

SPARSITY-INDUCING VARIATIONAL SHAPE PARTITIONING*

SERENA MORIGI[†] AND MARTIN HUSKA[‡]

Abstract. We propose a sparsity-inducing multi-channel multiple region model for the efficient partitioning of a mesh into salient parts. Our approach is based on rewriting the Mumford-Shah models in terms of piece-wise smooth/constant functionals that incorporate a non-convex regularizer for minimizing the boundary lengths. The solution of this optimization problem, obtained by an efficient proximal forward backward algorithm, is used by a simple thresholding/clusterization procedure to segment the shape into the required number of parts. Therefore, it is not necessary to further solve the optimization problem for a different number of partitioning regions. Experimental results show the effectiveness and efficiency of our proposals when applied to both single- and multi-channel (shape characterizing) functions.

Key words. mesh decomposition, variational segmentation, non-convex minimization, spectral clustering.

AMS subject classifications. 65M10, 78A48.

1. Introduction. The recent development of 3D scanning technology for reverse engineering and of sophisticated scan devices for medical imaging, have incredibly increased the availability of digital models of 3D physical objects simply represented by a set of 3D points on the surface of the object. These raw 3D data points, connected into spatial triangulations called 3D meshes, provide only local information of the structure of the surface. A high level insight of the raw 3D data is required to make the digital model useful for further processing, required in a variety of applications including computer graphics, CAD, CAM, and CAE. One of the fundamental processes which provide the necessary global insight on the model structure is the segmentation of a mesh, which represents the decomposition of the raw data into K -disjoint connected regions or parts that cover the entire object. Specific criteria dictate which elements belong to the same partition and these criteria are built upon the segmentation objective which in turn depends on the application. *Convexity/concavity* and *thickness* are popular shape criteria used in mesh decomposition. The convexity-driven segmentation of a shape finds a very intuitive match with the decomposition of an object made by the human vision system [12, 40]. This is due to the fact that an approximate convex decomposition can represent more accurately the important structural features of the model by ignoring insignificant features, such as wrinkles and other surface texture. Conversely, the thickness of parts of a shape is a less intuitive detection strategy for a human eye. Nevertheless, this geometry feature represents a strategic quantity in shape analysis, in the context of industrial design and production.

Much work has been done on approximate decomposition of a shape into convex components. Concavity-aware partitioning is proposed in [6] and [24]. In Asafi et al. [3], weakly convex components are obtained by a point-visibility test. The same idea is followed in [15] to approximate convex components of shape represented by possibly incomplete point clouds.

Segmentation methods based on spectral analysis mostly emphasize the concavity attribute, being able to partition even shallow concavities. The spectral analysis method uses the eigenvalues of properly defined matrices based on the connectivity of the graph in order to partition a mesh. Liu and Zhang [20] use spectral analysis on the dual graph of the mesh. They define an affinity matrix using both geodesic distances and angular distances, as proposed

*Received August 26, 2016. Accepted January 19, 2017. Published online on February 27, 2017. Recommended by L. Reichel.

[†]Department of Mathematics, University of Bologna, Bologna, Italy (serena.morigi@unibo.it).

[‡]Department of Mathematics, University of Padova, Padova, Italy (martin@math.unipd.it).

by the fuzzy clustering method in [16]. This type of matrix has been used successfully for clustering since it groups elements having high affinity; see for example [40].

Focusing on thickness as a segmentation property, the Shape Diameter Function (SDF), proposed in [32], is a measure of thickness that recovers volumetric information from the surface boundaries, thus providing a natural link between the object’s volume and its boundary. The SDF is a scalar function which maps, for every point on the surface, its distance to the opposite inner part of the object. As successfully proved in [32], this definition of the SDF is invariant to rigid body transformations of the whole object, and very robust to any deformation that does not alter the volumetric shape locally. In [14], the authors introduce an efficient dynamic approach to the computation of the SDF for a cloud of points.

In addition to the criteria that dictate the rules of the division into parts, the segmentation methods can be grouped into a few categories according to their computational methodology: (i) region growing, (ii) watershed-based, (iii) Reeb graphs, (iv) model-based, (v) skeleton-based, (vi) clustering, (vii) spectral analysis, (viii) explicit boundary extraction, (ix) critical points-based, (x) multiscale shape descriptors, (xi) Markov random fields, and (xii) variational segmentation. A detailed analysis of the aforementioned categories is given in [1] and exhaustive surveys are provided in [4, 31].

The concept of iteratively seeking a partition that minimizes a given error metric, named variational partitioning, has been introduced in [13] where the authors presented an optimization cost function based on clustering face normal of the mesh. Since then, several variational mesh partitioning have been proposed mostly for surface-based segmentation. In [39] a variational mesh segmentation framework based on fitting general quadrics (including planes as a special case) is proposed. Wu and Kobbelt [38] extend the results in [13] by introducing the sphere, the circular cylinder and the rolling ball patch as basic primitives. An important result on part-based segmentation has been presented in [40], where a convexified version of the variational Mumford-Shah model is presented and extended to 3D meshes. The cost function contains a data term measuring the variation within a segment and a regularization term based on the total variation of the gradient, measuring the length of the boundary between segments. This strategy produces piece-wise constant segmentation that may not be appropriate when applied to intensity inhomogeneous functions.

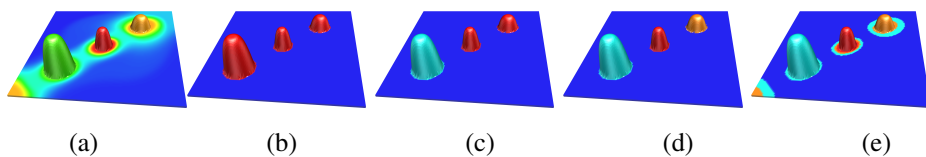


FIG. 1.1. From left to right, input function providing the cue for the segmentation of the shape: segmentation using the proposed SMCMMR approach into $K = 2$, $K = 3$, and $K = 4$ parts; segmentation obtained by applying the simulation of [40].

This paper focuses on a new strategy, named sparsity-inducing multi-channel multiple region (SMCMR), in the category of variational segmentations which share the common feature that they define the optimal segmentation as a the minimizer of an objective function, that generally depends on the given surface and on the scalar or vector functions used to identify the different salient regions. In particular we present a variational formulation based on a new variant of the Mumford-Shah models [26], where we adopt a sparsity inducing ℓ_p -norm approximation to the total length of the boundaries between parts, which promotes gradient-sparsifier solutions to our model. This newly introduced sparsity-inducing penalty

term better preserves the segmentation of small structured features in the shape, as it will be discussed in Section 3 and illustrated in Figure 3.1.

The proposed variational model will be named *single channel* when a scalar function is used to measure a given property of the surface, and *multi-channel* if a vector function is used in order to allow any logical combination of information in each channel to obtain the desired segmentation. In [40], the vector function is defined by the eigendecomposition strategy, while other examples of efficient objective functions that capture useful shape adjectives (compact, flat, narrow, perpendicular, etc.) are discussed in [33]. The proposed partitioning algorithm consists of two steps. The first computes the smooth/non-smooth minimizer and the second step automatically decomposes the object into a given number K of different regions, for example using a clustering K-means method. We can obtain any segmentation in K parts without recomputing the first minimization step, unlike the other proposed methods which require K to be fixed in advance. Such an independence on the number of partitions has already been considered in [5], where the number K of convex polyhedra is selected after the creation of a segmentation hierarchy of a tetrahedral mesh.

The simple example in Figure 1.1 illustrates two features of the proposed method: inhomogeneity and independence from the number of parts K . From the single channel input function shown in Figure 1.1(a), the SMCMR method decomposes the shape into K partitions, where $K = 2$ (Figure 1.1(b)), $K = 3$ (Figure 1.1(c)), and $K = 4$ (Figure 1.1(d)), which represents the most natural segmentation, i.e., three bumps and the base. In the presence of inhomogeneous functions, like the one in Figure 1.1(a), the segmentation can become particularly challenging. The sparsity in the magnitude of the solutions gradient allows for accurate segmentations. The variational segmentation method proposed in [40] produces, for $K = 4$, the result shown in Figure 1.1(e); this method fails in the decomposition of this bumpy shape, moreover it requires to recompute the minimization problem if the number of parts K changes.

