strategy. The image is coarsened by taking one pixel of two in the two directions, with rounding off when we have an odd number of pixels. In the numerical example 3 for instance, we have $n = 331$ and, thus, we consider a sequence of coarse images with $n = 165, 82, 41, 20$ and 10. Then we apply the topological derivative on the coarsest image, interpolate the result on the next finer grid and use it as the initial guess on the next level, until we reach the finest mesh. Below, we report on the iterations per level and the CPU-time improvement under mesh refinement. The segmentation result is identical to the one obtained before. Note that we improve the computational time by more than 50%.

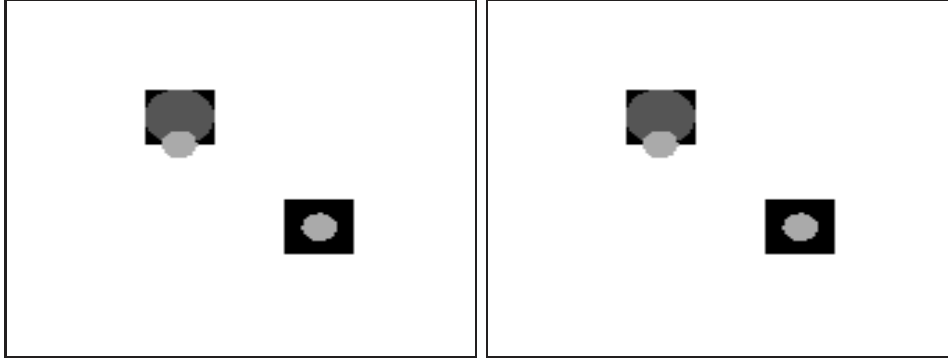


Figure 6: Original image (left), final segmentation with our algorithm (right).

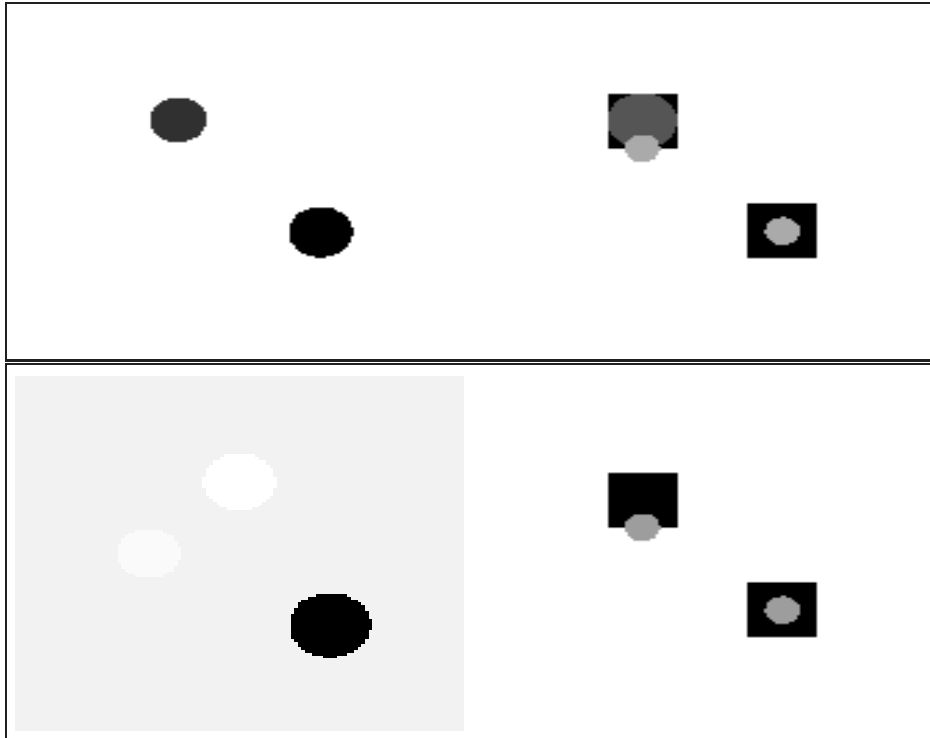


Figure 7: First initialization (upper left), final segmentation for the algorithm in [7] (upper right), Second initialization (lower left), final segmentation for the algorithm in [7] (lower right).

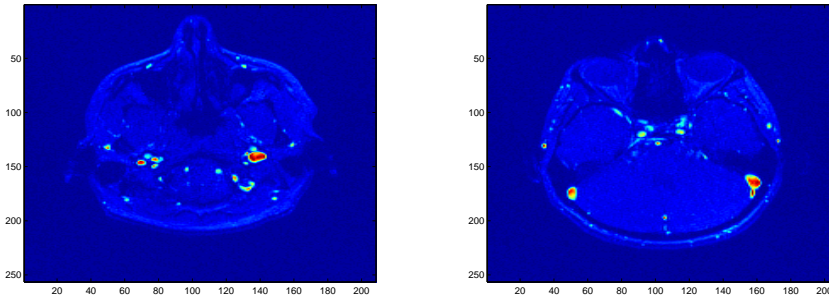


Figure 8: 2D-slices of 3D data of the brain.

n^2	10^2	20^2	41^2	82^2	165^2	331^2
Iterations topo	13	10	25	22	14	12
Improvement	> 50%					

4.4 3D segmentation

Algorithm 2 (topological derivative) is applied to a 3D-image showing blood vessels in the brain. The image size is $256 \times 208 \times 70$. We choose 4 levels (phases) for the segmentation. The algorithm converged after 30 iterations in 4.93 seconds. In figure 8 we show two slices of the image data stack. The segmentation result is depicted in figure 9. Notice that we only plot two phases for visualization purposes. One can clearly identify the vessel structure from our result.

