



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

Convex non-convex segmentation of scalar fields over arbitrary triangulated surfaces

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:

Convex non-convex segmentation of scalar fields over arbitrary triangulated surfaces / Huska, Martin; Lanza, Alessandro; Morigi, Serena; Sgallari, Fiorella. - In: JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS. - ISSN 0377-0427. - STAMPA. - 349:(2019), pp. 438-451. [10.1016/j.cam.2018.06.048]

Availability:

This version is available at: <https://hdl.handle.net/11585/642215> since: 2019-02-26

Published:

DOI: <http://doi.org/10.1016/j.cam.2018.06.048>

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).
When citing, please refer to the published version.

(Article begins on next page)

This is the final peer-reviewed accepted manuscript of:

Huska, M., et al. "Convex Non-Convex Segmentation of Scalar Fields Over Arbitrary Triangulated Surfaces." *Journal of Computational and Applied Mathematics*, vol. 349, 2019, pp. 438-451.

The final published version is available online at :
<http://dx.doi.org/10.1016/j.cam.2018.06.048>

Rights / License:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>)

When citing, please refer to the published version.

CONVEX NON-CONVEX SEGMENTATION OF SCALAR FIELDS OVER ARBITRARY TRIANGULATED SURFACES

MARTIN HUSKA*, ALESSANDRO LANZA†, SERENA MORIGI‡, AND FIORELLA SGALLARI§

Abstract. An extension of the Mumford–Shah model for image segmentation is introduced to segment real-valued functions having values on a complete, connected, 2-manifold embedded in \mathbb{R}^3 . The proposed approach consists of three stages: first, a multi-phase piecewise smooth partition function is computed, then its values are clustered and, finally, the curve tracking is computed on the segmented boundaries. The first stage, which constitutes the key novelty behind our proposal, relies on a Convex Non-Convex variational model where an ad-hoc non-convex regularization term coupled with a space-variant regularization parameter allows to effectively deal with both the boundaries and the inner parts of the segments. The cost functional is minimized by means of an efficient numerical scheme based on the Alternating Directions Methods of Multipliers. Experimental results are presented which demonstrate the effectiveness of the proposed three-stage segmentation approach.

1. Introduction. Image segmentation is a hot topic of research due to its applicability as a pre-processing technique in many image understanding applications. In order to facilitate the extraction of high-level information and to improve the performance of classification and object recognition algorithms, the segmentation preliminary process aims to decompose an image into K disjoint regions that are “homogeneous” according to a certain feature such as intensity or texture, to identify more meaningful high level information in the image. One of the most widely studied segmentation models in computer vision and in image processing fields is the Mumford and Shah model [10] which addresses the complex objectives of detecting both edges and smooth regions. The Mumford–Shah model approaches the problem of segmentation from a variational viewpoint by defining an optimization problem based on a functional involving a piecewise smooth representation of an image and a set of the discontinuities of the image.

Many applications ranging from 3D medical imaging, QR/barcode readers, or even the feature identification and tracking in sequences of textured triangular surfaces, rely on the segmentation of scalar fields printed, painted or etched onto 3D object surfaces. The even more challenging segmentation task must faces, in these cases, the partitioning of non-flat images, or, in general, of scalar fields defined on an arbitrary topology 2-manifold \mathcal{M} embedded in \mathbb{R}^3 .

Since the 2-manifold is not restricted to be a parametric manifold, but rather a more generic arbitrary topology manifold, the well-known successful methods for image segmentation [2, 3, 10] can not be directly exploited. Popular variational formulations for image segmentation in the Euclidean space \mathbb{R}^n , have been adapted and generalized on surfaces to take into account the geometry of the manifold itself. The segmentation model first introduced by Chan and Vese [3] for images has been proposed for surfaces in [16] and [15]. However, in [15] the number of parts has to be chosen a priori, and the model relies on a TV penalty which is less sparsity promoting with respect to the non-convex penalty here proposed.

In this work we propose a generalization of the Mumford–Shah image segmentation model for the segmentation of scalar-valued features on manifolds. In particular, a generic real-valued function $f : \mathcal{M} \rightarrow \mathbb{R}$ defined on a complete, connected 2-manifold \mathcal{M} infers a decomposition of \mathcal{M} in salient parts representing global or local textural information. The Mumford–Shah functional involves three terms designed to meet three different and competing requirements that arise quite naturally from the intuitive concept of segmentation: (1) fidelity to the data; (2) discontinuity preserving smoothing; (3) discontinuity detection and control to prevent over-segmentation. The

*Department of Mathematics, University of Bologna, Bologna, Italy.
E-mail: martin.huska@unibo.it

†Department of Mathematics, University of Bologna, Bologna, Italy.
E-mail: alessandro.lanza2@unibo.it

‡Department of Mathematics, University of Bologna, Bologna, Italy.
E-mail: serena.morigi@unibo.it

§Department of Mathematics, University of Bologna, Bologna, Italy.
E-mail: fiorella.sgallari@unibo.it

basic step of the proposed framework requires the minimization of the following variational model:

$$\mathcal{J}(u; \lambda, \eta, a) := \frac{\lambda}{2} \int_{\mathcal{M}} (u - f)^2 d\mathcal{M} + \frac{1}{2} \int_{\mathcal{M}} \eta(|\nabla f|) |\nabla u|^2 d\mathcal{M} + \int_{\mathcal{M}} (1 - \eta(|\nabla f|)) \phi(|\nabla u|; a) d\mathcal{M}, \quad (1.1)$$

where ∇u is the intrinsic (Riemannian) gradient defined on \mathcal{M} and $d\mathcal{M}$ is the manifolds element measure, while $|\cdot|$ is the Riemannian norm.

The function $u : \mathcal{M} \rightarrow \mathbb{R}$ is the manifold-valued partition function, and represents a smooth approximation of the given function f . The parameter $\lambda > 0$ is a given regularization parameter which, together with the edge-detecting function $\eta(\cdot) \geq 0$, control the relative strength of the three terms. In particular, we consider the bounded nonnegative continuous and monotonically descending function

$$\eta(t) = 1 / (1 + (t/\kappa)^2), \quad (1.2)$$

which have been widely used in planar image processing. The role of $\eta(\cdot)$ in discontinuity detection is crucial since the detection of the regions boundaries is as relevant as the detection of the regions themselves, therefore, the boundaries should be preserved from being smoothed.

The function $\phi(\cdot; a) : [0, +\infty) \rightarrow \mathbb{R}$ in (1.1) is a parameterized, non-convex penalty function with parameter $a \geq 0$, which controls the degree of non-convexity and will be referred to as the *concavity parameter*. For a values approaching to zero, $\phi(\cdot; a)$ behaves as a total variation regularization term, for other a values $\phi(\cdot; a)$ is a sparsity-inducing penalty which well preserves sharp boundaries while penalizing their length and avoiding staircases effects.

The functional is composed by the sum of smooth convex (quadratic) terms and a non-smooth non-convex regularization term designed for penalizing simultaneously the non-smoothness of the inner regions and the length of the segmented boundaries. Thus the functional \mathcal{J} in (1.1) is non-smooth and can be convex or non-convex depending on the values of $\lambda, \eta(\cdot)$ and a . Although the definition of the functional can be easily understood, as it mimics the original Mumford–Shah non-convex model, non-trivial difficulties arise in the mathematical treatment and in the numerical solution of such a minimum problem.

This work presents a discretization of (1.1) on discretized "manifold", represented as arbitrary triangular mesh, and the numerical treatment of the Convex Non-Convex (CNC) functional \mathcal{J} for the solution of the optimization problem (1.1).

The idea of constructing and then optimizing convex functionals containing non-convex (sparsity-promoting) terms, referred to as Convex Non-Convex approach, dates back to the seminal works in [1], and [11] for CNC image denoising; very interesting developments have been presented by Selesnick and others for different purposes, see [4, 7, 8, 9] for more details. The attractiveness of such CNC approach resides in its ability to promote sparsity more strongly than it is possible by using only convex terms while at the same time maintaining convexity of the total optimization problem, so that well-known reliable convex minimization approaches can be used to compute the (unique) solution.

In this paper, we propose a three-stage variational segmentation method inspired by the piecewise smoothing proposal in [2] which is a convex variant of the classical Mumford–Shah model. In the first stage an approximate solution u^* to the optimization problem (1.1) is computed. Once u^* is obtained, then in the second stage the segmentation is done by thresholding u^* into different parts. The thresholds can be given by the users or can be obtained automatically using any clustering methods, such as for example the K-means algorithm. As discussed in [2], this allows for a K-phase segmentation ($K \geq 2$) by choosing $(K - 1)$ thresholds after u^* is computed in the first stage. In contrast, many multiphase methods require K to be given in advance which implies that if K changes, the minimization problem has to be solved again. Finally, a contour track phase is computed to extract boundary curves delimiting the segmented regions on the manifold.