Another important issue we address is how to improve the computational efficiency of the proposed variational segmentation models. The typical approach of gradient flow (i.e., marching the Euler-Lagrange PDE to steady state) usually presents very slow convergence. One standard way to overcome the computational issues is to treat the models as discrete optimization problems. Following this direction, we propose a proximal forward backward strategy, and an efficient split of the global formulation into simpler vertex-wise problems.

Summarizing, this work provides two main contributions:

- We define a multi-channel object partitioning framework, where the first step is based on a variational formulation with a sparsity-forcing penalty, to better fit the boundaries of the segmented regions, and a smooth regularizer, to deal with function inhomogeneity. This formulation makes the algorithm efficient since it does not depend on the number of partitions required.
- We derive a fast iterative algorithm to approximate faithfully the minimizer of the partitioning functional, which can be smooth in case of piece-wise constant segmentation, or nonsmooth for piece-wise smooth segmentation.

The outline of the paper is as follows. The affinity matrix and other basic notations are introduced in Section 2. In Section 3, we describe the proposed SMCMR segmentation model, and in Section 4 we present an efficient numerical solution to the derived optimization problem and an overview of our algorithm is given in Section 5. Numerical examples in Section 6 demonstrate the ability of the proposed segmentation methodology in partitioning meshes, considering both single and multi-channels functions. Conclusions and possible directions for future research are discussed in Section 7.

2. Construction of the affinity matrix. In this section, we describe the graph matrix which plays the role of the affinity matrix that will be used for human perception segmentation [21].

Let us consider a triangle mesh $\Omega := (V, T)$, which discretizes a manifold \mathcal{M} embedded in \mathbb{R}^3 , where $V = \{X_1, \dots, X_n\}$ is the set of n vertices, T is the connectivity graph, and we denote by $\mathcal{E} \subseteq V \times V$ the set of edges. Each vertex $X_i \in V$ has immediate neighbors $X_j, j \in N(X_i)$, to which it is connected by a single edge $e_{i,j}$. We denote by $N_\Delta(X_i)$ the set of triangles with vertex X_i , and by $|N_\Delta(X_i)| := \sum_{j \in N_\Delta(X_i)} A(\tau_j)$, where $A(\tau_j)$ is the area of the triangle τ_j .

The associated affinity matrix should encode mesh structural information which reflects how vertices are grouped in accordance with human perception.

Taking into account the curvature as shape information, we want to determine a perceptual partition of Ω such that the edges between different parts have very low weights (vertices in different clusters are dissimilar from each other), and the edges belonging to the same part have high weights (vertices within the same cluster are similar to each other). To this aim we define the affinity matrix $\mathcal{L} \in \mathbb{R}^{n \times n}$,

$$(2.1) \quad \mathcal{L}_{i,j} = \begin{cases} -w_{ij}, & i \neq j \text{ and } e_{ij} \in \mathcal{E}, \\ \sum_{j \in N(X_i)} w_{ij}, & i = j, \\ 0 & \text{otherwise,} \end{cases}$$

with the following similarity non-negative weights

$$(2.2) \quad w_{ij} := \frac{|N_\Delta(X_j)|}{\#N(X_i)} e^{-\|H(X_i) - H(X_j)\|_2^2 / (2\sigma^2)},$$

where the parameter $\sigma \in (0, 1]$ in (2.2) controls the width of the local neighborhoods. The mean curvature field H in (2.2) is obtained by exploiting the well-known relation

$$(2.3) \quad \Delta_{LB} X = -2HN,$$

between the vector field HN and the Laplace-Beltrami differential operator Δ_{LB} , applied to the coordinate functions X of a surface. According to [23, 29], the discretization of the Laplace-Beltrami operator (2.3) reads

$$L(X_i) = \frac{1}{2|N_\Delta(X_i)|} \sum_{j \in N(X_i)} \omega_{ij} (X_j - X_i),$$

$$\omega_{ij} = \frac{1}{2} (\cot \gamma_j + \cot \delta_j),$$

where γ_j, δ_j are the angles opposite to the edge $e_{i,j}$ in the triangles tuple connected by the edge.

The spectral decomposition of \mathcal{L} , defined in the following, provides a set of $(n - 1)$ non-trivial, smooth, shape intrinsic isometric-invariant maps. We refer the reader to [36] for details.

PROPOSITION 2.1. *The matrix $\mathcal{L} \in \mathbb{R}^{n \times n}$ defined in (2.1), associated to a connected mesh Ω of n vertices, satisfies the following properties:*

- 1) \mathcal{L} is symmetric and positive semidefinite;
- 2) $\mathcal{L} = U\Lambda U^T$, $\Lambda = \text{diag}(\lambda_i)$, $0 = \lambda_0 < \lambda_1 < \dots < \lambda_n$;
- 3) $\lambda_i, \forall i$ are real eigenvalues, $U^T U = I_n$ with I_n the identity matrix of order n , $U = \{v_0, v_1, \dots, v_n\}$ form an orthogonal basis of \mathbb{R}^n ;

4) If $f = \sum_{i=1}^n \langle f, v_i \rangle v_i$, the k -term approximation of f is given by

$$f_k = \sum_{i=1}^k \langle f, v_i \rangle v_i.$$

The first k eigenvectors associated to the smallest nonzero eigenvalues correspond to smooth and slowly varying functions, while the last one show more rapid oscillations. Property 4) defines the truncated spectral approximation of the \mathcal{L} matrix, that considers the contribution of the first k eigenpairs related to the smallest eigenvalues, which hold for identifying the main shape features at different scale forming a signature for shape characterization.

In case of eigendecomposition-based segmentation, a vector function f is simply the truncated spectral coordinates of a vertex X_i , denoted by

$$(2.4) \quad f(X_i) = (v_1(X_i), v_2(X_i), \dots, v_d(X_i)), \quad d \leq k,$$

where each v_j is normalized in the range $[-1, 1]$.

The number k , which represents the number of computed eigenpairs, is independent of the number K of partition required, and it should be chosen according to the shape resolution. In Figure 2.1 the first $k = 6$ eigenvectors of the affinity matrix (2.1) corresponding to the first

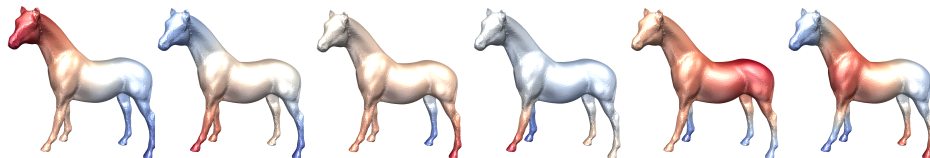


FIG. 2.1. The smallest $k = 6$ eigenfunctions of the horse mesh.

six nonzero eigenvalues are illustrated for the horse mesh, visualized in false colors in the range [blue, red].

The multi-channel function f for the proposed mesh segmentation algorithm can take the form (2.4), which is a vector function defined for each vertex X_i of the mesh. However, we are not limited to spectral information, and many other shape properties can be analogously exploited as multi-channel input function f .

Properties of the Laplacian spectrum have been widely investigated in shape analysis and employed in several applications in surface processing, such as shape segmentation, matching, and retrieval; see [19, 30]. The choice of the Laplacian matrix influences the spectral segmentation results as documented, for example, in [40], where instead of the more common cotangent based Laplacian proposed in [23, 29], the Laplacian matrix of the dual graph (triangle-based) is considered, weighted by the dihedral angles.

3. The sparsity-inducing multi-channel multiple region segmentation model. In this section we introduce the partitioning framework which exploits global or local shape information represented by a generic vector function $f : \Omega \rightarrow \mathbb{R}^d$, $d \geq 1$ at the points V , to infer a decomposition of the surface \mathcal{M} in salient parts. In Section 6, we present segmentation results for a well-known single-channel (scalar) function f , the shape diameter function, which measures the thickness property of an object, as well as results for a vector function defined in (2.4) (multi-channel) derived from spectral decomposition, which better reflects the human perception of shape decomposition. In the latter case, d is thus the number of considered eigenvectors of the affinity matrix \mathcal{L} in (2.1).

Our proposal is based on the well-known Mumford-Shah variational model introduced in [26] for image segmentation, briefly reported here for better understanding the main idea behind our model.