5 Modulation recovery and segmentation

Now we turn to the more general version of the Mumford-Shah functional introduced in (4). We recall that we assume scaled image intensities satisfying $f \in [0, 1]$. Hence, we also require $u \in [0, 1]$. This together with the modulation constraints $0 \leq \sigma \leq \bar{\sigma}$ in Ω motivates the following approximation of (4):

$$\begin{aligned}
 J_\nu^{(1)}(u, \Gamma, \sigma) &= \int_\Omega (f - \sigma u)^2 + \delta \int_\Omega |\nabla^p \sigma|^2 + \mu \int_{\Omega \setminus \Gamma} |\nabla u|^2 + \nu \mathcal{H}^1(\Gamma) \\
 &\quad + \kappa \int_\Omega \max(u - 1, 0)^2 - \lambda \int_\Omega (\ln(\sigma) + \ln(\bar{\sigma} - \sigma)), \quad (37)
 \end{aligned}$$

with $\kappa > 0$ and $\lambda > 0$. Note that the max-term in $J_\nu^{(1)}$ penalizes violations of $u \leq 1$ through the associated penalty parameter κ . We shall see later that the non-negativity of u arises automatically in our context. In a slight

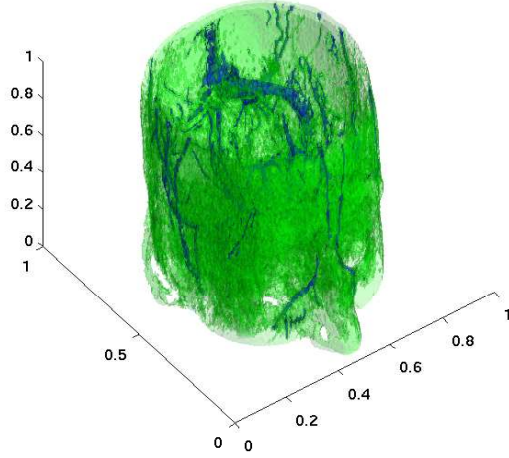


Figure 9: 3D segmentation using the topological derivative algorithm

deviation from usual tensor notation, $\nabla^2\sigma$ (i.e., $p = 2$ in (37)) denotes the Hessian of σ (rather than the Laplacian) with $|\nabla^2\sigma|$ the Frobenius norm of $\nabla^2\sigma$. The ln-terms involving σ follow an interior point philosophy [22] and guarantee that $0 < \sigma < \bar{\sigma}$ for $\lambda > 0$. Eventually we are interested in $\kappa \rightarrow \infty$ and $\lambda \rightarrow 0$ as this guarantees $0 \leq u \leq 1$ and $0 \leq \sigma \leq \bar{\sigma}$. We also introduce an alternative version of $J_\nu^{(1)}$ which is given by

$$\begin{aligned}
J_\nu^{(2)}(u, \Gamma, \sigma) = & \int_{\Omega} (\sigma^{-1}f - u)^2 + \delta \int_{\Omega} |\nabla^p\sigma|^2 + \mu \int_{\Omega \setminus \Gamma} |\nabla u|^2 + \nu \mathcal{H}^1(\Gamma) \\
& + \kappa \int_{\Omega} \max(u - 1, 0)^2 - \lambda \int_{\Omega} (\ln(\sigma) + \ln(\bar{\sigma} - \sigma)). \quad (38)
\end{aligned}$$

Although (38) is quite non-linear with respect to σ , in our numerical tests we found that (38) outperforms a version of this functional involving $\nabla^p\sigma^{-1}$ rather than $\nabla^p\sigma$ in the associated regularization term (and corresponding modifications of the ln-terms). Indeed, the modulation may get close to zero rapidly in certain image regions, and thus the minimization of the norm of $\nabla^p\sigma^{-1}$ produces an adverse effect which must be avoided. The segmentation and modulation recovery problems we are interested in, thus, consist in minimizing either

$$\min_{u, \Gamma, \sigma} J_\nu^{(1)}(u, \Gamma, \sigma) \quad \text{or} \quad \min_{u, \Gamma, \sigma} J_\nu^{(2)}(u, \Gamma, \sigma). \quad (39)$$

Observe that in the piecewise constant context we have

$$\begin{aligned} J_\nu^{(1)}(u, \Gamma, \sigma) &= \int_\Omega (f - \sigma u)^2 + \delta \int_\Omega |\nabla^p \sigma|^2 + \nu \mathcal{H}^1(\Gamma) \\ &\quad + \kappa \int_\Omega \max(u - 1, 0)^2 - \lambda \int_\Omega (\ln(\sigma) + \ln(\bar{\sigma} - \sigma)), \end{aligned}$$

and analogously for $J_\nu^{(2)}$. Moreover, u is given by (3).

