Int J Interact Des Manuf (2017) 11:503–514
DOI 10.1007/s12008-016-0300-0

ORIGINAL PAPER

Spectral-based mesh segmentation

Daniel Mejia[1] · **Oscar Ruiz-Salguero**[1] · **Carlos A. Cadavid**[1]

Received: 15 December 2015 / Accepted: 4 January 2016 / Published online: 18 January 2016

Abstract In design and manufacturing, mesh segmentation is required for FACE construction in boundary representation (B-Rep), which in turn is central for feature-based design, machining, parametric CAD and reverse engineering, among others. Although mesh segmentation is dictated by geometry and topology, this article focuses on the topological aspect (graph spectrum), as we consider that this tool has not been fully exploited. We pre-process the mesh to obtain a edge-length homogeneous triangle set and its Graph Laplacian is calculated. We then produce a monotonically increasing permutation of the Fiedler vector (2nd eigenvector of Graph Laplacian) for encoding the connectivity among part feature sub-meshes. Within the mutated vector, discontinuities larger than a threshold (interactively set by a human) determine the partition of the original mesh. We present tests of our method on large complex meshes, which show results which mostly adjust to B-Rep FACE partition. The achieved segmentations properly locate most manufacturing features, although it requires human interaction to avoid over segmentation. Future work includes an iterative application of this algorithm to progressively sever features of the mesh left from previous sub-mesh removals.

Keywords Fiedler vector · Mesh Laplacian · Mesh segmentation · Spectral analysis

✉ Oscar Ruiz-Salguero
oruiz@eafit.edu.co

Daniel Mejia
dmejiap@eafit.edu.co

Carlos A. Cadavid
ccadavid@eafit.edu.co

[1] Laboratorio de CAD CAM CAE, Universidad EAFIT, K 49 #7 Sur-50, Medellin, Colombia

Abbreviations

$\boldsymbol{M}$ Triangular mesh of a connected 2-manifold embedded in $\mathbb{R}^3$ composed by the set of points $X = \{x_1, x_2, \ldots, x_n\}$ and the set of triangles $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_p\}$.

$\boldsymbol{E}$ Set of the edges $\{e_1, e_2, \ldots, e_n\}$ of all the triangles $\mathcal{T}$ describing the connectivity of M.

$\boldsymbol{G}$ Graph representation of M consisting of the pair (X, E).

$\boldsymbol{W}$ Weighted adjacency matrix of G of size $n \times n$.

$\boldsymbol{D}$ $n \times n$ diagonal matrix where D_{ii} is equal to the degree (weighted neighborhood size) of the vertex x_i.

$\boldsymbol{L}$ Laplacian matrix of G defined as $L = D - W$.

λ_i i^{th} eigenvalue of the matrix L (sorted in ascending order).

$\boldsymbol{u_i}$ Corresponding eigenvector of λ_i.

$\boldsymbol{u'_2}$ Second eigenvector of L sorted in ascending order.

$\boldsymbol{V}$ Indices of the vertices of G in concordance with $\boldsymbol{u'_2}$ (re-labeling).

$\boldsymbol{d}$ Second differences of $\boldsymbol{u'_2}$ with respect to V.

$\boldsymbol{d'}$ Filtered version of d.

$\boldsymbol{t_i}$ i^{th} local maximum of the set of all local maxima of d' sorted in descending order.

$\boldsymbol{\mathcal{M}}$ A connected and oriented Riemannian 2-manifold embedded in $\mathbb{R}^3$.

$\frac{\partial}{\partial y^i}$ Tangent vectors defining a local coordinate system at a point $p \in \mathcal{M}$.

$\boldsymbol{g}$ Metric tensor which defines an inner product on $\mathcal{M}$ where g_{ij} is the local inner product between the coordinates $\frac{\partial}{\partial y^i}$ and $\frac{\partial}{\partial y^j}$ at a point $p \in \mathcal{M}$.

∇	Gradient operator acting on the surface defined by $\mathcal{M}$.
div	Divergence operator acting on the surface defined by $\mathcal{M}$.
Δ	Laplace-Beltrami operator on manifolds defined as $\Delta = -(\text{div} \circ \nabla)$.