Summarizing, this work provides three main contributions.

- We formulate a CNC variational model for the segmentation of single-valued functions over triangulated surfaces.

- Theoretical conditions are derived to ensure that the functional \mathcal{J} in problem (1.1) is convex – despite the regularization term being nonconvex. The inclusion of the manifold’s geometry is done by characterizing the convexity parameter a locally. Therefore the model (1.1) has a unique global minimizer.
- A fast iterative algorithm to approximate faithfully the minimizer of the CNC partitioning functional is proposed which is based on the Alternating Directions Method of Multipliers (ADMM) optimization strategy.

This work extends the preliminary investigation reported in [6]. The proposed variational segmentation model presents a new space-variant regularization parameter which allows to effectively deal with both the boundaries and the inner parts of the segments, similarly to the Mumford–Shah model. A new more accurate discretization on meshes rather than on graphs is presented which leads to a more geometrically faithful numerical scheme.

The outline of the paper is as follows. In Section 2 we characterize the non-convex penalty functions $\phi(\cdot; a)$ considered in the above model. In Section 3 the discretization of the CNC functional in (1.1) is provided and sufficient condition for the strict convexity are derived. In Section 4 details of the Alternating Directions Methods of Multipliers (ADMM)-based numerical algorithm used to compute approximate solutions of (1.1) are described. Results in Section 5 clearly shows the quality and the practical convenience of variational segmentation. Conclusion are drawn in Section 6.

2. Non-convex penalty functions. In this section we characterize the non-convex penalty functions $\phi(\cdot; a)$ used in (1.1). We denote the sets of non-negative and positive real numbers as $\mathbb{R}_+ := \{t \in \mathbb{R} : t \geq 0\}$ and $\mathbb{R}_+^* := \{t \in \mathbb{R} : t > 0\}$, respectively. Analogously to [4, 7, 8], we consider parameterized penalty functions $\phi(t; a) : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ such that for any value of the parameter $a \in \mathbb{R}_+$ the following assumptions are satisfied:

- A1) $\phi(t; a) \in \mathcal{C}^2(\mathbb{R}_+)$ (ϕ twice continuously differentiable in t on \mathbb{R}_+)
- A2) $\phi'(t; a) > 0 \quad \forall t \in \mathbb{R}_+$ (ϕ strictly increasing in t on \mathbb{R}_+)
- A3) $\phi''(t; a) \leq 0 \quad \forall t \in \mathbb{R}_+$ (ϕ concave in t on \mathbb{R}_+)
- A4) $\phi(0; a) = 0, \phi'(0^+; a) = 1,$ (ϕ, ϕ', ϕ'' normalization)
 $\inf_{t \in \mathbb{R}_+} \phi''(t; a) = \phi''(0^+; a) = -a.$

We remark that a represents a scalar indicator of the “degree of concavity” of the penalty function ϕ , thus justifying the name *concavity parameter*. Moreover, we set $\phi(t; 0) := t$, such that the ℓ_1 -norm penalty is recovered as a special case of $\phi(\cdot; a)$ when $a = 0$. We refer to [8, 4] for a detailed discussion on commonly used penalty functions satisfying assumptions A1)–A4) above, such as

$$\phi_{\log}(t; a) = \frac{\log(1 + at)}{a}, \quad \phi_{\text{rat}}(t; a) = \frac{t}{1 + at/2}, \quad \phi_{\text{atan}}(t; a) = \frac{\text{atan}\left(\frac{1+2at}{\sqrt{3}}\right) - \frac{\pi}{6}}{a\sqrt{3}/2}. \quad (2.1)$$

In Figure 2.1 we show the plots of these penalty functions, for three different values $a \in \{0.2, 2, 4\}$ of the concavity parameter.

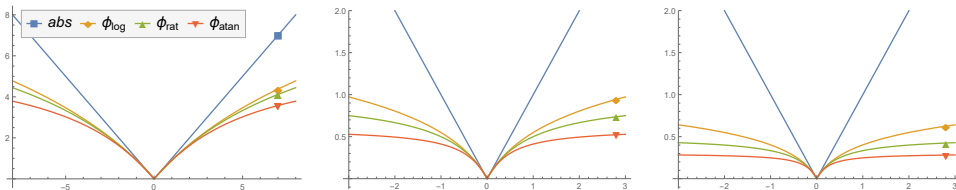


FIG. 2.1. Plots of the penalty functions $\phi_{\log}(t; a)$, $\phi_{\text{rat}}(t; a)$, $\phi_{\text{atan}}(t; a)$ defined in (2.1), for different values of the concavity parameter a : $a = 0.2$ (left), $a = 2$ (center), $a = 4$ (right).

3. Model discretization and convexity analysis. The numerical solution of variational problems on surfaces involves the definition of the discrete manifold that represents the under-

lying object of interest, and the discrete approximation of the first order operators of the weak formulation in (1.1).

We assume Ω is a triangulated surface of arbitrary topology approximating a 2-manifold \mathcal{M} embedded in \mathbb{R}^3 and defined by a mesh $\Omega := (V, T)$, where $V = \{X_i\}_{i=1}^n \in \mathbb{R}^{n \times 3}$ is the set of vertices, and $T \in \mathbb{N}^{n_T \times 3}$ is the set of face triangles.

The direct connection between the neighbors X_i and X_j is referred as edge e_{ij} and thus the whole set $E = \{e_{ij}\} \subseteq V \times V$. For each vertex X_i , we denote by $N(X_i) = \{X_j : e_{ij} \in E\}$ its 1-ring neighborhood, defined as the set of incident vertices to X_i , by N_i the corresponding ordered set of vertex indices, and by $D(X_i)$ the 1-disk of the vertex X_i . Then $v_i = |N(X_i)|$ is the so-called valence of vertex X_i . Similarly, by $N_\Delta(X_i)$ we denote the triangle neighbourhood around X_i and $|N_\Delta(X_i)|$ is the area of these neighbouring triangles. Under this discrete setting, the scalar functions f, u defined on \mathcal{M} can be sampled over the vertices V of the mesh Ω , $f, u : \Omega \rightarrow \mathbb{R}$, and are understood as piecewise linear functions.

Denoting by $f_i := f(X_i)$, and analogously by $u_i := u(X_i)$, $i = 1, \dots, n$, the sampled functions are represented by vectors $u, f \in \mathbb{R}^n$. The evaluation of a vertex-based piecewise linear function u at a location x inside a triangle $\tau = [x_i, x_j, x_k] \in T$, is thus given as:

$$u(x)|_\tau = u_i B_i(x) + u_j B_j(x) + u_k B_k(x), \quad (3.1)$$

where $\{B_\ell : \ell = 1, \dots, n\}$ represent the corresponding piecewise linear basis function over τ such that $B_\ell(x_j) = \delta_{\ell j}$, $\ell, j = 1, \dots, n$, which satisfy the following properties

1. local support: $\text{supp } B_\ell = D(X_\ell)$;
2. nonnegativity: $B_\ell \geq 0$
3. partition of unity: $\sum_\ell B_\ell \equiv 1$.

By differentiating these vertex-based bases, we obtain a piecewise constant (face-based) tangent vector field. The intrinsic gradient ∇u of u in (1.1) lies in the tangent space $T_X \mathcal{M} \subset \mathbb{R}^{3 \times n_T}$, and restricted on τ is therefore expressed as

$$\nabla u(x)|_\tau = u_i \nabla B_i(x) + u_j \nabla B_j(x) + u_k \nabla B_k(x) . \quad (3.2)$$