Before we commence with technical details, we briefly motivate the above functionals. For this purpose, note that in practice radio frequency coils are used in magnetic resonance imaging (MRI) for both nuclear excitation and for signal detection. Large homogeneous coils may be used for the uniform elucidation of a volume but a smaller surface coil can be used to resolve local details with greater sensitivity. However surface coils suffer from a nonuniform sensitivity in relation to that of a body coil. This results in a modulated image where the modulation is unknown. We refer to [10] for more details. In order to restore images with such a degradation, the purpose of the functionals (37) and (38) is to find simultaneously the unknown modulation σ and the contour Γ as well as the piecewise constant reconstruction u .

The possible values for the regularization parameter p in (37) and (38) lead to different reconstructions of the modulation. Indeed, in the case $p = 1$, the necessary optimality conditions for σ result in the resolution of a second-order partial differential equation. Fixing σ on parts of the boundary and using a maximum principle one may disregard the logarithmic terms in (37) and (38), as the nonnegativity and boundedness of σ are automatically guaranteed. However, such boundary conditions typically assume some pre-knowledge of the position of the coil, which is not always the case and is, in particular, not assumed throughout this paper. By choosing $p = 2$ we allow more "freedom" in the possible reconstruction of σ as the kernel of the associated differential operator contains harmonic functions as opposed to constant functions for Neumann boundary conditions in the case $p = 1$. Of course, $p = 2$ induces higher regularity of σ than $p = 1$. In practical applications, however, rather high regularity of σ appears to be the case, and therefore it may not be considered a restriction. But, for $p = 2$, we need to take care of $\sigma \in [0, \bar{\sigma}]$ for some sufficiently large $\bar{\sigma} > 0$. In our approach this is achieved by the logarithmic terms in (37) and (38), respectively, which reflect an interior point treatment of the constraints; see [22] for a general discussion of interior point methods in optimization. Higher order regularization could also be considered, at the expense of the amount of computations.

In what follows, we focus on the case $p = 2$. We also consider only the application of a slightly modified version of the topology optimization algorithm 1. Therefore, we disregard the $\mathcal{H}^1(\Gamma)$ -term by choosing $\nu = 0$. In

the piecewise constant setting, (37) becomes

$$\begin{aligned}
J_\nu^{(1)}(u, \Gamma, \sigma) &= \int_{\Omega} (f - \sigma u)^2 + \kappa \int_{\Omega} \max(u - 1, 0)^2 \\
&\quad + \delta \int_{\Omega} |\nabla^2 \sigma|^2 - \lambda \int_{\Omega} (\ln(\sigma) + \ln(\bar{\sigma} - \sigma)) \\
&= \sum_{i=1}^m \int_{\Omega_i} (f - \sigma c_i)^2 + \kappa \sum_{i=1}^m |\Omega_i| \max(c_i - 1, 0)^2 \\
&\quad + \delta \int_{\Omega} |\nabla^2 \sigma|^2 - \lambda \int_{\Omega} (\ln(\sigma) + \ln(\bar{\sigma} - \sigma)),
\end{aligned}$$

In our approach, the minimization of $J_\nu^{(1)}(u, \Gamma, \sigma)$ is performed in two steps:

- (i) Minimize $J_\nu^{(1)}$ with respect to σ and keep u and Γ fixed.
- (ii) Minimize $J_\nu^{(1)}$ with respect to u and Γ and keep σ fixed.

Step (ii) relies on a modified version of algorithm 1. The involved modifications concern the formulas for the topological derivatives and for c_i , $i \in \{1, \dots, m\}$. The relevant details are given below in this section.

We emphasize that the primary goal of this section is to recover the image by detecting the modulation, and not the segmentation of the image. Still, the application of algorithm 1 is primordial for the de-modulation procedure. Once an (approximately) demodulated image has been achieved, the segmentation procedure described in the previous sections can be applied.

5.1 Minimization with respect to σ

The first order necessary optimality conditions for the minimization of (37) with respect to σ are

$$\begin{aligned}
(\delta \Delta^2 + u^2)\sigma - \frac{\lambda}{2\sigma} + \frac{\lambda}{2(\bar{\sigma} - \sigma)} &= uf \quad \text{in } \Omega, \\
\partial_{nn}\sigma = \partial_{n\tau}\sigma = \partial_n \Delta \sigma &= 0 \quad \text{on } \Gamma,
\end{aligned} \tag{40}$$

where ∂_n stands for the normal derivative and ∂_τ the tangential derivative on the boundary Γ . We are interested in applying Newton's method for solving (40). Hence, we need the corresponding linearization of (40). For this purpose we set

$$\underline{\lambda} = \frac{\lambda}{2\sigma}, \tag{41}$$

$$\bar{\lambda} = \frac{\lambda}{2(\bar{\sigma} - \sigma)}, \tag{42}$$

$$A(u) = \delta \Delta^2 + u^2 \text{id},$$

where id represents the identity operator. The linearization of (41)-(42) and (40) gives

$$\underline{\lambda}d\sigma + \sigma d\underline{\lambda} = \frac{\lambda}{2} - \underline{\lambda}\sigma, \quad (43)$$

$$-\bar{\lambda}d\sigma + (\bar{\sigma} - \sigma)d\bar{\lambda} = \frac{\lambda}{2} - \bar{\lambda}(\bar{\sigma} - \sigma), \quad (44)$$

$$A(u)d\sigma - d\underline{\lambda} + d\bar{\lambda} = uf - A(u)\sigma + \underline{\lambda} - \bar{\lambda}. \quad (45)$$