1 Introduction

Mesh segmentation has become an important task in many CAD CAM CAE areas. Its applications range widely in topics such as mesh animation [1,2], surface parameterization [3,4], mesh compression [5,6] and shape processing [7,8].

The problem is as follows: given the mesh of a (connected) surface, break it up into a set of smaller sub-meshes which together compose the initial mesh. Solving this problem becomes crucial when some procedure must be carried on surfaces with particular properties, such as *developability* for parameterization or decomposition into primitive shapes for shape processing or mesh animation.

Usually a geometric approach is followed to face this problem as geometric constraints can be easily defined on the surface based on the user desired result. However, there are cases in which geometry transitions are not strong enough to trigger a mesh partition, while the topology of the mesh indicates a clear discontinuity, that should result in a mesh partition.

Since a mesh can be seen as an undirected graph, study of graph topology and methods of graph partitioning can be almost immediately extrapolated to mesh segmentation. Graph and mesh Laplacians have been a topic of extensive research and their spectra have shown to be a powerful tool for segmentation [9–12].

Finally, mixed approaches have also been proposed. Mixed approaches define schemes based on graph theory, but including geometric data. For example, finite elements schemes of the Laplace-Beltrami operator lead to the cotangent weights method [13]. These schemes are strong in terms of both geometric and topologic information which makes them so reliable for mesh processing.

Despite the advantages that mixed schemes may have, we believe that there should be first a deep understanding of relevant topological information for mesh segmentation.

Since there are voids in the understanding of Topological and Geometrical mesh properties, human interaction is still required to supervise these automatic mesh segmentation methods. In this article we illustrate how the second eigenvector of the Laplacian (Fiedler vector) carries important features for graph and mesh segmentation. We use the monotonically increasing permutation of the Fiedler vector to re-label the mesh nodes. This re-labeling reflects strength of connectivity among graph components and therefore determines a partition of the mesh.

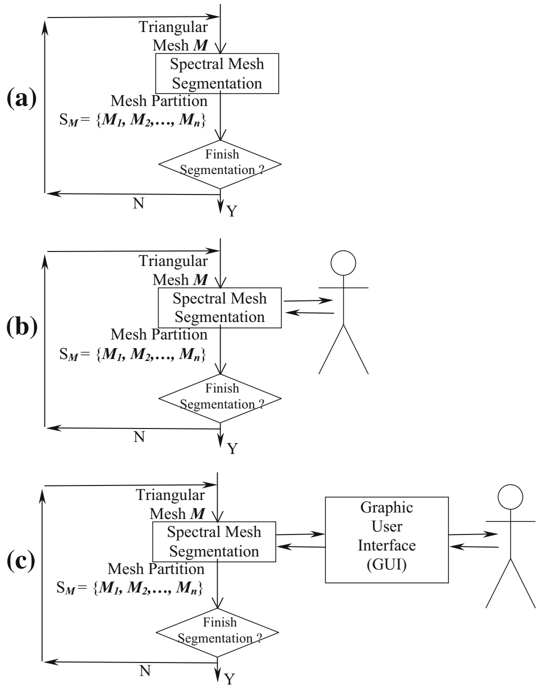

(a)
Triangular Mesh M
Spectral Mesh Segmentation
Mesh Partition
$S_M = \{M_1, M_2, \ldots, M_n\}$
Finish Segmentation ?
N
Y
(b)
Triangular Mesh M
Spectral Mesh Segmentation
Mesh Partition
$S_M = \{M_1, M_2, \ldots, M_n\}$
Finish Segmentation ?
N
Y
(c)
Triangular Mesh M
Spectral Mesh Segmentation
Graphic User Interface (GUI)
Mesh Partition
$S_M = \{M_1, M_2, \ldots, M_n\}$
Finish Segmentation ?
N
Y

Fig. 1 Scenarios of mesh segmentation

Figure 1 shows 3 scenarios of application of our mesh partition algorithm. Figure 1a presents a fully automatic scenario. Figure 1b displays our algorithm steered by a human operator, via direct setting of the mesh separation parameters. Figure 1c depicts our algorithm parameters being controlled via a Graphic User Interface (GUI) which would set the mesh separation parameters. In all 3 cases, some sub-meshes are removed from the source mesh in iteration i. The remaining mesh becomes the source mesh for iteration $i + 1$. The iterations proceed until the input mesh is null or should not be split further.