From the equation (3.2), the corresponding basis function gradients are constant and can be expressed as defined in [14]

$$\nabla B_i = \frac{x_i - O_i}{|x_i - O_i|^2} = \frac{1}{|x_i - O_i|} \frac{x_i - O_i}{|x_i - O_i|} = \frac{1}{|h_i|} \frac{h_i}{|h_i|} \quad (3.3)$$

where O_i is the orthogonal projection of x_i onto (x_j, x_k) , thus, h_i is the triangle's height w.r.t. x_i , and $\frac{h_i}{|h_i|}$ is a unit vector in direction of the triangle's height. For each vertex $x_i \in \tau$, the corresponding projection O_i can be expressed as

$$O_i = x_j + \frac{e_{ji} \cdot e_{jk}}{|e_{jk}|} e_{jk} .$$

Using vector-matrix notation, the gradient operator (3.2) can be locally discretized and represented by a matrix $G = (\nabla B_i, \nabla B_j, \nabla B_k)$ as follows

$$\nabla u(x)|_\tau = Gu = \begin{pmatrix} \nabla B_i^1 & \nabla B_j^1 & \nabla B_k^1 \\ \nabla B_i^2 & \nabla B_j^2 & \nabla B_k^2 \\ \nabla B_i^3 & \nabla B_j^3 & \nabla B_k^3 \end{pmatrix} \begin{pmatrix} u_i \\ u_j \\ u_k \end{pmatrix}, \quad (3.4)$$

from which the gradient magnitude can be approximated as follows

$$\|\nabla u(x)|_\tau\|_2^2 = \langle Gu, Gu \rangle = u^T G^T G u = u^T Q_\tau u \quad (3.5)$$

where Q_τ is defined as

$$Q_\tau = \begin{pmatrix} \|\nabla B_i\|_2^2 & \langle \nabla B_i, \nabla B_j \rangle & \langle \nabla B_i, \nabla B_k \rangle \\ \langle \nabla B_j, \nabla B_i \rangle & \|\nabla B_j\|_2^2 & \langle \nabla B_j, \nabla B_k \rangle \\ \langle \nabla B_k, \nabla B_i \rangle & \langle \nabla B_k, \nabla B_j \rangle & \|\nabla B_k\|_2^2 \end{pmatrix}. \quad (3.6)$$

The matrix elements in Q_τ can be computed as follows

$$\|\nabla B_i\|_2^2 = \frac{1}{|h_i|^2}, \quad \langle \nabla B_i, \nabla B_j \rangle = \frac{\cos \angle(h_i, h_j)}{|h_i||h_j|},$$

and similarly for the other elements.

To derive the discrete formulation of $\mathcal{J}(\cdot)$ in (1.1) we first recall that the integral of a function f over the whole \mathcal{M} represented by the triangular mesh Ω is simply the sum over all of the triangles τ , which reads as

$$\int_{\mathcal{M}} f(x) d\mathcal{M} = \sum_{\tau=1}^{n_\tau} \int_{\tau} f(x) d\tau. \quad (3.7)$$

Now we focus on a triangle τ with d nodes, thus the quadrature approximates an integral via a discrete weighted sum of function values over the set of nodes

$$\int_{\tau} f(x) d\tau \approx \sum_{i=1}^d w_i f_i. \quad (3.8)$$

In case of piecewise constant interpolant functions ($d = 1$) $w = s_\tau$ for the centroid, for piecewise linear interpolants ($d = 3$) then $w_i = \frac{s_\tau}{3}, i = 1, 2, 3$ for the three vertices, while for piecewise quadratic interpolants, one obtains a quadrature that is exact for all polynomials of degree $p \leq 3$, and the weights are: $w = s_\tau 27/60$ for the 1 centroid, $w = s_\tau 1/20$ for the 3 vertices, $w = s_\tau 2/15$ for the 3 edge midpoints. For piecewise linear function f thus (3.8) with $d = 3$ leads to the exact quadrature

$$\int_{\mathcal{M}} f(x) d\mathcal{M} = \sum_{\tau=[X_i, X_j, X_k]} \frac{s_\tau}{3} (f_i + f_j + f_k) = \sum_{i=1}^n f_i s_i, \quad (3.9)$$

where we used the fact that $s_i = \frac{|N_\Delta(X_i)|}{3} = \sum_{j \in N(X_i)} \frac{s_{\tau_j}}{3}$. However, if f is a quadratic function,

thus the sum on the right hand side of (3.9) is only an approximation to the integral on the left.

The integral (1.1) over the whole mesh Ω is simply the sum over all of the triangles $\tau_j = [x_0, x_1, x_2] \in T$ as

$$\mathcal{J}(u; \lambda, \eta; a) = \sum_{j=1}^{n_\tau} \left[\frac{\lambda s_{\tau_j}}{6} \sum_{k=0}^2 (u_k - f_k)^2 + \frac{\eta_j s_{\tau_j}}{2} \|(\nabla u)_j\|_2^2 + s_{\tau_j} (1 - \eta_j) \phi(\|(\nabla u)_j\|_2; a_j) \right]. \quad (3.10)$$

Since the gradient field on Ω is supposed to be constant on each face, the second and third integral terms in (1.1) are exactly evaluated by the applied quadrature rule for $d = 1$, while the first integral involving a quadratic function, according to (3.9), is only approximated. The use of an exact quadrature formula for the first term will be consider in future work since it involves more complicated convexity conditions. By plugging in (3.10) the gradient definition (3.5)-(3.6) we get

$$\mathcal{J}(u; \lambda, \eta, a) = \sum_{j=1}^{n_\tau} \left[\frac{\lambda s_{\tau_j}}{6} \sum_{k=0}^2 (u_k - f_k)^2 + \frac{\eta_j s_{\tau_j}}{2} u_{\tau_j}^T Q_{\tau_j} u_{\tau_j} + s_{\tau_j} \phi\left(\sqrt{u_{\tau_j}^T Q_{\tau_j} u_{\tau_j}}, a_j\right) \right], \quad (3.11)$$

where the local discretization scheme (3.6) has been used in (3.11), with u_{τ_j} representing the function values $(u_0, u_1, u_2)^T$ at vertices of τ, j .

In view of the convexity conditions that will be derived, the convexity parameter a in (1.1) is locally defined as a_j in (3.11) for each triangle τ_j , to take into account the manifold's geometry, while the parameter η_j is locally defined to take into account the edge-detection function.

In the remaining of this section we seek for a sufficient condition on the parameters $\lambda > 0$, $\eta, a \geq 0$, such that the objective functional $\mathcal{J}(\cdot; \lambda, \eta, a)$ in (1.1) discretized as in (3.11) is strictly convex.

In the following, we give two lemmas which allow us to reduce convexity analysis from the original functional $\mathcal{J}(\cdot; \lambda, \eta, a)$ of n_T variables to easier functions $f_\tau(\cdot; \lambda, \eta, a)$ and $g_\tau(\cdot; \lambda, \eta, a)$ of 3 and 2 variables, respectively. Then in Theorem 3.3 we finally give sufficient conditions for strict convexity of \mathcal{J} in (3.11).

LEMMA 3.1. *Let $f_\tau(\cdot; \lambda, \eta, a): \mathbb{R}^3 \rightarrow \mathbb{R}$ be the function defined by*

$$f_\tau(x_0, x_1, x_2; \lambda, \eta, a) := \frac{\lambda}{6} x^T x + \frac{\eta}{2} x^T Q_\tau x + (1 - \eta) \phi\left(\sqrt{x^T Q_\tau x}; a\right). \quad (3.12)$$

Then, the functional $\mathcal{J}(\cdot; \lambda, \eta, a): \mathbb{R}^n \rightarrow \mathbb{R}$ defined in (3.11) is strictly convex if all the functions $f_{\tau_j}(\cdot; \lambda, \eta, a_j): \mathbb{R}^3 \rightarrow \mathbb{R}$, $j = 1, \dots, n_T$, are strictly convex.

Proof. The functional \mathcal{J} in (3.11) can be rewritten in the following equivalent form:

$$\mathcal{J}(u; \lambda, \eta, a) = \mathcal{A}(u) + \sum_{j=1}^{n_T} f_{\tau_j}(u_1, u_2, u_3; \lambda, \eta_j, a_j), \quad (3.13)$$

where $\mathcal{A}(u)$ is an affine function of u and the function f_{τ_j} is defined in (3.12). We remark that the area factor s_{τ_j} multiplies the whole functional, thus, can be eliminated while as indicated the first sum in (3.10) can be rewritten as sum over triangles τ_j where each triangle constitutes to generic vertex $x_i \in \tau_j$ by one third of its area. Since the affine function $\mathcal{A}(u)$ does not affect convexity, we can conclude that the functional \mathcal{J} in (3.13) - or, equivalently, in (3.11) - is strictly convex if all the functions f_i in (3.13) are strictly convex. \square

LEMMA 3.2. *The function $f_\tau(\cdot; \lambda, \eta, a): \mathbb{R}^3 \rightarrow \mathbb{R}$ defined in (3.12) is strictly convex if the function $g_\tau(\cdot; \lambda, \eta, a): \mathbb{R}^2 \rightarrow \mathbb{R}$ defined by*

$$g_\tau(y_1, y_2; \lambda, \eta, a) = \frac{1}{2} \left(\eta + \frac{\lambda}{3\lambda_1} \right) \sum_{j=1}^2 y_j^2 + (1 - \eta) \phi\left(\sqrt{\sum_{j=1}^2 y_j^2}; a\right) \quad (3.14)$$

is strictly convex, where λ_1 is the largest eigenvalue of matrix $Q_\tau \in \mathbb{R}^{3 \times 3}$ defined in (3.6).