From this system we obtain

$$d\underline{\lambda} = -\frac{\lambda}{\sigma}d\sigma + \frac{\lambda}{2\sigma} - \underline{\lambda}, \quad (46)$$

$$d\bar{\lambda} = \frac{\bar{\lambda}}{\bar{\sigma} - \sigma}d\sigma + \frac{\lambda}{2(\bar{\sigma} - \sigma)} - \bar{\lambda}, \quad (47)$$

$$\left[A(u) + \left(\frac{\lambda}{\sigma} + \frac{\bar{\lambda}}{\bar{\sigma} - \sigma} \right) \text{id} \right] d\sigma = uf - A(u)\sigma + \frac{\lambda}{2\sigma} - \frac{\lambda}{2(\bar{\sigma} - \sigma)}. \quad (48)$$

Therefore, for obtaining the update direction $d\sigma$ we solve

$$\begin{aligned} \left[A(u) + \left(\frac{\lambda}{\sigma} + \frac{\bar{\lambda}}{\bar{\sigma} - \sigma} \right) \text{id} \right] d\sigma &= uf - A(u)\sigma + \frac{\lambda}{2\sigma} - \frac{\lambda}{2(\bar{\sigma} - \sigma)}, \\ \partial_{nn}d\sigma = \partial_{n\tau}d\sigma = \partial_n\Delta d\sigma &= 0. \end{aligned} \quad (49)$$

Then $d\underline{\lambda}$ and $d\bar{\lambda}$ are deduced from (46)-(47). Next we compute the step length α such that

$$\bar{\sigma} \geq \sigma + \alpha d\sigma \geq 0, \quad \underline{\lambda} + \alpha d\underline{\lambda} \geq 0, \quad \bar{\lambda} + \alpha d\bar{\lambda} \geq 0.$$

In order to stay safely in the interior of $[0, \bar{\sigma}]$ we choose

$$\alpha_\sigma = \min \left\{ 0.99 \min \left\{ -\frac{\sigma}{d\sigma} \mid d\sigma < 0 \right\}; 0.99 \min \left\{ \frac{\bar{\sigma} - \sigma}{d\sigma} \mid d\sigma > 0 \right\} \right\}$$

and, for $z \in \{\underline{\lambda}, \bar{\lambda}\}$, $\alpha_z = 1$ if $dz \geq 0$, and $\alpha_z = 0.99 \min\{-\frac{z}{dz} \mid dz < 0\}$, otherwise. Then we set $\alpha = \min\{1, \alpha_\sigma, \alpha_{\underline{\lambda}}, \alpha_{\bar{\lambda}}\}$. Finally we update $\sigma, \underline{\lambda}$ and $\bar{\lambda}$ in iteration k of Newton's method for solving (40) by

$$\begin{aligned} z^{k+1} &= z^k + \alpha_k dz, \quad z \in \{\sigma, \underline{\lambda}, \bar{\lambda}\}, \\ \lambda^{k+1} &= \gamma \lambda^k \end{aligned}$$

where $\gamma \in]0, 1[$ is a constant that is chosen beforehand. The modulation σ is initialized as a positive constant. For instance, we use $\sigma \equiv \bar{\sigma}/2$.

5.2 Minimization with respect to u

In this case we minimize the function (38) instead of (37). Note that due to the logarithms in the functional (38) we have $\sigma > 0$. The necessary optimality conditions for c_i , $i \in \{1, \dots, m\}$, is

$$2\kappa|\Omega_i| \max(c_i - 1, 0) + \int_{\Omega_i} 2 \left(c_i - \frac{f}{\sigma} \right) = 0$$

This leads to the two cases

$$c_i = |\Omega_i|^{-1} \int_{\Omega_i} \frac{f}{\sigma} \text{ if } c_i \leq 1,$$

and

$$c_i = \frac{\kappa + |\Omega_i|^{-1} \int_{\Omega_i} \frac{f}{\sigma}}{1 + \kappa} \text{ if } c_i > 1.$$

Note that in both situations $c_i \geq 0$ for all i .

5.3 Minimization with respect to the shape

In the previous sections, the formulas for the topological derivative are given by (13) and (14), depending on whether Ω_j is empty or not. In the case of the functional (38), we obtain a similar formula. If $|\Omega_j| \neq 0$, we get

$$\begin{aligned} \mathcal{T}_{i,j}(x_0) &= \left(\frac{f(x_0)}{\sigma(x_0)} - c_j \right) \left(\frac{f(x_0)}{\sigma(x_0)} - c_j - \max(c_j - 1, 0) \frac{2\kappa|\Omega_j|}{(\kappa + |\Omega_j|)^2} \right) \\ &\quad - \left(\frac{f(x_0)}{\sigma(x_0)} - c_i \right) \left(\frac{f(x_0)}{\sigma(x_0)} - c_i - \max(c_i - 1, 0) \frac{2\kappa|\Omega_i|}{(\kappa + |\Omega_i|)^2} \right). \end{aligned}$$

When $|\Omega_j| = 0$ we obtain

$$\mathcal{T}_{i,j}(x_0) = - \left(\frac{f(x_0)}{\sigma(x_0)} - c_i \right) \left(\frac{f(x_0)}{\sigma(x_0)} - c_i - \max(c_i - 1, 0) \frac{2\kappa|\Omega_i|}{(\kappa + |\Omega_i|)^2} \right).$$

5.4 Numerical results

Numerical results for simultaneous modulation recovery and segmentation are presented in figure 10–12. In the left column of each figure we depict the original image in the first row, the piecewise constant Mumford-Shah-based reconstruction in the middle and the modulation which is applied to the original in the last row. The modulated image is shown in the first row of the right column. The reconstruction obtained by our simultaneous segmentation and modulation recovery scheme and the reconstructed modulation are shown in the second and third row of the right column, respectively.