Let $\Omega \subset \mathbb{R}^3$ be a given bounded open set and $f : \Omega \rightarrow \mathbb{R}^d$ a measurable function on it. For gray-scale images, i.e., $d = 1$, the Mumford-Shah functional provides a partition $\Omega = \cup_{i=1}^K \Omega_i$, with respect to f , by combining a smoothing of homogeneous regions to the enhancement of boundaries among them, represented by the set of curves $\Gamma \subset \Omega$. The problem is formulated as the minimization of the following functional

$$(3.1) \quad \mathcal{J}_{MSs}(u, \Gamma) = \int_{\Omega} |f - u|^2 d\Omega + \alpha \text{Length}(\Gamma) + \beta \int_{\Omega \setminus \Gamma} |\nabla u|^2 d\Omega,$$

which is known as the *piece-wise smooth Mumford-Shah model*. This model approximates f by a piece-wise smooth function $u : \Omega \rightarrow \mathbb{R}$ which is differentiable everywhere in Ω except for a possible $(d - 1)$ -dimensional jump set Γ , at which u is discontinuous. The weight $\alpha > 0$ controls the length of the jump set Γ and $\beta > 0$ enforces the smoothness of u away from Γ . The second term in (3.1) imposes that the boundaries Γ be as short as possible. The restriction of (3.1) for the limiting case $\beta \rightarrow \infty$ imposes zero gradient outside Γ , that is, u is required to assume the constant value \bar{f}_i on each connected component Ω_i . The resulting minimization problem, known as the *piece-wise constant Mumford-Shah model*, often referred to as a special case of the Chan-Vese model [11], considers the following functional

$$(3.2) \quad \mathcal{J}_{MSc}(\Gamma) = \sum_{i=1}^K \int_{\Omega_i} |f - \bar{f}_i|^2 d\Omega + \sum_{i=1}^K \alpha \text{Length}(\Gamma_i),$$

where $\text{Length}(\Gamma_i) = |\partial\Omega_i|$ and $\bar{f}_i := \text{mean}_{\Omega_i} f$. Minimizing the models (3.1) or (3.2) represents a non-convex optimization problem, so the obtained solutions are in general local minimizers. Nevertheless, non-smooth, non-convex functionals have recently shown remarkable advantages over convex ones, for example in the image restoration context; theoretical explanation and numerical examples can be found in numerous papers [17, 18, 28].

The variational mesh decomposition introduced in [40] is based on a convex relaxation of (3.2) proposed by Nikolova et al. in [9]. However, the model (3.2) works well only if the intensity function f is homogeneous in each region. When this is not the case, that is, in the presence of inhomogeneities inside the regions to be segmented, the model (3.1) behaves better. For image segmentation, the authors introduced in [8] a convex relaxation where the boundary information is extracted from the total variation term.

Our goal is to develop an object partitioning framework that has the following properties:

- work on multi-channel (vector-valued) functions characterizing arbitrary object features;
- exploit an ad hoc sparsity-inducing regularizer for minimizing the total length of the boundaries while preserving their geometric features (corner, flat, etc.);
- make the procedure in the first step of the method independent of the number K of segments required, so there is no need to solve the whole problem again for different K values;
- work both for homogeneous and piece-wise smooth function f over each channel;
- detect portion of objects whose boundaries are characterized by significant changes both in f and in the local curvature.

We present a strategy for partitioning meshes, based on a new variant of the Mumford-Shah models (3.1) and (3.2), where we adopt an ℓ_p -norm approximation of the total length of the boundaries.

Let $f = (f_1, \dots, f_d)$ be a given vector-valued function with channels $f_i : \Omega \rightarrow \mathbb{R}$, $i = 1, \dots, d$, and let $u = (u_1, \dots, u_d)$ be a vector function on Ω , eventually nonsmooth, named the *partition function*. Unlike the color image segmentation process where all image channels participate jointly in driving the segmentation process [34], here we apply the variants of the Mumford-Shah models (3.1) and (3.2) to each channel u_i of u , for $i = 1, \dots, d$. In particular, in the first step, each channel u_i is separately computed by minimizing the *piece-wise smooth partitioning functional*

$$(3.3) \quad \min_{u_i} \mathcal{J}_s(u_i)$$

with

$$(3.4) \quad \mathcal{J}_s(u_i) := \frac{1}{2} \int_{\Omega} |f_i - u_i|^2 d\Omega + \frac{\alpha}{p} \int_{\Omega} \phi(\|\nabla u_i\|) d\Omega + \frac{\beta}{2} \int_{\Omega} |\nabla u_i|^2 d\Omega,$$

or the *piece-wise constant partitioning functional*:

$$(3.5) \quad \min_{u_i} \mathcal{J}_c(u_i)$$

with

$$(3.6) \quad \mathcal{J}_c(u_i) := \frac{1}{2} \int_{\Omega} |f_i - u_i|^2 d\Omega + \frac{\alpha}{p} \int_{\Omega} \phi(\|\nabla u_i\|) d\Omega,$$

where $\phi(t) := |t|^p$ is a penalty function with $p \in (0, 2]$, sparsity-inducing for $p < 1$, and $\beta := \beta(x)$, $\beta : \Omega \rightarrow [0, 1]$, is an adaptive function which approaches zero at the high curvature points of Ω .

In the second step, we apply a multi-channel clustering procedure to the vector function u to finalize the object partitioning. The number of parts K (phases) is only required in this second step, so users can choose or change it without the need of solving the previous stage again.

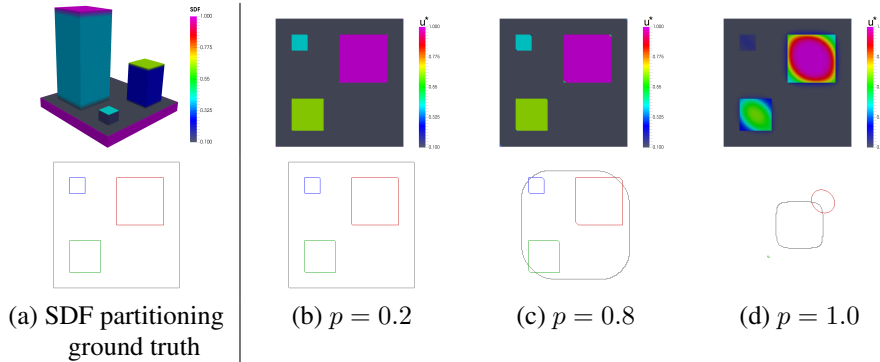


FIG. 3.1. Effect of the ℓ_p regularizer w.r. to the ℓ_1 regularizer for the SDF partitioning of the blocks mesh.

The ℓ_p penalty term is introduced in (3.4) and (3.6) to better control the length of the boundaries and substantially improves upon the ℓ_1 norm results. In particular, for $p = 1$ the penalty term in (3.4) and (3.6) corresponds to the total variation (TV) term which have been used in [40] to measure the length of the boundaries.

The benefit of using $p < 1$ is illustrated in Figure 3.1 for the segmentation of a mesh composed of variable sized boxes (Figure 3.1(a) top). Since the thickness property is used as criteria for partitioning, from the top view, the expected results are four boxes which are shown in Figure 3.1(a) bottom. The true thicknesses (heights) were used as thresholds. The top row of Figure 3.1(b), (c), and (d), shows the results obtained from the proposed variational model for different p values, which are used in step 2 to produce the simple partitions according to the given thresholds which represent the true heights. In the bottom row, we plot the partition boundaries, obtained as iso-contours of u^* in the top row, according to the thresholds. For the choice $p < 1$ in (3.6) our model preserves the sharp boundary shape, as illustrated in Figure 3.1(b) and (c), while for $p = 1$ the boundaries shrink and the small features disappear as illustrated in Figure 3.1(d). In particular for p approaching zero the boundary shape improves and the original intensities are preserved.

This behavior is justified from the fact that the well-known TV regularizer is defined as the continuous ℓ_1 norm, $p = 1$, which inevitably curtails originally salient boundaries to penalize their magnitudes. In particular, as discussed in [35], the TV of a feature is directly proportional to its boundary size, so that one way of minimizing the TV of that feature would be to reduce its boundary size, in particular by smoothing corners. Moreover, the change in intensity due to TV regularization is inversely proportional to the scale of the feature, so that very small-scaled features are removed.