The proof is provided in the Appendix.

THEOREM 3.3. *Let the function $\phi(\cdot; a): \mathbb{R}_+ \rightarrow \mathbb{R}$ satisfy assumptions A1)–A4) in Section 2. Then, a sufficient condition for functional $\mathcal{J}(\cdot; \lambda, \eta, a)$ in (3.11) to be strictly convex is that the parameters (λ, η, a) satisfy:*

$$a_j < \frac{\lambda}{3\lambda_1(1 - \eta_j)} + \frac{\eta_j}{(1 - \eta_j)} \iff a_j = \tau_c \left(\frac{\lambda}{3\lambda_1(1 - \eta_j)} + \frac{\eta_j}{(1 - \eta_j)} \right), \quad \tau_c \in [0, 1), \quad (3.15)$$

for every $j \in \{1, \dots, n_T\}$.

Proof. Based on Lemmas 3.1–3.2, the functional $\mathcal{J}(\cdot; \lambda, \eta, a)$ is strictly convex if all the functions $g_i, i \in \{1, \dots, n_T\}$, defined in (3.14) are strictly convex. Then, based on Proposition 2 in [4], the statement (3.15) follows. \square

We conclude by highlighting some properties of functional \mathcal{J} in (3.11).

PROPOSITION 3.4. *Let $\phi(\cdot; a): \mathbb{R}_+ \rightarrow \mathbb{R}$ be a penalty function as defined in (2.1) and let the parameters (λ, η, a) satisfy condition (3.15). Then, the functional $\mathcal{J}(\cdot; \lambda, \eta, a)$ in (3.11) is proper, continuous, bounded from below by zero, coercive and strongly convex.*

4. Applying ADMM to the proposed CNC model. In this section, we illustrate the ADMM-based iterative algorithm used to compute minimizers of our discrete objective functional \mathcal{J} in (3.11) in case that parameters λ, η, a satisfy condition (3.15), such that \mathcal{J} is strictly convex.

To proceed with ADMM on triangular mesh surfaces, we introduce the auxiliary variable $t \in \mathbb{R}^{3 \times n_T}$, and reformulate problem (3.11) in the equivalent form:

$$\{u^*, t^*\} \leftarrow \arg \min_{u, t} \left\{ \sum_{j=1}^{n_T} \left[\frac{\lambda}{6} \sum_{k=0}^2 (u_k - f_k)^2 + \frac{\eta_j}{2} \|t_j\|_2^2 + (1 - \eta_j) \phi(\|t_j\|_2; a_j) \right] \right\} \quad (4.1)$$

$$\text{subject to : } t = Mu, \quad (4.2)$$

where the discrete gradient operator is represented by the matrix $M \in \mathbb{R}^{(3n_T) \times n}$, with $M = (M_1^T, M_2^T, M_3^T)^T$, where each sub-matrix $M_j \in \mathbb{R}^{n_T \times n}$, $j = 1, 2, 3$, represents the linear operator which simultaneously computes the j -th gradient defined component (see (3.4)). Formally we can define M_j as

$$(M_j)_{ik} = \begin{cases} \nabla B_k^j & \text{if } X_k \in \tau_i \\ 0 & \text{otherwise} \end{cases}, \quad (4.3)$$

creating highly sparse matrix of $3n_T$ non-zero elements for each M_j .

To solve problem (4.1)–(4.2), we define the augmented Lagrangian functional

$$\begin{aligned} \mathcal{L}(u, t; \rho; \lambda, \eta, a) &= \sum_{j=1}^{n_T} \left[\frac{\lambda}{6} \sum_{k=0}^2 (u_k - f_k)^2 + \frac{\eta_j}{2} \|t_j\|_2^2 + (1 - \eta_j) \phi(\|t_j\|_2; a_j) \right] \\ &\quad - \langle \rho, t - Mu \rangle + \frac{\beta}{2} \|t - Mu\|_2^2, \end{aligned} \quad (4.4)$$

where $\beta > 0$ is a scalar penalty parameter and $\rho \in \mathbb{R}^{3 \times n_T}$ is the vector of Lagrange multipliers associated with the linear constraint $t = Mu$ in (4.2). We then consider the following saddle-point problem:

$$\begin{aligned} \text{Find } (u^*, t^*; \rho^*) &\in \mathbb{R}^n \times \mathbb{R}^{3 \times n_T} \times \mathbb{R}^{3 \times n_T} \\ \text{s.t. } \mathcal{L}(u^*, t^*; \rho; \lambda, \eta, a) &\leq \mathcal{L}(u^*, t^*; \rho^*; \lambda, \eta, a) \leq \mathcal{L}(u, t; \rho^*; \lambda, \eta, a) \\ \forall (u, t; \rho) &\in \mathbb{R}^n \times \mathbb{R}^{3 \times n_T} \times \mathbb{R}^{3 \times n_T}. \end{aligned} \quad (4.5)$$

Given the previously computed (or initialized for $k = 0$) vectors $u^{(k)}$ and $\rho^{(k)}$, the k -th iteration of the proposed ADMM-based iterative scheme applied to the solution of the saddle-point problem (4.4)–(4.5) reads as follows:

$$t^{(k+1)} \leftarrow \arg \min_{t \in \mathbb{R}^{3 \times n_T}} \mathcal{L}(u^{(k)}, t; \rho^{(k)}; \lambda, \eta, a) \quad (4.6)$$

$$u^{(k+1)} \leftarrow \arg \min_{u \in \mathbb{R}^n} \mathcal{L}(u, t^{(k+1)}; \rho^{(k)}; \lambda, \eta, a) \quad (4.7)$$

$$\rho^{(k+1)} \leftarrow \rho^{(k)} - \beta (t^{(k+1)} - Mu^{(k+1)}) \quad (4.8)$$

In the following we show in detail how to solve the two minimization sub-problems (4.6) and (4.7) for the primal variables t and u , respectively.

Although the minimization sub-problems are all strictly convex and admit a unique solution, convergence of the overall ADMM algorithm is clearly not guaranteed. We postpone the analysis of convergence of the proposed ADMM scheme to a future extended version of this work.

4.1. Solving the sub-problem for t . The minimization sub-problem for t in (4.6) can be rewritten as follows:

$$t^{(k+1)} \leftarrow \arg \min_{t \in \mathbb{R}^{3 \times n_T}} \left\{ \sum_{j=1}^{n_T} \left[\frac{\eta_j}{2} \|t_j\|_2^2 + (1 - \eta_j) \phi(\|t_j\|_2; a_j) \right] + \frac{\beta}{2} \|t - r^{(k+1)}\|_2^2 \right\} \quad (4.9)$$

where constant terms have been omitted and where the vector $r^{(k+1)} \in \mathbb{R}^{3 \times n_T}$ is constant with respect to the optimization variable t and is given by:

$$r^{(k+1)} = Mu^{(k)} + \frac{1}{\beta} \rho^{(k)}. \quad (4.10)$$

The minimization problem in (4.9) rewritten in component-wise (triangle-by-triangle) form, is equivalent to the following n_T independent lower dimensional problems:

$$t_j^{(k+1)} \leftarrow \arg \min_{t_j \in \mathbb{R}^3} \left\{ (1 - \eta_j) \phi(\|t_j\|_2; a_j) + \frac{\eta_j}{2} \|t_j\|_2^2 + \frac{\beta}{2} (t_j - r_j^{(k+1)})^2 \right\}, \quad (4.11)$$

with $j = 1, \dots, n_T$, $r_j^{(k+1)} := (Mu^{(k)})_j + (\rho^{(k)})_j / \beta$ and $(Mu^{(k)})_j, (\rho^{(k)})_j \in \mathbb{R}^3$ denote the gradient and the associated vector of Lagrange multipliers at triangle τ_j , respectively.

Since we are imposing that condition (3.15) is satisfied, such that the original functional $\mathcal{J}(u; \lambda, \eta, a)$ in (1.1) is strictly convex, we aim at avoiding non-convexity of the ADMM sub-problems (4.11). In the first part of Proposition 4.1 below, we give necessary and sufficient conditions for strict convexity of the cost functions in (4.11). In particular, based on (4.15)–(4.16), we can state that the problems in (4.11) are strictly convex if and only if the following conditions hold:

$$\beta > \max_{j=1, \dots, n_T} (a_j(1 - \eta_j) - \eta_j). \quad (4.12)$$

In case conditions in (4.12) are satisfied, the unique solutions of the strictly convex problems in (4.11) can be obtained by the soft-thresholding operator defined in (4.18)–(4.19). We remark that the nonlinear equation in (4.19) can be solved up to a sufficient accuracy by very few steps of the iterative Newton method. By setting

$$\bar{\eta} = \frac{\eta}{(1 - \eta)}, \quad \bar{\beta} = \frac{\beta}{(1 - \eta)}, \quad (4.13)$$

we can rewrite (4.11) for t_j in form

$$t \leftarrow \arg \min_{t \in \mathbb{R}^3} \left\{ \frac{\bar{\eta}_j}{2} \|t\|_2^2 + \phi(\|t\|_2; a_j) + \frac{\bar{\beta}}{2} \|t - r_j\|_2^2 \right\}, \quad (4.14)$$

what allows us to use the following Proposition for the solution of this sub-problem.