The remainder of this article is organized as follows: Sect. 2 reviews the relevant literature. Section 3 presents our segmentation approach. Section 4 discusses results for many datasets. Section 5 concludes the article and introduces what remains for future work.

2 Literature review

Mesh segmentation has been an important research topic for computer design. Different algorithms have been proposed for segmentation based on different foundations of many areas such as statistics, optimization, graph theory and many others. An algorithm lies in one of the following classes depending on which data is relevant for segmentation: (i) geometry-based segmentation, (ii) topology-based segmentation and (iii) mixed-approach segmentation. A brief discussion of current state of the art methods is presented below following the proposed classification.

2.1 Geometry-based segmentation

Geometry-based methods extract geometric data from the mesh such as euclidean and geodesic distances or curvatures. Then, a clustering algorithm is usually applied assuming that geometrically similar points likely belong to the same cluster. In [6] a k-means algorithm is directly applied to the geometry data of the mesh and therefore needs a post-processing algorithm (which they call multiple principal plane analysis) for smooth definition of boundaries between sub-meshes. In [4] tangent curvatures are computed by curve fitting before clustering. In [14] curvatures and normals are computed and then a Student-t mixture model is used for the partition.

Region-growing methods are expansive clustering techniques that define seed points or faces on the mesh and as their name suggest, expand until some geometric constraint is violated. In [15] an angle variation threshold between faces is defined for the algorithm and in [16] a variational formulation is proposed for segmentation by fitting quadric surfaces.

Other statistical techniques have been applied for mesh segmentation. By generating some random field on a mesh, the distribution of some objective function can be used for partitioning. In [17] several segmentation algorithms generate random fields on the surface which are later evaluated by their cost function. In [18] an energy function is first defined and random fields are applied to the mesh. The distribution of the energy field in terms of the random fields is then used to divide the surface.

For these algorithms, we remark two potential drawbacks: (i) most of them are not fully deterministic which does not guarantee the same results for the same surface and (ii) as we mentioned before, topology information is usually discarded which might be important in several cases.

Learning approaches try to replicate shapes from previously learned geometries. This in fact requires the algorithm to be calibrated first using already segmented meshes as can be seen in [19–21]. However, these methods require lots of datasets for training that must be similar to the mesh which limits the algorithm and as can be noted, training meshes must be segmented somehow.

2.2 Topology-based segmentation

Topology-based algorithms on the other hand, rely on the features lying on the structure of the mesh without considering geometric features. Recall that a mesh has a graph representation and the topologic properties of this graph can be used for segmentation. However, some assumptions must be first made on the mesh sampling since the same surface may be represented by different connectivities.

The motorcycle algorithm first proposed by Eppstein et al. [22] consists of following a *particle* across straight paths that start at what they define an *extraordinary vertex* and end at vertices visited by other *particles*. Gunpinar et al. [23,24] have presented several approaches based on this idea though, this method is currently limited to quadrilateral meshes.

In [17] segmentation is achieved by simultaneous clustering of similar meshes. However, this task requires maximum correspondence between faces which is not usual in CAD models due to meshing procedures and refinements. In [8] the authors propose a segmentation by labeling vertices based on convexity flags.

Graph cuts have been successfully used for mesh segmentation as seen in [25], where an interactive approach is followed by letting the user to draw strokes on areas where he wants the partition.

Reeb graphs have been also explored for mesh segmentation. After a field is defined on the mesh graph, the reeb graph is computed. Correct choice of the field is critical for an adequate segmentation. For example, in [26,27] the authors propose geodesic-based fields, while in [28] eigenvectors are used instead.

Algorithms that rely solely on the mesh topology are less common in the literature due to the sampling constraints and the lack of geometric features which might be important in many applications leading to the next class of algorithms.

2.3 Mixed-approach segmentation

A mixed approach can be followed by extracting features from both topology and geometry. For example, in [29] an improvement of the motorcycle algorithm is presented. By assigning different velocities to *particles* the geometry of the surface is taken into account however, the limitation of quadrilateral meshes is still present.