In the first example (see figure 10) the size of the image is 100×100 pixels. The applied modulation is $\sigma = x_1^2 + x_2^2$, where (x_1, x_2) are the

cartesian coordinates in the square Ω . The segmentation obtained after the reconstruction of the modulation σ is very close to the original segmentation, as one can see from the middle row of figure 10. The parameter δ for the regularization of σ is $\delta = 6.10^{-6}$.

In the second example (see figure 11) the image size and the applied modulation are as in the previous example. In the segmentation of the reconstructed image a global brightening effect occurs, which is due to an overestimation of the modulation σ . However, all the features of the original image are present in the reconstruction. The parameter δ is equal to $\delta = 1.10^{-6}$.

In the third example (see figure 12) the size of the image is 100×100 pixels and the modulation is $\sigma = \exp(x_1 x_2) - 1$. The result is still satisfying although the exponential modulation is more difficult to reconstruct. The parameter δ is equal to $\delta = 1.10^{-6}$.

6 Conclusion

Numerical results show the efficiency of our topology optimization based modulation recovery and image segmentation algorithm. The scheme requires no particular initialization to obtain an excellent segmentation result. Therefore this technique is completely automatized. The part based on topological sensitivity is highly efficient with respect to CPU-time. We also emphasize that the algorithm for the topological derivative can be easily parallelized, which could result in a dramatic reduction in CPU-time consumption.

7 Appendix

7.1 The level set method

Now we describe the numerical solution of the level set equation (31) and the computation of the extension of v_n to the whole domain Ω . Let ϕ be a level set function, i.e.,

$$\Omega = \{x \mid \phi(x) < 0\}, \quad \partial\Omega = \{x \mid \phi(x) = 0\}. \quad (50)$$

We introduce the nodes P_{ij} , whose coordinates are given by $(i\Delta x, j\Delta y)$ where Δx and Δy are the discretization steps in the x -direction and y -direction, respectively. Let us also denote by $t^k = k\Delta t$ the discrete time for $k \in \mathbb{N}$, where Δt is the time step. We are seeking for an approximation $\phi_{ij}^k \simeq \phi(P_{ij}, t^k)$. Following Osher and Sethian [15, 16], we use the explicit upwind scheme

$$\phi_{ij}^{k+1} = \phi_{ij}^k - \Delta t g(D_-^x \phi_{ij}^k, D_+^x \phi_{ij}^k, D_-^y \phi_{ij}^k, D_+^y \phi_{ij}^k) \quad (51)$$

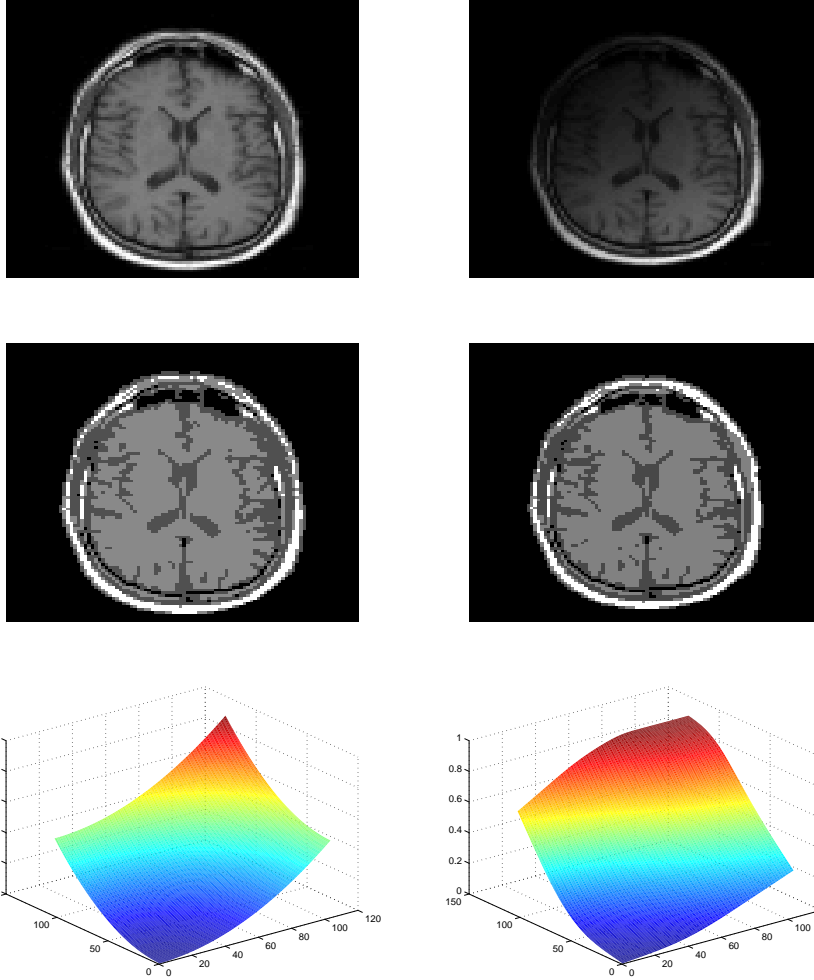


Figure 10: Original image (upper left), modulated image (upper right), segmentation of original image (middle left), segmentation of modulated image (middle right), original modulation (lower left), reconstructed modulation (lower right).