PROPOSITION 4.1. *Let $\eta, a \geq 0, \beta > 0$ and $r \in \mathbb{R}^3$ be given constants, and let $\phi(\cdot; a) : \mathbb{R}_+ \rightarrow \mathbb{R}$ be a function satisfying assumptions A1)–A4) in Section 2. Then, the function*

$$\theta(z) := \phi(\|z\|_2; a) + \frac{\bar{\eta}}{2} \|z\|_2^2 + \frac{\bar{\beta}}{2} \|z - r\|_2^2, \quad z \in \mathbb{R}^3, \quad (4.15)$$

is strictly convex if and only if the following condition holds:

$$\bar{\beta} > a - \bar{\eta} \quad \Rightarrow \quad \beta > a(1 - \eta) - \eta. \quad (4.16)$$

Moreover, in case that (4.16) holds, the strictly convex minimization problem

$$\arg \min_{z \in \mathbb{R}^3} \theta(z) \quad (4.17)$$

admits the unique solution $z^ \in \mathbb{R}^3$ given by the following shrinkage operator:*

$$z^* = \xi^* r, \quad \text{with } \xi^* \in [0, 1[, \text{ and} \quad (4.18)$$

$$\begin{aligned} \text{a) } \quad & \xi^* = 0 && \text{if } \|r\|_2 \leq \frac{1}{\bar{\beta}} \\ \text{b) } \quad & \xi^* \in]0, 1[\text{ unique solution of:} && \\ & \phi'(\|r\|_2 \xi; a) + \|r\|_2 ((\bar{\eta} + \bar{\beta})\xi - \bar{\beta}) = 0 && \text{otherwise.} \end{aligned} \quad (4.19)$$

The proof is provided in the Appendix.

4.2. Solving the sub-problem for u . The minimization sub-problem for u in (4.7) can be rewritten as follows:

$$u^{(k+1)} \leftarrow \arg \min_{u \in \mathbb{R}^n} \left\{ \frac{\lambda}{2} \|u - f\|_2^2 + \langle \rho^{(k)}, Mu \rangle + \frac{\beta}{2} \|t^{(k+1)} - Mu\|_2^2 \right\}, \quad (4.20)$$

where constants have been omitted. The quadratic minimization problem (4.20) has first-order optimality conditions which lead to the following linear system:

$$(\lambda I + \beta M^T M)u = \lambda f + \beta M^T \left(t^{(k+1)} - \frac{1}{\beta} \rho^{(k)} \right). \quad (4.21)$$

Since $M^T M$ is symmetric and the ratio $\frac{\beta}{\lambda}$ is positive, the $n \times n$ coefficient matrix of the linear system (4.21) is symmetric semi pos. def. and highly sparse. Hence, (4.21) admits a unique solution obtained very efficiently by the iterative (preconditioned) conjugate gradient method.

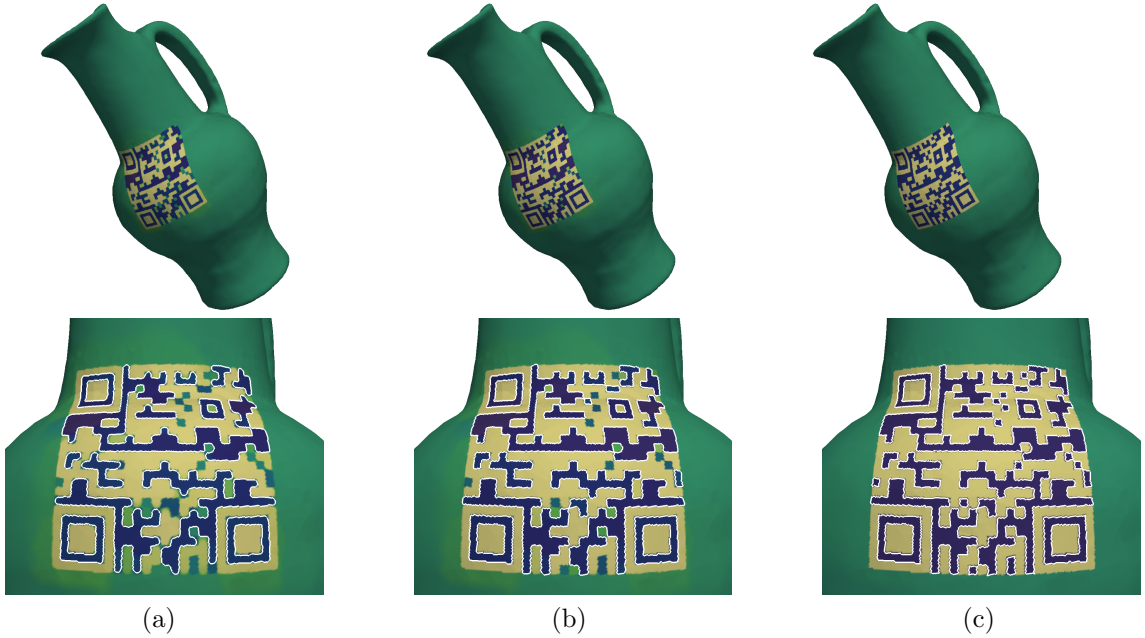


FIG. 4.1. Example 1: (a) Convex regime for $\tau_c = 0$, (b) CNC regime for $\tau_c = 0.99$, (c) Non-convex regime for $\tau_c = 10$. Boundary contours detected by the third stage of the segmentation algorithm are overlaid in white.

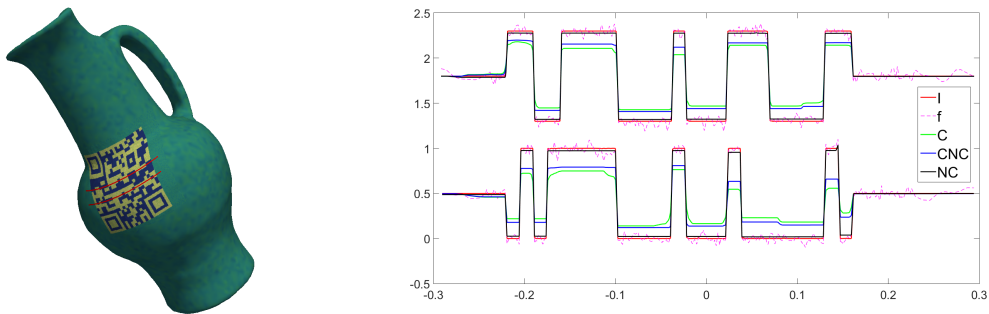


FIG. 4.2. Example 1: Plot of the values u^* along the two lines along the vase mesh (left in red). The comparison is between the original (I) and input intensity (f) w.r.t convex (C), CNC and non-convex (NC) regimes.

5. Experimental evaluation. In this section we describe the experimental results which assess the performance of the proposed segmentation algorithm where in (1.1) we used the ϕ_{\log} penalty function defined in (2.1). For what concerns the ADMM, the parameter β is chosen according to condition (4.12), and the ADMM iterations are terminated as soon as the relative change between two successive iterates satisfies

$$err := \|u^{(k)} - u^{(k-1)}\|_2 / \|u^{(k-1)}\|_2 < 10^{-4}. \quad (5.1)$$

All reported images were created by the software ParaView and its VTK reader, which natively interpolates the false colors sampled over vertices. Experimental tests were performed on Intel®Core™i7-4720HQ Quad-Core 2.6 GHz machine, with 12 GB/RAM and Nvidia GeForce GTX 860M graphics card in a Windows OS. The code is written in MATLAB language and it was executed without any additional machine support, e.g. parallelization or GPU support.