In [7] slippage analysis is used for mesh segmentation. Localities of points are used to compute some measures such as slippage and curvature. Depending on the measured features, primitive shapes such as planes, spheres and cylinders are recognized. The problem with this method is that some surfaces may present complex shapes not recognized by the algorithm. Additionally, since the method is presented for clouds of points, the algorithm is very sensitive to neighbor sizes.

An important operator has been widely used for mesh segmentation: the Laplacian operator. The definition of most mesh Laplacian follow classic graph Laplacian definitions naturally which strongly encodes topologic features. Geometric features are also considered by weighting schemes defined on the graph of the mesh. In [12] and [30] interactive approaches are presented where eigenvectors of the Laplacian are chosen for the segmentation and in [9] eigenvectors are automatically chosen by an empirical criteria and a k-means algorithm is then applied on the selected subset of eigenvectors.

Harmonic functions of the Laplacian operator have been also used. After some region is selected, dirichlet conditions are introduced to the operator and a linear system of equations arises. In [31] an interactive algorithm is presented where the user draws strokes across desired boundaries and in [32] a similar algorithm is presented for segmentation of dental meshes. Heat kernels have been also explored for mesh segmentation [33,34] where dirac delta initial conditions are imposed.

2.4 Conclusions of the literature review

Significant uncertainty currently exists about how the topology of the mesh encodes relevant features for segmentation. This uncertainty usually leads most researchers to address the mesh segmentation problem from the geometry-based approach. In contrast, we consider that spectral analysis presents large potential for segmentation [9,12]. However, heavy user-interactivity is required and parameterizations of the algorithms are based on empirical results. Figure 2 shows the result of application of state-of-art software by GeoMagic TM for geometry-based segmentation and parameterization at the same time. To obtain this result, a significant amount of user actions is required, to correct non-manifold situations and to eliminate patches with extreme shape factor. We contend that by separating segmentation from parameterization, it is possible to obtain larger sub-meshes. In addition, we will show that using topologic over of geometric segmentation would better highlight functional features of the object.

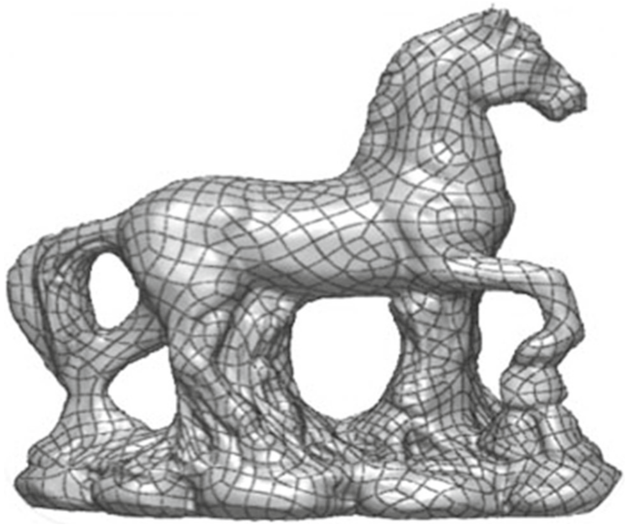


Fig. 2 Result of mesh segmentation/parameterization with GeoMagic TM. Intensive user editing is required to correct non-manifold situations and patches with extreme shape factor

In this article we illustrate how the Fiedler parameters encode connectedness properties of the graph mesh. Based on these results, an intuitive method is presented for exploiting these properties. A topology-based algorithm is introduced by constructing the classic graph Laplacian with constant weights and successful segmentation of homogeneous meshes is achieved. The limitation to homogeneous meshes can be partially overcome by extending the scheme to a mixed approach but the pure topologic nature of the algorithm gives faster and computationally stable results. Also, since the method follows an intuitive result from spectral analysis, a deeper understanding of topology-based implicitly arises and opens the doors for further research.

Our proposed algorithm produces segmentations that obey the connectivity of a homogeneously connected mesh (i.e. with quasi-uniform triangle edge length). Notice that such an algorithm requires a human steering (likely to be interactive), to reject over-fragmentation, or to set the threshold of Fiedler vector discontinuity, that triggers split of a sub-mesh from the source mesh.