In order to make the model independent of the scale of the feature to segment, we could use the ℓ_0 measure of the discrete gradient explicitly defined as $\|\nabla u\|_0 := \#\{x \mid \|\nabla u\|_2 \neq 0\}$, where $\#$ is the counting operator which measures how many times u changes its value. We propose to approximate the ℓ_0 measure of the gradient with the non-smooth non-convex and non-Lipschitz regularization term, ℓ_p quasi-norm, $\phi(t) = |t|^p$, with $0 < p < 1$, which has recently been proposed in image processing and compressed sensing since it promotes gradient-sparsier solutions or sparser solutions, substantially improving upon the ℓ_1 norm results [17].

This choice may lead to a challenging computation problem, since it requires non-convex (when $p < 1$), non-smooth minimization which, since it involves many minima, can get stuck in shallow local minima. However, in Section 4, we show how to solve efficiently these optimization problems.

4. Discretization of the SMCMR model. In the discrete setting, the 2-manifold \mathcal{M} embedded in \mathbb{R}^3 represents the boundary of the volumetric object to be partitioned, and it is discretized into a triangular mesh, denoted by Ω , which consists of the finite set V of n vertices, together with a subset $\mathcal{E} \subseteq V \times V$ of edges. We assume that functions on the manifold are sampled at the vertices V .

Approximate solutions to the shape partitioning problems (3.3) and (3.5) read respectively as the minimizations of the following functions

$$(4.1) \quad \mathcal{J}_s(u_i) := \frac{1}{2} \|u_i - f_i\|_2^2 + \frac{\alpha}{p} \sum_{j=1}^n \phi(\|(\nabla_w u_i)_j\|_2) + \frac{\beta}{2} \sum_{j=1}^n \|(\nabla_w u_i)_j\|_2^2,$$

$$(4.2) \quad \mathcal{J}_c(u_i) := \frac{1}{2} \|u_i - f_i\|_2^2 + \frac{\alpha}{p} \sum_{j=1}^n \phi(\|(\nabla_w u_i)_j\|_2),$$

where $f_i \in \mathbb{R}^n$ is a vector of values associated to the set of vertices V , and $u_i \in \mathbb{R}^n$ represents the discretization of the i th component of the partition function u to be estimated. The discrete operator $\nabla_w u(v)$ denotes the discretization of the weighted local variation of the function u at vertex v . Towards its computation we define the discrete analog of the *directional derivative*

on a 2-manifold \mathcal{M} as the edge derivative of u at a vertex $X_\ell \in V$ along an edge $e_{\ell,j} \in \mathcal{E}$ by the following difference operator $du(X_\ell, X_j)$

$$(4.3) \quad \frac{\partial u}{\partial e_{\ell,j}} \approx du(X_\ell, X_j) := \sqrt{w(X_\ell, X_j)}(u(X_j) - u(X_\ell)),$$

where $w : V \times V \rightarrow \mathbb{R}_+$ is a symmetric measure defined between the points X_ℓ and X_j , and $w(X_\ell, X_j) = 0$ if $(X_\ell, X_j) \notin \mathcal{E}$. Hence, the weighted gradient operator $\nabla_w u(X_\ell)$ of a function u at a vertex X_ℓ can be defined as the vector of all partial derivatives $du(X_\ell, X_j), \forall j \in N(X_\ell)$. Then its magnitude is given by

$$(4.4) \quad \|\nabla_w u(X_\ell)\|_2^2 = \sum_{j \in N(X_\ell)} w(X_\ell, X_j)(u(X_j) - u(X_\ell))^2.$$

The regularization terms in (4.1) and (4.2) encode a prior knowledge on the local variation of the partition function, expressed as (4.4).

In the following proposition we report the relation between the continuous p -Laplacian operator and its discretization which will be used in the sequel.

PROPOSITION 4.1. *Given a set of points $V = \{X_\ell\}_{\ell=1}^n$ on a 2-manifold \mathcal{M} , the nonlinear operator L_p^w of a twice differentiable function u defined as*

$$(4.5) \quad L_p^w u(X_\ell) = \frac{1}{2} \sum_{j \in N(X_\ell)} \gamma_p^w(X_\ell, X_j)(u(X_j) - u(X_\ell)),$$

with

$$(4.6) \quad \gamma_p^w(X_\ell, X_j) = w(X_\ell, X_j)(|\nabla u(X_j)|^{p-2} + |\nabla u(X_\ell)|^{p-2}),$$

represents the discrete approximation of the weighted p -Laplacian operator

$$(4.7) \quad \Delta_p^w u := \nabla_w \cdot (|\nabla_w u|^{p-2} \nabla_w u),$$

where ∇_w is the weighted gradient of u on \mathcal{M} .

Proof. Let $b(u) := |\nabla u|^{p-2}$ and b_ℓ be the evaluation of $b(u)$ at $X_\ell \in V$. By applying (4.3), the discretization of the weighted p -Laplacian operator (4.7) is given by

$$(4.8) \quad \begin{aligned} L_p^w u(X_\ell) &= \sum_j \sqrt{w_{\ell j}}(b_j du(X_j, X_\ell) - b_\ell du(X_\ell, X_j)) \\ &= \sum_j w_{\ell j}(b_j(u(X_\ell) - u(X_j)) - b_\ell(u(X_j) - u(X_\ell))) \\ &= \sum_j w_{\ell j}(b_\ell + b_j)(u(X_\ell) - u(X_j)). \end{aligned}$$

Replacing (4.6) in (4.8) we get (4.5). \square

The p -Laplacian is a nonlinear operator with the exception of the special case when $p = 2$, where it reduces to the regular Laplacian operator $\Delta_2 f = \operatorname{div}(\nabla f)$, while for $p = 1$ we get $\Delta_1 f = \nabla \cdot (\frac{\nabla f}{|\nabla f|})$, which is the mean curvature operator.

The classical gradient descent method for the numerical integration of the optimization problems (3.3) and (3.5) would involve the p -Laplacian flow, which is used, for example, in [25] for polygonal mesh simplification. However, while its numerical implementation could be straightforward, because of stability constraints the gradient descent has rather undesirable asymptotic convergence properties which can make it very inefficient.

In the rest of this section we propose a fast iterative method to approximate faithfully the minimizer of (4.1) and (4.2), which represent the discretized versions of (3.3) and (3.5) for $p \in (0, 2]$. The method presents a global minimum for $1 \leq p \leq 2$, while (4.1) and (4.2) are non-convex when $p < 1$ and a global optimal solution is not insured. The proposed iterative method has been implemented and evaluated as described in Section 5.

We focus on the minimization of \mathcal{J}_s in (4.1), since the functional \mathcal{J}_c in (4.2) can be seen as a special case of \mathcal{J}_s when $\beta = 0$. However, since \mathcal{J}_c is nonsmooth, in our unified treatment of the two optimal problems, we will adapt a proximal forward backward (PFB) strategy for nonsmooth optimization.

We first split the objective function into two terms, $h : \Omega \rightarrow \mathbb{R}$ and $g : \Omega \rightarrow \mathbb{R}$, where $h(u)$ is differentiable but $g(u)$ may not be differentiable (in case \mathcal{J}_c in (4.2) is applied). Then, (4.1) reads as

$$\mathcal{J}_s(u_i) = \underbrace{\frac{1}{2} \|u_i - f_i\|_2^2}_{h(u)} + \underbrace{\frac{\alpha}{p} \sum_{j=1}^n \phi(\|(\nabla_w u_i)_j\|_2) + \frac{\beta}{2} \sum_{j=1}^n \|(\nabla_w u_i)_j\|_2^2}_{g(u)}.$$

In the following for simplicity of notations we drop the subscripts i .