Example 1. The first example illustrates a qualitative comparison among the segmentation results obtained running the segmentation framework in convex (C), convex non-convex (CNC), and pure non-convex regime (NC). We applied the segmentation method to detect a QR code textured on a smooth manifold, a *vase*, discretized by a mesh of resolution $|V| = 52028$ and $|T| = 104056$ and corrupted by an additive white Gaussian noise with standard deviation $\sigma = 5 \times 10^{-2}$. The input is shown in Fig. 4.2(left). In particular, the convex case corresponds to the classical TV- L_2 model introduced for image denoising in [12], and is obtained by setting $\tau_c = 0$ in (3.15), the CNC model is given for $\tau_c = 0.99$ thus maximizing the non-convexity of the penalty term while preserving the overall convexity of the functional \mathcal{J} . Finally, for $\tau_c = 10$ we get a strictly non-convex model solved by the same ADMM procedure. The results u^* of the first phase of our segmentation algorithm are shown in Fig. 4.1 top row, together with the zoomed QR details (bottom row). The values of the resulting partitioning functions along the two red lines indicated in Fig. 4.2 (left) are illustrated in Fig. 4.2 (right). The original intensity values are plotted in red solid line. For the convex regime in Fig. 4.1(a) we can observe the typical behavior of TV- L_2 model: corners are smoothed and the contrast is decreased. The CNC regime in Fig. 4.1(b) presents sharper edges and the loss of contrast is decreased. The non-convex regime produces the nicest result shown in Fig. 4.1(c). In this case the algorithm converges very slowly to the given threshold within the maximum number of iterations 400. Results in Fig. 4.1 indicate that higher quality restorations can be achieved by pushing model (1.1) beyond its convexity limits, however the numerical convergence is affected by the non-convexity of the functional \mathcal{J} . This confirms the usefulness of the proposed CNC approach.

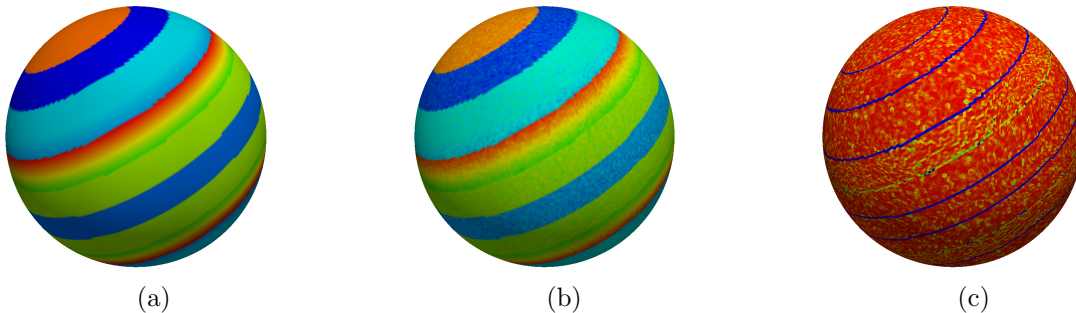


FIG. 5.1. *Example 2: (a) original map over sphere mesh $|V| = 65538$, $|T| = 131072$, (b) input corrupted image by white noise with $\sigma = 10^{-2}$, (c) space variant η map in interval $[0(\text{blue}), 1(\text{red})]$.*

Example 2 In the second example, we investigate the influence of the smoothing regularization term in (1.1) with space adaptive η carried out by (1.2). To that aim, we take the *sphere* mesh with artificial scalar map painted on it, see Fig. 5.1 (a), and we corrupted it with white noise of standard deviation $\sigma = 10^{-2}$, Fig. 5.1 (b). We applied the proposed CNC model with $\lambda = 1000$ and $\eta = 0$, $\eta = 0.2$ and η adaptive with the map in interval $\eta \in [0, 1]$ illustrated in Fig. 5.1 (c) in false colors in the interval [blue,red]. The corresponding partition functions u^* are reported in the first row of Figure 5.2. We notice that for $\eta = 0$ the piecewise-constant reconstruction of u^*

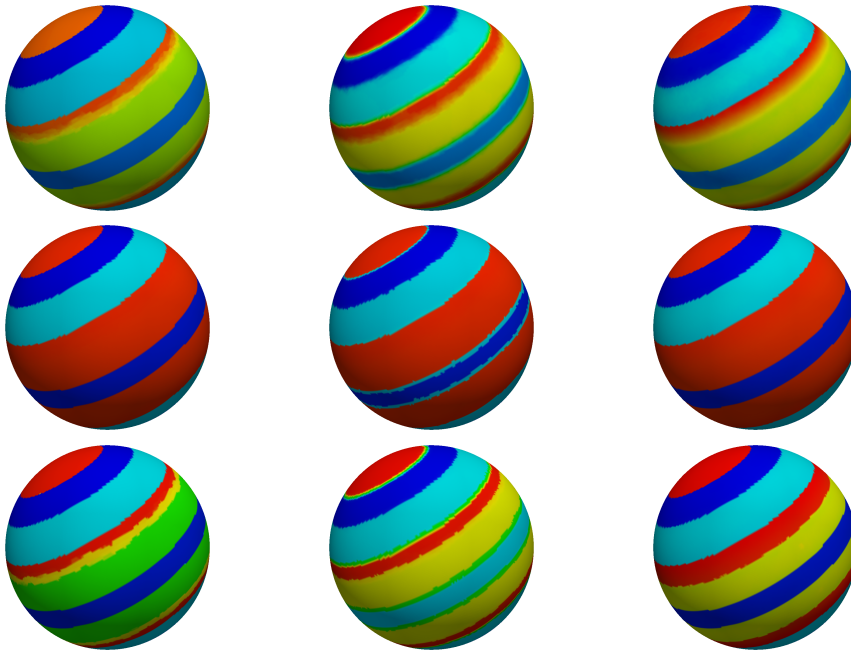


FIG. 5.2. *Example 2: Influence of the quadratic smoothing term and η space variant. First row from left to right: u^* obtained for $\eta = 0$, $\eta = 0.2$ and η space variant. Second and third row: associate clusterizations using three and five intensity levels.*



FIG. 5.3. *Example 3: noisy textured mesh, segmented parts in false colors with overimposed the boundary curves, detailed zoom.*

creates the typical staircase effect, while η constant everywhere causes smoothing over the sharp boundaries. The adaptive η well recovers the piece-wise smooth parts as well as the piecewise-constant parts. In the second and third row of Fig. 5.2 we plot the segmentation results obtained using three and five different levels of the intensity function. In the second row, the smoothed boundaries corrupt the segmentation result and we notice that in the third row left, the staircase effect creates additional unwanted parts for the piecewise-smooth area.

Example 3. Unlike image segmentation, the segmentation over surfaces demands for a post-processing boundary tracking, realized in the third phase of the segmentation algorithm, to nicely connect vertices identified as boundaries of the salient structures and then produce boundary curves. In Fig.5.3 we show the segmentation of a real-valued function on the mesh *horse*, $|V| = 129218$, $|T| = 258432$; which has been corrupted by an additive Gaussian noise with $\sigma = 5 \times 10^{-2}$. From left to right: the input noisy data, the boundary curves overimposed to the segmentation of u^* , and a zoomed detail. The boundaries between the segmented parts (black solid line) are detected and then smoothed by projecting the boundary vertices onto the cubic spline obtained by least-squares approximation (white solid line).

Example 4. In the last example we applied the proposed approach to segment a real-valued function over a 2-manifold into K disjoint regions that are “homogeneous” according to a cer-

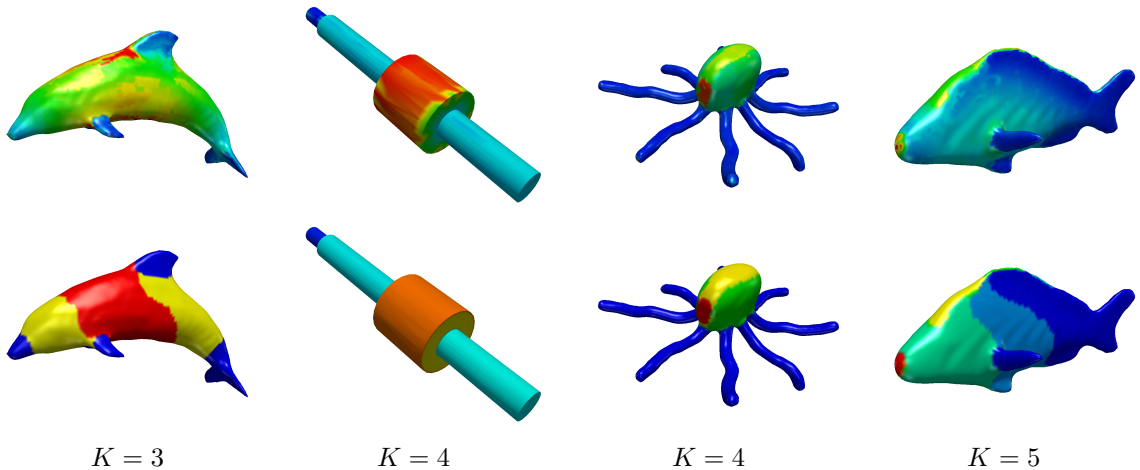


FIG. 5.4. Example 4: segmentation results (bottom row) obtained from the input SDF maps (top row). From left to right: *dolphin* $|V| = 7121$, $|T| = 14238$; *mech* $|V| = 1512$, $|T| = 3020$; *octopus* $|V| = 7251$, $|T| = 14498$; *fish* $|V| = 6656$, $|T| = 13308$.

tain feature. In this context the real-valued function represents the thickness map of an object, computed on the mesh by carrying out the so-called Shape Diameter Function (SDF) using the method proposed in [5]. The real-valued SDFs are shown in Fig. 5.4 top for several input meshes. The partition function is computed by our algorithm in the CNC regime, namely with $\tau_c = 0.99$, $\lambda = 1000$, η adaptive. Afterwards, the simple thresholding is used in order to clusterize the object into K homogeneously thick parts. The segmented parts, together with the number of partitions K below, are shown in Fig. 5.4 bottom, using false colors.