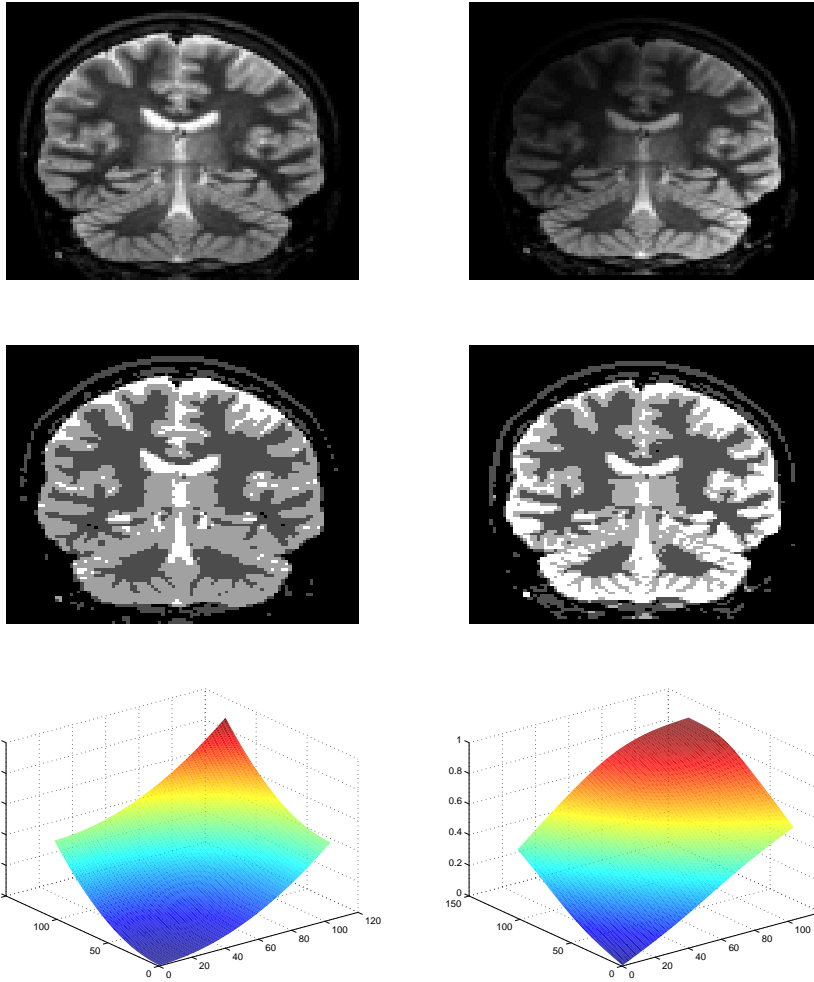


Figure 11: Original image (upper left), modulated image (upper right), segmentation of original image (middle left), segmentation of modulated image (middle right), original modulation (lower left), reconstructed modulation (lower right).

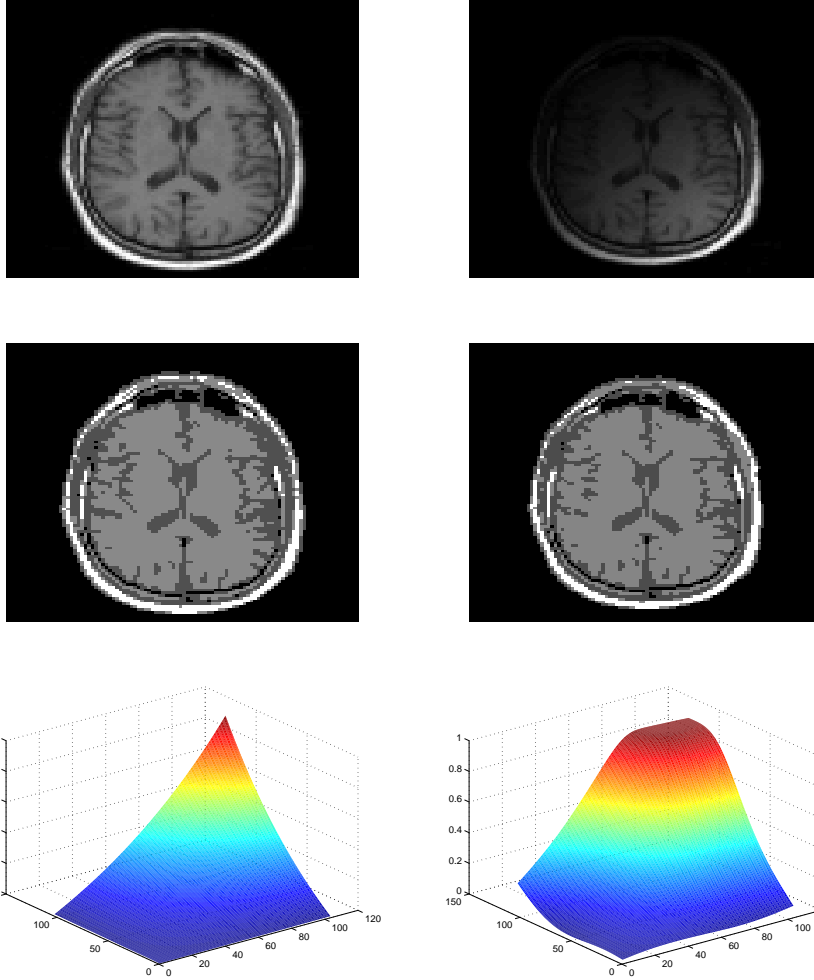


Figure 12: Original image (upper left), modulated image (upper right), segmentation of original image (middle left), segmentation of modulated image (middle right), original modulation (lower left), reconstructed modulation (lower right).