The optimization problem $\min(h(u) + g(u))$ is then solved by applying an iterative PFB-based scheme [7], where each iteration step is given by

$$\begin{aligned} (4.9) \quad v^{(k)} &:= u^{(k-1)} - \lambda_k \nabla h(u^{(k-1)}), \\ u^{(k)} &:= \arg \min_u \left\{ g(u) + \frac{1}{2\lambda_k} \|u - v^{(k)}\|_2^2 \right\} := (I + \lambda_k \partial [g](u^{(k)}))^{-1} v^{(k)} \\ (4.10) \quad &:= \text{prox}_{\lambda_k}^g(v^{(k)}), \end{aligned}$$

where $\partial_x[\phi](x^*)$ denotes the subdifferential with respect to x of the function ϕ calculated at x^* , and when ϕ is differentiable we have $\partial_x[\phi](x^*) = \{\nabla\phi(x^*)\}$ for all x^* . The explicit updating (4.9) represents the forward step, whereas the evaluation of the proximity operator (4.10) represents the implicit backward step, which leads to the following system of equations

$$(4.11) \quad (I + \lambda_k(\beta L_2^w + \alpha L_p^w))u^{(k)} = (1 - \lambda_k)u^{(k-1)} + \lambda_k f,$$

where I denotes the identity matrix of order n and L_p^w denotes the discretization of the weighted p -Laplacian operator given in (4.4). The presence in L_p^w of the (diffusivity) coefficient $\gamma_p^w(X_i, X_j)$ defined in (4.6) makes it highly nonlinear, and for arbitrary p even non-differentiable. A solver like Newton's method, which converges rapidly near a minimizer, provided the objective functional depends smoothly on the solution, does not work satisfactorily on it or eventually fails. Therefore, we introduce a gradient linearization technique for the nonlinear equations (4.11), resulting in the *lagged diffusivity fixed point algorithm* [37], based on the following idea.

In order to solve the equation $\nabla J(\hat{x}) = 0$, we write

$$\nabla J(x) = L(x)x - z,$$

where z is independent of x . Then, at each iteration k , one finds $x^{(k+1)}$ by solving the linear problem

$$(4.12) \quad L(x^{(k)})x^{(k+1)} = z.$$

A connection between the gradient linearization approach, the lagged diffusivity fixed point iterations, and the half-quadratic minimization has been investigated in [27], where it is shown that the methods construct exactly the same sequence of iterates $x^{(k+1)}$.

Setting $u^0 = f$ and following (4.12), the backward iteration (4.11) is then replaced by the linear system

$$(4.13) \quad (I + \lambda_k(\beta L_2^w + \alpha L_p^w(u^{(k-1)})))u^{(k)} = (1 - \lambda_k)u^{(k-1)} + \lambda_k f,$$

where the nonlinear (diffusion) operator L_p^w has been linearized by applying it to the function $u^{(k-1)}$.

The coefficient matrix of the linear system is symmetric positive definite and the linear system (4.13) is solvable. The unique solution is the approximate solution of (4.10). The linear convergence of the lagged diffusivity fixed point method for $p = 1$ is discussed in [10].

The backward iteration (4.13) can be further simplified using (4.5) thus obtaining n independent linear equations for each vertex $X_\ell \in V$:

$$(4.14) \quad u^{(k)}(X_\ell) = \frac{(1 - \lambda_k)u^{(k-1)} + \lambda_k(f + \sum_j(\alpha\gamma_{\ell j}^{(k-1)} + \beta w_{\ell j})u^{(k-1)}(X_j))}{1 + \lambda_k \sum_j(\alpha\gamma_{\ell j}^{(k-1)} + \beta w_{\ell j})},$$

where we omitted the γ dependence on w and p to improve readability. Since for each vertex X_ℓ , at each iteration k , the solution of the linear system (4.13) is reduced to the explicit solution of a linear diffusion equation, whose diffusivity depends on the previous iterate $u^{(k-1)}$, the overall computational cost for the solution of this problem is linear in the number of vertices.

For all $p \in [1, \infty)$, if the algorithm converges, then it converges to the solution of the minimized function u in (4.1). However, when $p < 1$ (non-convex case), if the algorithm converges to some function u the latter is not guaranteed to be the global minimum of the minimized function (4.1).

To finalize the partitioning algorithm, we need a suitable proposal for the weights in (4.4). To this aim, we remark that a genuine partitioning algorithm should make the boundaries correspond to strong affinity changes in the function values between adjacent regions. Therefore the weights are chosen to be boundary detecting functions

$$(4.15) \quad w(X_\ell, X_j) = e^{-\|f(X_\ell) - f(X_j)\|_2^2 / \sigma}.$$

The parameter $\sigma \in (0, 1]$ in (4.15) controls how much the similarities of two local neighbors are penalized. Smaller values of σ preserve smaller differences in the function f .

By using (4.15) we get a good measurement of similarity, which penalizes in (4.5)-(4.6) the spatial clusterization flow of the vertices with different features.

5. The Algorithm SMCMR. To summarize the previous results, we report in Algorithm 1 the main steps of the proposed Algorithm SMCMR for mesh decomposition, based on the variational formulations (4.1) and (4.2).

The partitioning algorithm consists of two steps; the first one computes the minimizer u_i for the i th channel, $i = 1, \dots, d$, by the PFB-based iterative scheme described in Section 4. In particular, the partition function u_i is obtained by iterating (4.14) with the weights given in (4.15) for each vertex $X_j \in V$, until the relative change of u_i is below a fixed small tolerance ϵ . The step sizes λ_k can be found by a line search, that is, their values are chosen at each iteration. However, we followed the strategy to set $\lambda_0 = 10$ at the beginning, and update λ_k at each iteration by a factor 0.9.

ALGORITHM 1: SMCMR segmentation.

inputs: mesh data Ω , $f \in \mathbb{R}^d$, number of parts K
output: classification vector $\text{Label} \in \mathbb{R}^n$
parameters:

- penalty p , tolerance ϵ
- length regularizer $\alpha > 0$
- smooth regularizer $\beta > 0$
- similarity coefficient $\sigma > 0$ for w in (4.15)

STEP 1: PFB Initialization: $u^{(0)} = f$,
for $i = 1, \dots, d$ **do:**

- $k := 1$, $\lambda_0 := 10$,
- repeat**
- FS: $v_i^{(k)} := (1 - \lambda_{k-1})u_i^{(k-1)} + \lambda_{k-1}f_i$
- BS: compute $u_i^{(k)}$ by (4.14)
- Update: $\lambda_k := 0.9\lambda_{k-1}$, $k = k + 1$
- until** $\|u_i^{(k)} - u_i^{(k-1)}\|_2 < \epsilon$
- end for**

 $u^* = u^{(k)}$

STEP 2: Segmentation of the mesh into K parts, using u^* .
 $\text{Label}(X_i) = J$, $J \in \{1, \dots, K\}$, $\forall X_i \in V$ by (5.1).

STEP 2 is an automatic thresholding/clusterization procedure, and we could follow the classical K-means algorithm, with the K-means++ algorithm for cluster center initialization [22, 2]. However, the K-means method is strongly sensitive to the initialization of cluster centers due to its non-convexity. In particular, it favors the centroids as far away as possible from each other, while for the proposed segmentation model, two salient parts of the object can have centroids not too far away. Therefore, instead of using K-means++ initialization, we can set the cluster centroids c_i , $i = 1, \dots, K$, by simply assigning to each of them the value of u^* at a point in each salient part. Then the clusterization is achieved in only one iteration by labeling each vertex as

$$(5.1) \quad \text{Label}(X_i) = \arg \min_{j=1, \dots, K} \|u^*(X_i) - c_j\|_2,$$

without updating the cluster centroids, as it is required instead in the K-means algorithm.

The simple procedure mentioned above, in case of a single-channel function, coincides with thresholding.

6. Experimental results. In this section we describe the experimental results which demonstrate the performance of our segmentation approach, both in case of a single channel input function, $f \in \mathbb{R}$, in regime of piece-wise constant segmentation (see Section 6.1), and in case of multi-channel piece-wise smooth segmentation; see Section 6.2.