We conclude this section presenting an empirical investigation on the numerical convergence of the proposed ADMM-based minimization scheme in CNC regime. In Figure 5.5, we present the convergence plots for all the above-mentioned models considered. Even though we used the stopping criterion 10^{-4} , in Fig. 5.5 right we present the relative change plot (5.1) for the first 400 iterations together with the evaluation of the functional \mathcal{J} (left). The plots reported confirm that the numerical proposed CNC scheme smoothly converge. Moreover, the tests conducted so far suggest that the non-convex regime converges but it is much slower than the CNC regime.

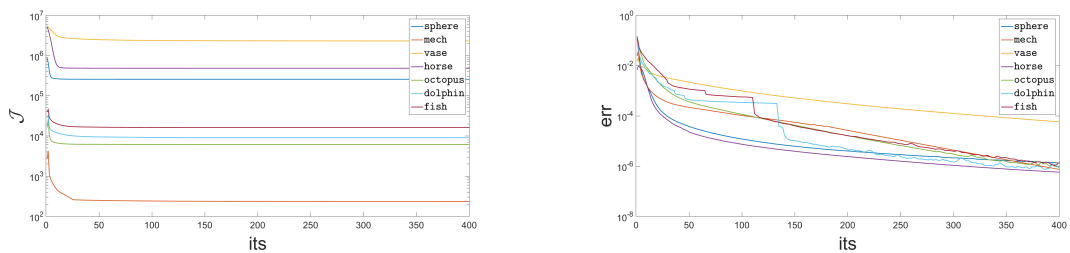


FIG. 5.5. Experimental convergence for the reported meshes. From left to right: \mathcal{J} and the relative change err.

6. Conclusions. We presented a variational CNC model for multiphase segmentation of real-valued functions on manifolds, which aims to represent the analogous of the Mumford–Shah functionals for images. A non-convex regularization term is a meaningful alternative to the ℓ_1 norm which allows for penalizing the non-smoothness of the inner segmented parts and preserves the length of the boundaries. The CNC strategy is generalized on 2-manifold represented by unstructured three dimensional meshes, and theoretical local conditions of convexity are derived for the proposed model where we explicitly used the manifold’s geometry. This result allows to benefit

from the advantages of using such non-convex regularizer while, at the same time, maintaining strict (or, better, strong) convexity of the optimization problem to be solved. An ad hoc iterative ADMM-based algorithm is introduced together with a specific multivariate proximal map to solve the minimization problem. The segmentation process is then completed by a thresholding step and a boundary tracking step to detect boundary curves of the segmented parts. Empirical convergence of the ADMM proposed iterative procedure is shown and future work will certainly investigate the theoretical aspects of the convergence.

REFERENCES

- [1] A. Blake, A. Zisserman, Visual Reconstruction, Cambridge, USA: MIT Press, 1987.
- [2] X. H. Cai, R. H. Chan, T. Y. Zeng, A two-stage image segmentation method using a convex variant of the Mumford–Shah model and thresholding, *SIAM Journal on Imaging Sciences*, 6(1), 368–390, 2013.
- [3] T. Chan, L. Vese, Active contours without edges. *IEEE Transactions on Image Processing*, 10, 266–277, 2001.
- [4] P.Y. Chen, I.W. Selesnick, Group-Sparse Signal Denoising: Non-Convex Regularization, *Convex Optimization*, *IEEE Trans. on Sign. Proc.*, 62, 464–3478, 2014.
- [5] M. Huska, S. Morigi, A meshless strategy for shape diameter analysis, *Visual Computer*, 33(3), 303–315, 2017.
- [6] M. Huska, A. Lanza, S. Morigi, F. Sgallari, Convex Non-Convex Segmentation over Surfaces, In: Lauze F., Dong Y., Dahl A. (eds) *Scale Space and Variational Methods in Computer Vision*. SSVM 2017. *Lecture Notes in Computer Science*, vol 10302. Springer, Cham, pp. 348 – 360, 2017.
- [7] A. Lanza, S. Morigi, F. Sgallari, Convex Image Denoising via Non-Convex Regularization. In: Aujol JF., Nikolova M., Papadakis N. (eds) *Scale Space and Variational Methods in Computer Vision*, SSVM 2015, *Lecture Notes in Computer Science*, vol 9087. Springer, Cham, 2015.
- [8] A. Lanza, S. Morigi, F. Sgallari, Convex Image Denoising via Non-convex Regularization with Parameter Selection, *Journal of Math. Imaging and Vision*, 56(2), 195–220, 2016.
- [9] A.Lanza, S.Morigi, I.Selesnick, F.Sgallari, Nonconvex Nonsmooth Optimization via Convex-Nonconvex Majorization-Minimization, *Numerische Math.*, DOI 10.1007/s00211-016-0842-x, 2016.
- [10] D.Mumford, J.Shah, Optimal approximations by piecewise smooth functions and associated variational problems, *Comm. pure and applied math.*,(42)5, 577–685, 1989.
- [11] M. Nikolova, Estimation of Binary Images by Minimizing Convex Criteria, in *Proc. IEEE Int. Conf. Image Process.*, 2, 108–112, 1998.
- [12] L.I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physics D*, 60(1-4), 259–268, 1992.
- [13] X. C. Tai, Y. Duan, Domain decomposition methods with graph cuts algorithms for total variation minimization, *Advances in Computational Mathematics*, 36(2), 175–199, 2012.
- [14] C. Wu, J. Deng, F. Chen, and X.-C. Tai, Scale-Space Analysis of Discrete Filtering over Arbitrary Triangulated Surfaces, *SIAM Journal on Imaging Sciences* 2009 2:2, 670–709, 2009.
- [15] C. Wu, J. Zhang, Y. Duan, X.-C. Tai, Augmented Lagrangian Method for Total Variation Based Image Restoration and Segmentation Over Triangulated Surfaces, *Journal of Scientific Computing*, 50(1), 145–166, 2012.
- [16] J. Zhang, J. Zheng, C. Wu, J. Cai, Variational Mesh Decomposition, *ACM Trans. Graph.*, 31(3), 21:1–21:14, 2012.

Appendix. Proof of Lemma 3.2.

Proof. Let $x := (x_0, x_1, x_2)^T \in \mathbb{R}^3$, be the vector of vertices associated with a generic triangle τ in T . We recall that the function $f_\tau(\cdot; \lambda, \eta, a)$ in (3.12) can be rewritten in more compact form as follows:

$$f_\tau(x; \lambda, \eta, a) = \frac{\lambda}{6} x^T x + \frac{\eta}{2} x^T Q_\tau x + (1 - \eta) \phi\left(\sqrt{x^T Q_\tau x}; a\right), \quad (6.1)$$

with the matrix Q in (3.6).