3 Methodology

We consider triangular meshes M which are connected, 2-manifolds embedded in $\mathbb{R}^3$. M is defined by a set points $X = \{x_1, x_2, \ldots, x_n\}$ which together describe the geometry of M and a set of triangles $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_p\}$. Each triangle is a sequence of edges $\tau_i = (e_a, e_b, e_c)$ where each e_j belongs to a set of edges $E = \{e_1, e_2, \ldots, e_m\}$. This structure describes the connectivity on M.

$G = (X, E)$ is therefore the (undirected) graph representation of M where X is seen as the set of vertices of the graph and E is the set of edges. Since M is connected, so is G.

The problem of mesh segmentation consists of dividing M into k disconnected components. For solving this problem we propose an algorithm that re-labels the vertices of the graph based on the Fiedler vector. Figure 3 presents the steps of the algorithm. Some preliminary key results of spectral theory are briefly discussed first.

3.1 Graph Laplacian

In spectral graph theory, the adjacency matrix W of G is defined as:

$$W_{ij} = \begin{cases} w_{ij} & \text{if } (x_i, x_j) \in E \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

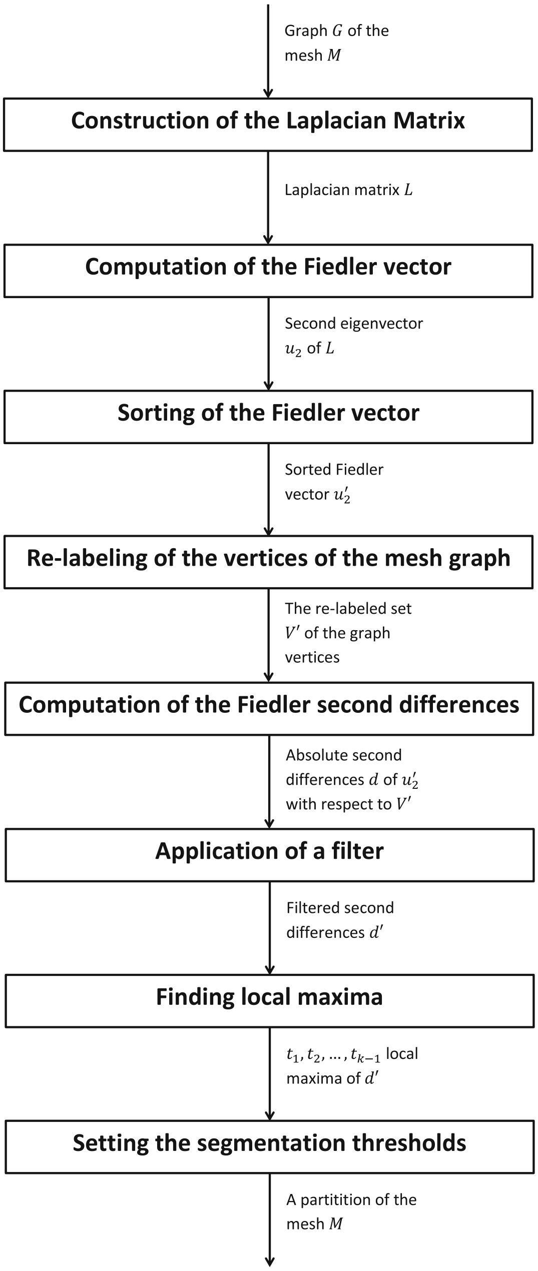

Graph G of the mesh M
Construction of the Laplacian Matrix
Laplacian matrix L
Computation of the Fiedler vector
Second eigenvector u_2 of L
Sorting of the Fiedler vector
Sorted Fiedler vector u_2'
Re-labeling of the vertices of the mesh graph
The re-labeled set V' of the graph vertices
Computation of the Fiedler second differences
Absolute second differences d of u_2' with respect to V'
Application of a filter
Filtered second differences d'
Finding local maxima
$t_1, t_2, \dots, t_{k-1}$ local maxima of d'
Setting the segmentation thresholds
A partitition of the mesh M

Fig. 3 Diagram of the algorithm

where w_{ij} is the adjacency weight between x_i and x_j. For our algorithm we take $w_{ij} = 1$ guaranteeing that only the topology of the graph mesh is considered.