Experimental tests were performed on an Intel® Core™i7-4720HQ Quad-Core 2.6 GHz machine, with 12 GB/RAM and a Nvidia GeForce GTX 860M graphics card, running a Windows OS. The code is written in C++ using the EIGEN mathematical library, and it was executed without any additional hardware support, e.g., parallelization, GPU support, register usage. To compute the solutions of the large sparse eigenproblems required for

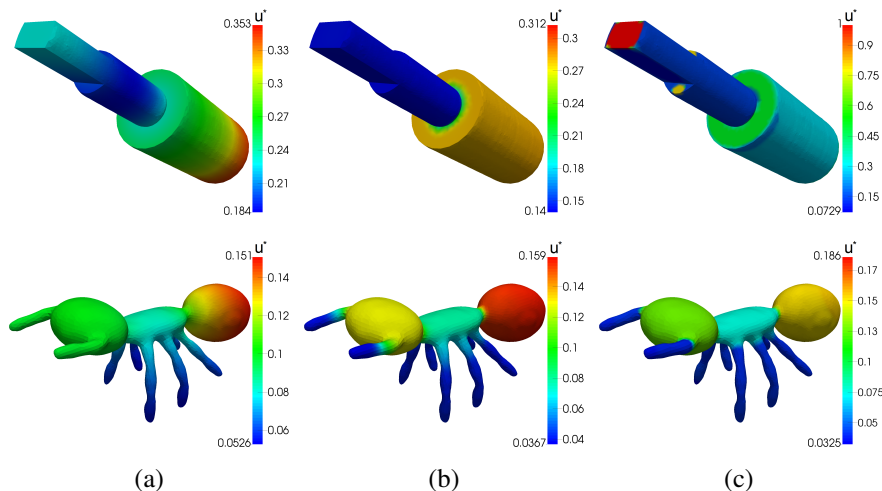


FIG. 6.1. Effect of the parameter p on STEP 1 of Algorithm SMC MR: (a) $p = 2$, (b) $p = 1$, (c) $p = 0.8$.

multi-channel segmentation, we used the wrapper EIGEN/Spectra (<http://yixuan.cos.name/spectra/>) which provides an efficient implementation of the Arnoldi method.

We tested the proposed algorithms on a set of meshes downloaded from the data repository website <http://segeval.cs.princeton.edu>, [12]. The dataset represents geometric models with different characteristics in terms of details, “sharpness”, and level of refinement, and present a medium dense vertex distribution.

The figures reported in this section were visualized by the software ParaView, and its VTK reader. In the examples illustrated in this section, we did not apply any post-process (smoothing, etc.) to the boundaries between the segmented parts in order to not alter the results computed by the application of the variational partitioning model.

6.1. Single-channel partitioning based on SDF. In this example we aim to decompose the surface boundary of an object into meaningful parts using the shape diameter values as a shape attribute to distinguish the salient parts. Therefore we expect the solution to be composed of homogeneous regions surrounded by closed contours which separates parts with significantly different thicknesses. We applied Algorithm SMC MR, with $\beta = 0$ and the input data $f \in \mathbb{R}$, which was the SDF map computed by the dynamic algorithm proposed in [14]. The result is a piece-wise constant approximation of the given SDF initial data enforcing sparsity in the gradient magnitude of the solution. The model data of the mesh samples from the data repository reported for this single-channel partitioning example are illustrated in Table 6.1.

The decomposition results strongly depend on the parameter p which forces the sparsity in the gradient of the solution u^* . The effects of the parameter p can be observed, for the `ant` and `mech_1` data sets, in Figure 6.1 where the colors (from red (large) to blue (small)) indicate the value of the solution u^* from STEP 1 of Algorithm SMC MR. In this experiment, we fixed $\alpha = 1$ to highlight the effect of parameter p , however, similar results can be obtained for different α values. When $p > 1$, the solution of the optimization process behaves like a smoothing flow, as illustrated in Figure 6.1(a), thus destroying the boundaries between parts. This effect is easily justified in terms of the p -Laplacian operator which, for $p = 2$, turns into the classical Laplace-Beltrami operator Δ_2 . For the choice of $p > 1$, the tuning of the parameter α does not help to improve the result. When $p \leq 1$, as shown in Figure 6.1(b)

TABLE 6.1

Timing results in seconds for single-channel (SDF) partitioning: computing time of one iteration (Iter), and total computing time using the tolerances $\epsilon = 10^{-2}$ and $\epsilon = 10^{-4}$.

Data set	Size($ V $)	Iter	$\epsilon = 10^{-2}$	$\epsilon = 10^{-4}$
ant	7038	0.008	0.025	1.832
armadillo	25193	0.048	0.209	12.718
blocks	6146	0.008	0.015	0.328
bird	8946	0.014	0.090	4.258
camel	9757	0.013	0.049	2.793
dolphin	7573	0.011	0.040	4.174
mech_1	10400	0.012	0.022	0.389
mech_2	1512	0.002	0.004	0.086
octopus_1	7251	0.009	0.029	1.740
octopus_2	243	0.001	0.002	0.029
pliers_1	3906	0.004	0.014	1.025
wolf	4712	0.006	0.017	1.645

and (c), the regularization term induces the sparsity of the u^* function leading to cleaner, straightforward partitioning clues for the underlying object.

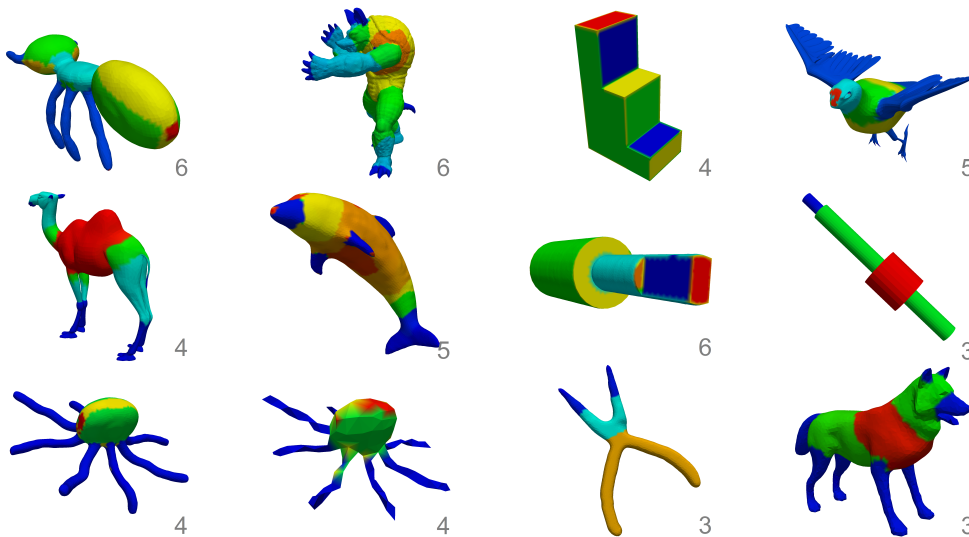


FIG. 6.2. Examples of single-channel partitioning based on SDF into patches with similar thickness.

In Figure 6.2 a sample set of objects partitioned into patches of different thickness is shown. The results were obtained by applying Algorithm SMCMM with $d = 1$, $p = 0.8$, tolerance $\epsilon = 10^{-4}$ and $\alpha = 1$. At the bottom right corner of each object we report the value of K , which in this type of partitioning is associated to the number of clusters having similar thickness. The corresponding computing times are reported in Table 6.1: for one iteration k (third column), the total time required by STEP 1 with $\epsilon = 10^{-4}$ and $\epsilon = 10^{-2}$ are reported in the fourth and fifth columns, respectively. Although we used the more stringent ϵ tolerance in the examples shown in Figure 6.2, we noticed that for many input shapes the results for $\epsilon \leq 10^{-2}$ are very favorable too. It is also worth noting that it is not necessary to require large

scale models to generate good results. In fact our algorithm generates acceptable segmentation results independently of the resolution of the meshes, as illustrated for the two octopus meshes in Figure 6.2 (last row – left) which present different resolutions.

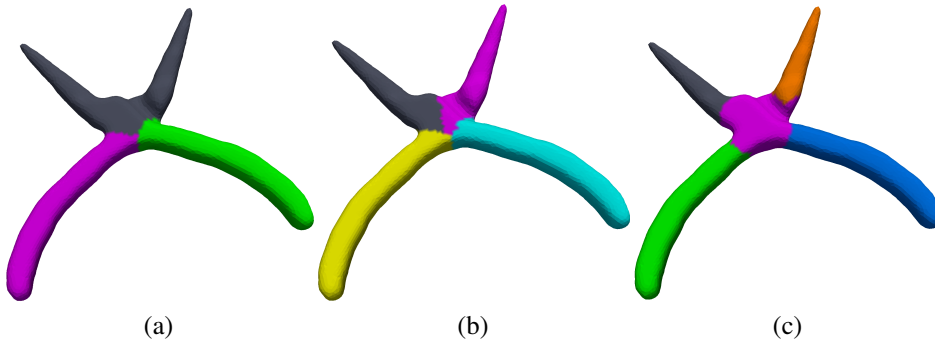


FIG. 6.3. Multi-channel partitioning of the *pliers* mesh into K parts: (a) $K = 3$, (b) $K = 4$, (c) $K = 5$.