where

$$D_-^x \phi_{ij} = \frac{\phi_{ij} - \phi_{i-1,j}}{\Delta x} \quad \text{and} \quad D_+^x \phi_{ij} = \frac{\phi_{i+1,j} - \phi_{ij}}{\Delta x} \quad (52)$$

are the backward and forward approximations of the x -derivative of ϕ at P_{ij} . Similarly we obtain D_-^y and D_+^y of the y -derivative. The numerical flux is given by

$$g(D_-^x \phi_{ij}, D_+^x \phi_{ij}, D_-^y \phi_{ij}, D_+^y \phi_{ij}) = \max(v_{ij}, 0) G^+ + \min(v_{ij}, 0) G^-$$

with

$$G^+ = \left[\max(D_-^x \phi_{ij}, 0)^2 + \min(D_+^x \phi_{ij}, 0)^2 + \max(D_-^y \phi_{ij}, 0)^2 + \min(D_+^y \phi_{ij}, 0)^2 \right]^{1/2},$$

$$G^- = \left[\min(D_-^x \phi_{ij}, 0)^2 + \max(D_+^x \phi_{ij}, 0)^2 + \min(D_-^y \phi_{ij}, 0)^2 + \max(D_+^y \phi_{ij}, 0)^2 \right]^{1/2}$$

and $v_{ij} = \langle V_{\text{ext}}, n \rangle(P_{ij})$ the extended normal velocity at the point P_{ij} as defined in (57) below. This upwind scheme is stable under the CFL condition

$$\left(\max_{\Omega} |\langle V_{\text{ext}}, n \rangle| \right) \Delta t \left(\frac{1}{\Delta x} + \frac{1}{\Delta y} \right) \leq \frac{1}{2}. \quad (53)$$

For numerical purposes, the solution ϕ of the level set equation should not be too flat or too steep. Hence, an instance of ϕ , which is numerically stable, is the signed distance function which satisfies $|\nabla \phi| = 1$. Unfortunately, even if the initial data ϕ_0 is given by a signed distance function, the solution ϕ of the level set equation need not remain close to a distance function, in general. To overcome this difficulty, we perform a reinitialization of ϕ at time t by computing the stationary state $\varphi_{\infty}(x) = \lim_{\tau \rightarrow \infty} \varphi(\tau, x)$ of the following equation (see [17]):

$$\varphi_{\tau} + S(\phi)(|\nabla \varphi| - 1) = 0 \quad \text{in } \mathbb{R}^+ \times \Omega, \quad (54)$$

$$\varphi(0, x) = \phi(t, x) \quad \text{for } x \in \Omega. \quad (55)$$

Here, S is an approximation of the sign function, i.e.,

$$S(d) = \frac{d}{\sqrt{d^2 + |\nabla d|^2 \delta^2}} \quad (56)$$

with $\delta = \min(\Delta x, \Delta y)$. Of course, other choices are possible; see [17] for details.

Next we describe the construction of the extension of the normal velocity to the whole domain. Besides the need of a velocity in Ω in the level set equation, another purpose of the extension is to enforce ϕ to remain a signed distance function. Indeed, if we are able to compute an extended normal velocity V_{ext} such that

$$\nabla V_{\text{ext}} \cdot \nabla \phi = 0 \quad \text{in } \mathbb{R}^+ \times \Omega, \quad (57)$$

then it can be shown (see [23]) that the solution ϕ of the level set equation satisfies $|\nabla\phi| = 1$. One way to construct the extension V_{ext} satisfying (57) is to solve the following equation up to stationary state (see [15, 17]):

$$q_\tau + S(\phi) \frac{\nabla\phi}{|\nabla\phi|} \cdot \nabla q = 0 \text{ in } \mathbb{R}^+ \times \Omega, \quad (58)$$

$$q(0, x) = q_0(x), \quad x \in \Omega. \quad (59)$$

Then we take $V_{\text{ext}}(x) = \lim_{\tau \rightarrow \infty} q(\tau, x)$. At each iteration k of the previous scheme, we compute the extended normal velocity as the stationary solution of (58)-(59). Then $q_{ij}^n \simeq q(P_{ij}, t^n)$ comes from the following upwind approximation of (58) :

$$q_{ij}^{n+1} = q_{ij}^n - \Delta\tau \left[\max(s_{ij}n_{ij}^x, 0) D_-^x q_{ij} + \min(s_{ij}n_{ij}^x, 0) D_+^x q_{ij} + \max(s_{ij}n_{ij}^y, 0) D_-^y q_{ij} + \min(s_{ij}n_{ij}^y, 0) D_+^y q_{ij} \right], \quad (60)$$

where $s_{ij} = S(\phi_{ij}^n)$. We use central differences to compute the approximation n_{ij} of the unit normal vector

$$n = (n^x, n^y) = \left(\phi_x / \sqrt{\phi_x^2 + \phi_y^2}, \phi_y / \sqrt{\phi_x^2 + \phi_y^2} \right)$$

at the node P_{ij} . The initial value q_0 is equal to v_n on the grid points with a distance less than $\min(\Delta x, \Delta y)$ to the interface and it is zero elsewhere.