If D is a diagonal matrix with entries $D_{ii} = \sum_j w_{ij}$, then D is the degree matrix of G. The Laplacian matrix of G is therefore defined as $L = D - W$. Let $f : X \rightarrow \mathbb{R}$ with $f(x_i) = f_i$ be a field defined on the vertices of the graph, then Lf acts locally on each vertex in the following manner:

$$(Lf)_i = \sum_j w_{ij} \left(f_i - f_j \right) \tag{2}$$

Equation (2) defines another field Lf on the vertices of the graph where each x_i gets assigned the weighted differences between f_i and its neighboring field i.e. all f_j such that $x_j \in N(x_i)$.

3.2 Fiedler vector

We consider the set of eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ of L and their corresponding eigenvectors $\boldsymbol{u_1}, \boldsymbol{u_2}, \dots, \boldsymbol{u_n}$. There are some important results from graph theory on these eigenvalues and eigenvectors [35,36]:

1. The first eigenvalue λ_1 is equal to 0 and its corresponding eigenvector $\boldsymbol{u_1}$ is the constant vector i.e. $u_1(x_i) = 1$.
2. The second eigenvalue λ_2 is known as the Fiedler value (or algebraic connectivity) and its corresponding eigenvector $\boldsymbol{u_2}$ is known as the Fiedler vector.
3. Since G is connected, $\lambda_2 > 0$ and therefore $\boldsymbol{u_2}$ is orthogonal to the constant vector i.e. $\langle \boldsymbol{u_1}, \boldsymbol{1} \rangle = 0$.
4. The Fiedler vector solves the following optimization problem:

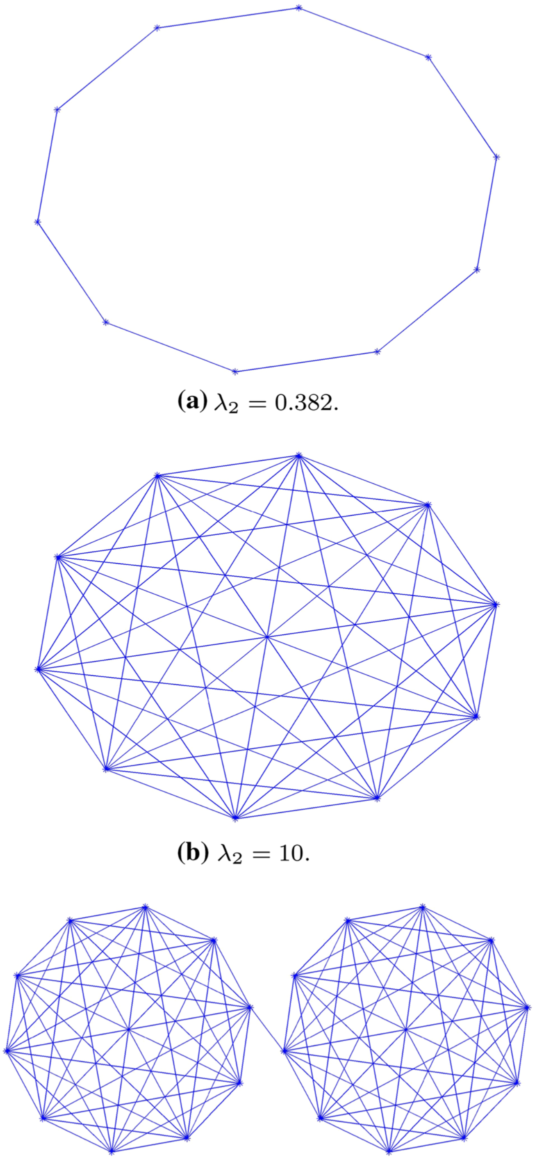
$$\boldsymbol{u_2} = \underset{\boldsymbol{u} \perp \boldsymbol{1}}{\operatorname{argmin}} \frac{\boldsymbol{u}^T L \boldsymbol{u}}{\boldsymbol{u}^T \boldsymbol{u}} \tag{3}$$

The algebraic connectivity is highly related to the connectedness of the graph. Figure 4 illustrates how the second eigenvalue of the Laplacian matrix can give us an idea of such connectedness. The loop graph presents less algebraic connectivity than the full graph since it requires less cuts to divide the graph. However, the two full graphs connected by a single edge show the smallest algebraic connectivity since cutting that edge is enough to split the graph despite the high connectedness of both graphs.