6.2. Multi-channel partitioning based on spectral analysis. For the spectral partitioning, which aims to simulate the decomposition performed by a human being, the eigendecomposition of the affinity matrix described in Section 2 is preliminarily applied, obtaining 15 non-constant eigenvectors for each object in the repository data set. The affinity matrix weights (2.2) were computed using $\sigma = 0.5$ for every object. The dimension d of the multi-channel function f used as input of Algorithm SMC MR can be smaller than or equal to the number of eigenvectors, that is, $d \leq 15$. The usual choice is to take the first d significant (well shape-describing) ones. Our choice is shown in the third column of Table 6.2. Finally, our algorithm allows us to consider a number of partitions K independents of d . An example of this benefit is illustrated in Figure 6.3. First, STEP 1 of Algorithm SMC MR is applied using $d = 3$ channels, by considering the first three eigenfunctions among the 15 computed ones. Then STEP 2 is recomputed for $K = 3$ (Figure 6.3(a)), $K = 4$ (Figure 6.3(b)), and $K = 5$ (Figure 6.3(c)).

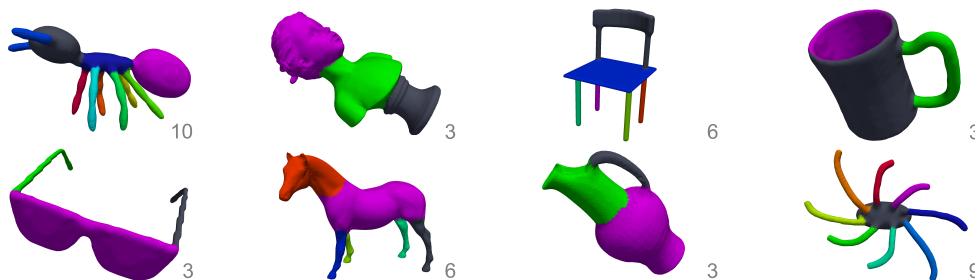


FIG. 6.4. Examples of multi-channel partitioning into patches simulating human-based segmentation.

The effectiveness of the proposed Algorithm SMC MR in partitioning the surface patches is shown in Figure 6.4 for a selected set of objects from the repository. At the bottom right corner of each object, we report the K value, that is, the number of partitions produced. Details on the model sizes ($\text{Size}(|V|)$), the number of channels considered (d), and the computing times for these objects are reported in Table 6.2. In particular, we denote as Spectra the timing

TABLE 6.2

Timing results in seconds for multi-channel partitioning: computing time for the eigendecomposition (Spectra) and overall computing time (Time) for the (d)-channel Algorithm SMCMR.

Data set	Size(V)	d	Spectra (s)	Time (s)
ant	7038	8	0.406	5.015
bust	25467	2	3.639	5.156
chair	14372	5	1.247	6.787
cup	15127	2	9.999	2.256
glasses	7407	2	0.455	0.593
horse	8078	6	0.451	5.084
octopus_3	5944	8	0.310	4.856
pliers_2	5110	3	0.293	1.654
vase	10637	3	0.591	3.235

for the eigendecomposition to compute the first 15 non-constant eigenvectors, while we report as Time the overall computing time in seconds for running the d -channel Algorithm SMCMR.

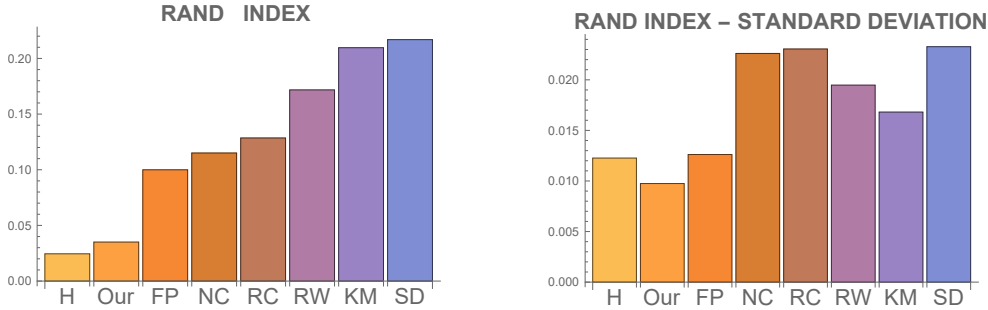


FIG. 6.5. Averaged dissimilarity ($1 - RI$) for the comparison of Algorithm SMCMR (Our) with other methods. Left: comparison w.r.to the human-generated segmentation. Right: standard deviation from the average. The lower the better.

We compared the results of Algorithm SMCMR with other popular segmentation methods and with human-generated segmentation, both provided by the benchmark in [12]. Namely, the methods considered are: fitting primitives (FP), normalized cuts (NC), randomized cuts (RC), random walks (RW), K-means (KM), shape diameter function (SD).

For the choice of a unifying comparison measure, we considered the *Rand Index* metric, denoted by RI , which measures the likelihood that a pair of faces are either in the same segment in two segmentations, or in different segments in both segmentations. If we denote S_1 and S_2 as two segmentations, s_i^1 and s_i^2 as the segment IDs of face i in S_1 and S_2 , and M as the number of faces in the polygonal mesh, $C_{ij} = 1$ if and only if $s_i^1 = s_j^1$ and $P_{ij} = 1$ if and only if $s_i^2 = s_j^2$, then we can define Rand Index as:

$$RI(S_1, S_2) = \binom{M}{2}^{-1} \sum_{i,j,i < j} [C_{ij}P_{ij} + (1 - C_{ij})(1 - P_{ij})].$$

This measure reflects the similarity between two segmentations, i.e., $C_{ij}P_{ij} = 1$ indicates that faces i and j have the same ID in both segmentations, and $(1 - C_{ij})(1 - P_{ij})$ indicates that faces i and j have different IDs in the segmentations being compared.

As well as in [12], we report in Figure 6.5 the estimates $1 - RI$ to show dissimilarities from the human-based segmentation averaging the results for each object in the repository data set. Therefore, the lower bars represent better results. The chart presented was computed as the average over RI for each segmentation method mentioned w.r.to human-based segmentation. In Figure 6.5, the bar labeled as “H” represents the average of the human-generated segmentations, in order to track the dissimilarities over human-produced results. We also report in Figure 6.5 (right) the standard deviation from the average. We can conclude that Algorithm SMCMR is quite consistent, compared to the other methods and the human-generated segmentations.

7. Conclusions and future work. In this paper we presented our proposal for the partitioning of an object, represented by a triangular mesh, into K separated regions. Our work is based on recent advances in sparsity-inducing penalties that have been successfully applied in image processing [17]. The multi-channel object partitioning framework is based on a variational formulation, where we introduced a novel shape metric, allowing the capture of more subtle details of the segmented boundaries than the traditional ℓ_1 metric. The sparsity imposed in the variational formulation represents the key aspect for a successful shape partitioning. In order to deal with function inhomogeneity, the functional has been enriched with a smooth regularizer. Thus, the resulting variational models hold the potential both for piece-wise constant and piece-wise smooth segmentations. We propose a fast iterative algorithm to accurately approximate the minimizer of the partitioning functional. From the computational point of view, the efficiency can be further improved by CPU/GPU parallelization of the vertex-wise computation in the backward step of the SMCMR Algorithm. This aspect deserves a deeper investigation.

The mathematical framework is robust and efficient; however the ℓ_p seminorm term in the functional leads to a non-convex optimization problem, whose solution can be stalled at local minima. A future improvement that we would like to explore is to replace the ℓ_p penalty term with a another sparsity inducing, non-convex, but parametric penalty function, in order to control the convexity of the overall functional and benefit of the convex optimization tools.

An advantage of this proposal with respect to other methods is that the solution is independent of the number of partitions K required, which is only exploited in the post-processing second step. However, the decomposition step is currently implemented as a naive K -means-alike clusterization algorithm which coincides with thresholding in the case of single channel variant of Algorithm SMCMR; therefore we also plan to replace it with a more convenient, convex, thresholding formulation.

The ability of the proposed algorithm to object partition has been illustrated for the shape diameter attribute and for spectral decomposition. However, our formulation is quite general, and other surface attributes can be used instead by suitably initializing the variational problem, thus obtaining a generalized partitioning framework.

Acknowledgments. We would like to thank the referees for comments that lead to improvements of the presentation. This work was partially supported by GNCS-INDAM, Italy.