Acknowledgement. Both authors are indebted to the anonymous referees for their valuable input which helped to improve the paper. The authors would also like to thank S.L. Keeling (University of Graz) for discussions and for providing a Matlab solver of the fourth-order equation, R. Stollberger and F. Knoll (TU Graz) for providing the 3D medical image data. The authors further acknowledge financial support by the Austrian Ministry of Science and Education and the Austrian Science Foundation FWF under START-grant Y305 "Interfaces and free boundaries" and the subproject "Freelevel" of the SFB F32 "Mathematical Optimization and Applications in Biomedical Sciences".

References

- [1] T. F. CHAN AND L. A. VESE, *A level set algorithm for minimizing the Mumford-Shah functional in image processing*, in: Proceedings of the IEEE Workshop on Variational and Level Set Methods (VLSM '01), pp. 161-168, Vancouver, BC, Canada, July 2001.
- [2] M. DELFOUR AND J.-P. ZOLESIO, *Shapes and Geometries. Analysis, Differential Calculus and Optimization*, SIAM Advances in Design and Control, SIAM, Philadelphia, 2001.

- [3] M. GIAQUINTA, *Introduction to Regularity Theory for Nonlinear Elliptic Systems*, Birkhäuser, Basel–Boston–Berlin, 1993.
- [4] W. HACKBUSCH, *Elliptic Differential Equations*, vol. 18 of Springer Series in Computational Mathematics, Springer Verlag, Berlin, 1992.
- [5] J.A. HARTIGAN, *Clustering Algorithms*, Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, New York-London-Sydney, 1975.
- [6] J.A. HARTIGAN AND M.A. WONG, *A k-means clustering algorithm*, Journal of the Royal Statistical Society (Series C), Applied Statistics, 28 (1979), pp.100-108.
- [7] 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, pp. 777-788, Springer Verlag, Berlin-New York-Heidelberg, 2007.
- [8] A. HENROT AND M. PIERRE, *Variation et Optimisation de Formes: Une Analyse Géométrique*, No. 48 de Mathématiques et Applications, Springer, Berlin-New York-Heidelberg, 2005.
- [9] M. HINTERMÜLLER AND W. RING, *A second-order shape optimization approach for image segmentation*. SIAM J. Appl. Math., 64 (2003), pp. 442-467.
- [10] S.L. KEELING AND R. BAMMER, *A variational approach to magnetic resonance coil sensitivity estimation*, Applied Mathematics and Computation, 158 (2004), pp. 53-82.
- [11] J. MACQUEEN, *Some methods for classification and analysis of multivariate observations*, 1967 Proc. Fifth Berkeley Sympos. Math. Statist. and Probability (Berkeley, Calif., 1965/66) Vol. I: Statistics, pp. 281–297 Univ. California Press, Berkeley, California.
- [12] V. MAZ'YA, S.A. NAZAROV AND B. PLAMENEVSKIJ, *Asymptotic Theory of Elliptic Boundary Value Problems in Singularly Perturbed Domains* Vol. 1 and 2, Birkhäuser, Basel, 2000.
- [13] J.-M. MOREL AND S. SOLIMINI, *Variational Methods in Image Segmentation*, Progress in Nonlinear Differential Equations and their Applications, 14, Birkhäuser Boston Inc., Boston, MA, 1995.
- [14] D. MUMFORD AND J. SHAH *Optimal approximations by piece-wise smooth functions and associated variational problems*, Commun. Pure Appl. Math., 42 (1989), pp. 577-685.

- [15] S. OSHER AND R. FEDKIW, *Level Set Methods and Dynamic Implicit Surfaces*, Springer, Berlin-New York-Heidelberg, 2004.
- [16] S. OSHER AND J. SETHIAN, *Fronts propagating with curvature-dependent speed : algorithms based on Hamilton-Jacobi formulation*, J. Comp. Phys., 79 (1988), pp. 12-49.
- [17] D. PENG, B. MERRIMAN, S. OSHER, H. ZHAO, AND M. KANG, *A PDE-based fast local level set method*, J. Comp. Phys., 155 (1999), pp. 410-438.
- [18] J.A. SETHIAN, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, second edition, Cambridge, 1999.
- [19] J. SOKOŁOWSKI AND A. ŻOCHOWSKI *On the topological derivative in shape optimization*, SIAM J. Control Optim., 37 (1999), pp. 1251-1272.
- [20] J. SOKOŁOWSKI AND J.-P. ZOLESIO, *Introduction to Shape Optimization*, vol. 16 of Springer Series in Computational Mathematics, Springer, Berlin-New York-Heidelberg, 1992.
- [21] 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 (2002), pp. 271-293.
- [22] S.J. WRIGHT, *Primal-Dual Interior-Point Methods*, SIAM Publications, Philadelphia, PA, 1997.
- [23] H.K. ZHAO, T. CHAN, B. MERRIMAN, AND S. OSHER, *A variational level set approach to multi-phase motion*, J. Comp. Phys., 122 (1996), pp. 179-195.
- [24] W.P. ZIEMER, *Weakly Differentiable Functions*, Graduate Texts in Mathematics, Springer, New York, 1989.