The matrix Q in (3.6) is by definition real, symmetric, and positive semi-definite with the following eigenvalue decomposition:

$$Q = V \Lambda V^T, \quad \Lambda = \text{diag}(0, \lambda_1, \lambda_2), \quad V V^T = V^T V = I, \quad (6.2)$$

where orthogonality of the modal matrix V in (6.2) follows from symmetry of the matrix Q and the nonzero eigenvalues satisfy $\lambda_1 \geq \lambda_2 > 0$. Then, we decompose the diagonal eigenvalues matrix Λ in (6.2) as follows:

$$\Lambda = Z\tilde{\Lambda}Z, \quad Z = \text{diag}(1, \sqrt{\lambda_1}, \sqrt{\lambda_2}), \quad \tilde{\Lambda} = \text{diag}(0, 1, 1). \quad (6.3)$$

Substituting (6.3) into (6.2), then (6.2) into (6.1), we obtain the following equivalent expression for the function f :

$$f_\tau(x; \lambda, \eta, a) = \frac{\lambda}{6} x^T x + \frac{\eta}{2} x^T V Z \tilde{\Lambda} Z V^T x + (1 - \eta) \phi\left(\sqrt{x^T V Z \tilde{\Lambda} Z V^T x}; a\right). \quad (6.4)$$

Recalling that the property of convexity for a function is invariant under non-singular linear application of its domain, we introduce the following one for the domain \mathbb{R}^3 of function f above:

$$x = Ty, \quad T := VZ^{-1} \in \mathbb{R}^{3 \times 3}, \quad (6.5)$$

which is non-singular due to V and Z being non singular matrices. By defining as $f_\tau^T := f_\tau \circ T$ the function f_τ in the T -transformed domain, we have:

$$f_\tau^T(y; \lambda, \eta, a) = \frac{\lambda}{6} y^T Z^{-2} y + \frac{\eta}{2} y^T \tilde{\Lambda} y + (1 - \eta) \phi\left(\sqrt{y^T \tilde{\Lambda} y}; a\right). \quad (6.6)$$

Recalling the definitions of Z and $\tilde{\Lambda}$ in (6.3), we can write (6.6) in explicit form:

$$\begin{aligned} f_\tau^T(y; \lambda, \eta, a) &= \frac{\lambda}{6} \left(y_0^2 + \frac{y_1^2}{\lambda_1} + \frac{y_2^2}{\lambda_2} \right) + \frac{\eta}{2} (y_1^2 + y_2^2) + (1 - \eta) \phi\left(\sqrt{(y_1^2 + y_2^2)}; a\right) \\ &= \frac{\lambda}{6} \left[y_0^2 + \left(\frac{1}{\lambda_2} - \frac{1}{\lambda_1} \right) y_2^2 \right] + \frac{\lambda}{6\lambda_1} (y_1^2 + y_2^2) + \frac{\eta}{2} (y_1^2 + y_2^2) + (1 - \eta) \phi\left(\sqrt{(y_1^2 + y_2^2)}; a\right) \\ &= \frac{\lambda}{6} \left[y_0^2 + \left(\frac{1}{\lambda_2} - \frac{1}{\lambda_1} \right) y_2^2 \right] + \frac{1}{2} \left(\eta + \frac{\lambda}{3\lambda_1} \right) (y_1^2 + y_2^2) + (1 - \eta) \phi\left(\sqrt{(y_1^2 + y_2^2)}; a\right) \\ &= \frac{\lambda}{6} \left[y_0^2 + \underbrace{\left(\frac{1}{\lambda_2} - \frac{1}{\lambda_1} \right)}_{>0} y_2^2 \right] + g_\tau(y_1, y_2; \lambda, \eta, a), \end{aligned} \quad (6.7)$$

where the function g_τ in (6.7) is defined in (3.14). Since the first term in (6.7) is (quadratic) convex, a sufficient condition for the function f_τ^T in (6.7) to be strictly convex is that the function g_τ in (3.14) is strictly convex. This concludes the proof after recalling that the function f_τ is strictly convex if and only if the function f_τ^T is strictly convex. \square

Proof of Proposition 4.1.

Proof. The demonstration of condition (4.16) for strict convexity of the function θ in (4.15) is straightforward. In fact, the function θ can be equivalently rewritten as

$$\theta(z) = \underbrace{\phi(\|z\|_2; a) + \frac{\bar{\eta} + \bar{\beta}}{2} \|z\|_2^2}_{\bar{\theta}(z)} + \mathcal{A}(z), \quad z \in \mathbb{R}^3, \quad (6.8)$$

with $\mathcal{A}(z)$ an affine function of z , so that a necessary and sufficient condition for θ to be strictly convex is that the function $\bar{\theta}$ in (6.8) is strictly convex. We then notice that $\bar{\theta}$ is almost identical to the function g_τ in (??), therefore, we can again use Proposition 2 in [4] leading to the following condition

$$\bar{\beta} > a - \bar{\eta}$$

and plugging in the definitions for $\bar{\beta}$ and $\bar{\eta}$ in (4.13), we obtain the convexity condition (4.16) for strict convexity of θ .

For the proof of statement (4.18), according to which the unique solution z^* of the strictly convex problem (4.17) is obtained by a shrinkage of vector r , we refer the reader to [8][Proposition 4.5].

We now prove statement (4.19). First, we notice that if $\|r\|_2 = 0$, i.e. r is the null vector, the minimization problem in (4.17) with the objective function $\theta(z)$ defined in (4.15) reduces to

$$\arg \min_{z \in \mathbb{R}^3} \left\{ \phi(\|z\|_2; a) + \frac{\bar{\eta} + \bar{\beta}}{2} \|z\|_2^2 \right\}. \quad (6.9)$$

Since both the terms of the cost function in (6.9) are monotonically increasing functions of $\|z\|_2$, the solution of (6.9) is clearly $z^* = 0$. Hence, the case $\|r\|_2 = 0$ can be easily dealt with by taking any value ξ^* in formula (4.18). We included the case $\|r\|_2 = 0$ in formula a) of (4.19).

In the following, we consider the case $\|r\|_2 > 0$. Based on the previously demonstrated statement (4.18), by setting $z = \xi r$, $\xi \geq 0$, we turn the original unconstrained 3-dimensional problem in (4.17) into the following equivalent constrained 1-dimensional problem:

$$\begin{aligned} \xi^* &= \arg \min_{0 \leq \xi < 1} \left\{ \phi(\|\xi r\|_2; a) + \frac{\bar{\eta}}{2} \|\xi r\|_2^2 + \frac{\bar{\beta}}{2} \|\xi r - r\|_2^2 \right\} \\ &= \arg \min_{0 \leq \xi < 1} \left\{ \phi(\|r\|_2 \xi; a) + \frac{\bar{\eta}}{2} \|r\|_2^2 \xi^2 + \frac{\bar{\beta}}{2} \|r\|_2^2 (\xi - 1)^2 \right\} \\ &= \arg \min_{0 \leq \xi < 1} \left\{ f(\xi) := \phi(\|r\|_2 \xi; a) + \|r\|_2^2 \left(\frac{\bar{\eta} + \bar{\beta}}{2} \xi^2 - \bar{\beta} \xi \right) \right\}, \end{aligned} \quad (6.10)$$

where in (6.10) we omitted the constant terms and introduced the cost function $f: \mathbb{R}_+ \rightarrow \mathbb{R}$ for future reference. Since the penalty function ϕ is twice continuously differentiable on \mathbb{R}_+ - see assumption A1) in Section 2 - the cost function f in (6.10) is also twice continuously differentiable on \mathbb{R}_+ . Moreover, f is strictly convex since it represents the restriction of the strictly convex function θ in (4.15) to the half-line ξr , $\xi \geq 0$. Hence, a necessary and sufficient condition for an inner point $0 < \xi < 1$ to be the global minimizer of f is as follows:

$$f'(\xi) = 0 \iff \|r\|_2 \left[\phi'(\|r\|_2 \xi; a) + \|r\|_2 ((\bar{\eta} + \bar{\beta}) \xi - \bar{\beta}) \right] = 0. \quad (6.11)$$

Since f is continuously differentiable and strictly convex on \mathbb{R}_+ , the first-order derivative $f'(\xi)$ is continuous and strictly increasing in the optimization domain $0 \leq \xi \leq 1$ and at the extremes we have:

$$f'(0^+) = \|r\|_2 [\phi'(0^+; a) - \bar{\beta} \|r\|_2], \quad f'(1) = \|r\|_2 [\phi'(\|r\|_2; a) + \bar{\eta} \|r\|_2]. \quad (6.12)$$

Since $\bar{\eta} \geq 0$, $\|r\|_2 > 0$ and $\phi'(t; a) > 0$ for any $t \geq 0$ by assumption A2) in Section 2, $f'(1)$ in (6.12) is positive. Hence, we have two cases. If $f'(0^+) \geq 0$, that is $\|r\|_2 \leq \phi'(0^+; a) / \bar{\beta} = 1 / \bar{\beta}$, $f'(t)$ is positive in $0 < t \leq 1$, hence the function f has its minimum at $\xi^* = 0$; if $f'(0^+) < 0$, that is $\|r\|_2 > 1 / \bar{\beta}$, then f has the minimum at its unique stationary point $0 < \xi^* < 1$, which can be obtained by solving the nonlinear equation in (6.11). The proof of statement (4.19) is thus completed. \square