REFERENCES

- [1] A. AGATHOS, I. PRATIKAKIS, S. PERANTONIS, N. SAPIDIS, AND P. AZARIADIS, *3D mesh segmentation methodologies for CAD applications*, Comput.-Aided Des. Appl., 4 (2007), pp. 827–841.
- [2] D. ARTHUR AND S. VASSILVITSKII, *K-means++: the advantages of careful seeding*, in Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, 2007, pp. 1027–1035.
- [3] S. ASAFI, A. GOREN, AND D. COHEN-OR, *Weak convex decomposition by lines-of-sight*, Comput. Graphics Forum, 32 (2013), pp. 23–31.

- [4] M. ATTENE, S. KATZ, M. MORTARA, G. PATANE, M. SPAGNUOLO, AND A. TAL, *Mesh segmentation—a comparative study*, in Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006, IEEE Conference Proceedings, Los Alamitos, 2006, pp. 7–20.
- [5] M. ATTENE, M. MORTARA, M. SPAGNUOLO, AND B. FALCIDIENO, *Hierarchical convex approximation of 3D shapes for fast region selection*, Comput. Graph. Forum, 27 (2008), pp. 1323–1332.
- [6] O. K.-C. AU, Y. ZHENG, M. CHEN, P. XU, AND C.-L. TAI, *Mesh segmentation with concavity-aware fields*, in IEEE Trans. Vis. Comp. Graphics, 18 (2012), pp. 1125–1134.
- [7] K. BREDIES, *A forward-backward splitting algorithm for the minimization of non-smooth convex functionals in Banach space*, Inverse Problems, 25 (2009), 015005 (20 pages).
- [8] X. CAI, R. CHAN, AND T. ZENG, *A two-stage image segmentation method using a convex variant of the Mumford-Shah model and thresholding*, SIAM J. Imaging Sci., 6 (2013), pp. 368–390.
- [9] T. F. CHAN, S. ESEDOGLU, AND M. NIKOLOVA, *Algorithms for finding global minimizers of image segmentation and denoising models*, SIAM J. Appl. Math., 66 (2006), pp. 1632–1648.
- [10] T. F. CHAN AND P. MULET, *On the convergence of the lagged diffusivity fixed point method in total variation image restoration*, SIAM J. Numer. Anal., 36 (1999), pp. 354–367.
- [11] T. F. CHAN AND L. A. VESE, *Active contours without edges*, IEEE Trans. Image Process., 10 (2001), pp. 266–277.
- [12] X. CHEN, A. GOLOVINSKIY, AND T. FUNKHOUSER, *A benchmark for 3D mesh segmentation*, ACM Trans. Graph., 28 (2009), Art. 73 (12 pages).
- [13] D. COHEN-STEINER, P. ALLIEZ, AND M. DESBRUN, *Variational shape approximation*, ACM Trans. Graph., 23 (2004), pp. 905–914.
- [14] M. HUSKA AND S. MORIGI, *A meshless strategy for shape diameter analysis*, The Visual Computer, 33 (2017), pp. 303–315.
- [15] O. V. KAICK, N. FISH, Y. KLEIMAN, S. ASAFI, AND D. COHEN-OR, *Shape segmentation by approximate convexity analysis*, ACM Trans. Graph., 34 (2014), Art. 4 (11 pages).
- [16] S. KATZ AND A. TAL, *Hierarchical mesh decomposition using fuzzy clustering and cuts*, in ACM SIGGRAPH 2003 Papers, SIGGRAPH '03, ACM, New York, 2003, pp. 954–961.
- [17] A. LANZA, S. MORIGI, AND F. SGALLARI, *Constrained TV_p - ℓ_2 model for image restoration*, J. Sci. Comput., 68 (2016), pp. 64–91.
- [18] ———, *Convex image denoising via non-convex regularization with parameter selection*, J. Math. Imaging Vision, 56 (2016), pp. 195–220.
- [19] B. LÉVY AND H. R. ZHANG, *Spectral mesh processing*, in ACM SIGGRAPH 2010 Courses, SIGGRAPH '10, ACM, New York, 2010, Art. 8 (312 pages).
- [20] R. LIU AND H. ZHANG, *Segmentation of 3D meshes through spectral clustering*, in Proceedings of the 12th Pacific Conference on Computer Graphics and Applications, IEEE Computer Society, Washington, 2004, pp. 298–305.
- [21] ———, *Mesh segmentation via spectral embedding and contour analysis*, Comput. Graph. Forum, 26 (2007), pp. 385–394.
- [22] D. J. C. MACKAY, *Information Theory, Inference and Learning Algorithms*, Cambridge University Press, New York, 2003.
- [23] M. MEYER, M. DESBRUN, P. SCHRÖDER, AND A. BARR, *Discrete differential-geometry operators for triangulated 2-manifolds*, in Visualization and Mathematics III, H.-C. Hege and K. Polthier, eds., Mathematics and Visualization, Springer, Berlin, 2003, pp. 35–57.
- [24] J.-M. LIEN AND N. M. AMATO, *Approximate convex decomposition of polyhedra*, in Proceedings of the ACM Symposium on Solid and Physical Modeling 2007, ACM, New York, 2007, pp. 121–131.
- [25] S. MORIGI AND M. RUCCI, *Multilevel mesh simplification*, The Visual Computer, 30 (2014), pp. 479–492.
- [26] D. MUMFORD AND J. SHAH, *Optimal approximations by piecewise smooth functions and associated variational problems*, Comm. Pure Appl. Math., 42 (1989), pp. 577–685.
- [27] M. NIKOLOVA AND R. CHAN, *The equivalence of half-quadratic minimization and the gradient linearization iteration*, IEEE Trans Image Process., 16 (2007), pp. 1623–1627.
- [28] M. NIKOLOVA, M. K. NG, AND C.-P. TAM, *Fast nonconvex nonsmooth minimization methods for image restoration and reconstruction*, IEEE Trans. Image Process., 19 (2010), pp. 3073–3088.
- [29] U. PINKALL AND K. POLTHIER, *Computing discrete minimal surfaces and their conjugates*, Experiment. Math., 2 (1993), pp. 15–36.
- [30] M. REUTER, S. BIASOTTI, D. GIORGI, G. PATANE, AND M. SPAGNUOLO, *Discrete Laplace-Beltrami operators for shape analysis and segmentation*, Comput. Graph., 33 (2009), pp. 381–390.
- [31] A. SHAMIR, *A survey on mesh segmentation techniques*, Comput. Graph. Forum, 27 (2008), pp. 1539–1556.
- [32] L. SHAPIRA, A. SHAMIR, AND D. COHEN-OR, *Consistent mesh partitioning and skeletonisation using the shape diameter function*, The Visual Computer, 24 (2008), pp. 249–259.
- [33] P. SIMARI, D. NOWROUZSAHRAI, E. KALOGERAKIS, AND K. SINGH, *Multi-objective shape segmentation and labeling*, Comput. Graph. Forum, 28 (2009), pp. 1415–1425.
- [34] E. STREKALOVSKIY, A. CHAMBOLLE, AND D. CREMERS, *A convex representation for the vectorial Mumford-*

- Shah functional*, in 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE Conference Proceedings, Los Alamitos, 2012, pp. 1712–1719.
- [35] D. M. STRONG AND T. F. CHAN, *Edge-preserving and scale-dependent properties of total variation regularization*, Inverse Problems, 19 (2000), pp. S165–S187.
 - [36] B. VALLET AND B. LEVY, *Spectral geometry processing with manifold harmonics*, Comput. Graph. Forum, 27 (2008), pp. 251–260.
 - [37] C. R. VOGEL AND M. E. OMAN, *Iterative methods for total variation denoising*, SIAM J. Sci. Comput., 17 (1996), pp. 227–238.
 - [38] J. WU AND L. KOBBELT, *Structure recovery via hybrid variational surface approximation*, Comput. Graph. Forum, 24 (2005), pp. 277–284.
 - [39] D.-M. YAN, W. WANG, Y. LIU, AND Z. YANG, *Variational mesh segmentation via quadric surface fitting*, Comput.-Aided Des., 44 (2012), pp. 1072–1082.
 - [40] J. ZHANG, J. ZHENG, C. WU, AND J. CAI, *Variational mesh decomposition*, ACM Trans. Graph., 31 (2012), Art. 21 (14 pages).