Figure 5 shows the Fiedler vector for the connected full graphs (Fig. 4c). Here the Fiedler vector is a positive function in terms of the current vertex labeling. Also, low connectivity regions [edge (v_{10}, v_{11})] show high changes in the Fiedler vector compared with high connectivity regions (other vertices). This behaviour is expected since Eq. (3) will see less penalization in low connectivity regions, and motivates our algorithm: by re-labeling the vertices of the graph such that the Fiedler vector becomes a positive function in terms of the re-labeled vertices, we can find low and high connectivity regions based on this intuition and set a segmentation criterion.

3.3 Laplace-Beltrami operator

Let $\mathcal{M}$ be an oriented Riemannian 2-manifold embedded in $\mathbb{R}^3$. $\mathcal{M}$ is connected. The Laplace-Beltrami operator $\Delta : L^2(\mathcal{M}) \rightarrow L^2(\mathcal{M})$ on $\mathcal{M}$ is defined as [37]:



(a) $\lambda_2 = 0.382.$

(b) $\lambda_2 = 10.$

(c) $\lambda_2 = 0.169.$

Fig. 4 Some graphs and their algebraic connectivity: **a** loop graph with ten vertices, **b** full graph with ten vertices and **c** two full graphs of ten vertices connected by an edge

$$\Delta\phi = -(\mathrm{div} \circ \nabla)\phi \tag{4}$$

where $\phi \in L^2(\mathcal{M})$, ∇ is the gradient operator on $\mathcal{M}$ defined as:

$$\nabla\phi = \sum_i g^{ij} \frac{\partial\phi}{\partial \boldsymbol{y}^i} \frac{\boldsymbol{\partial}}{\boldsymbol{\partial y}^j} \tag{5}$$

and div is the divergence operator defined as:

$$\mathrm{div}\boldsymbol{Y} = \frac{1}{\det(g)} \sum_i \frac{\partial}{\partial \boldsymbol{y}^i} \left(Y^i \sqrt{\det(g)} \right) \tag{6}$$

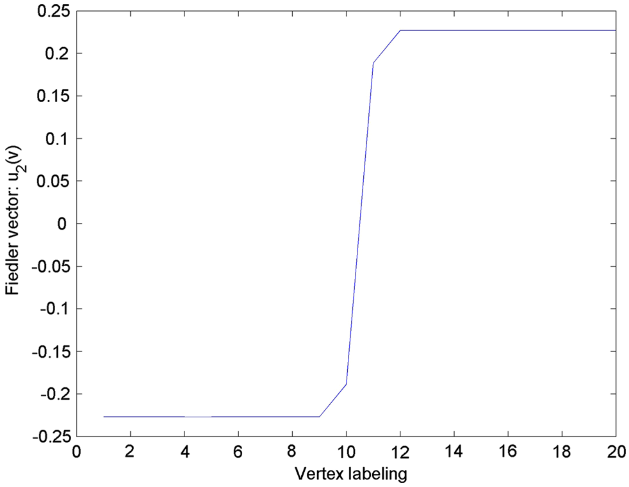


Fig. 5 Fiedler vector for the two full graphs (Fig. 4c). Vertices labeled from 1 to 10 belong to the first full graph and vertices labeled from 11 to 20 belong to the second full graph. The vertices 10 and 11 connect the graphs

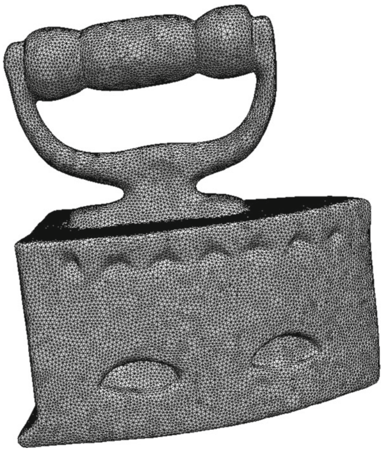
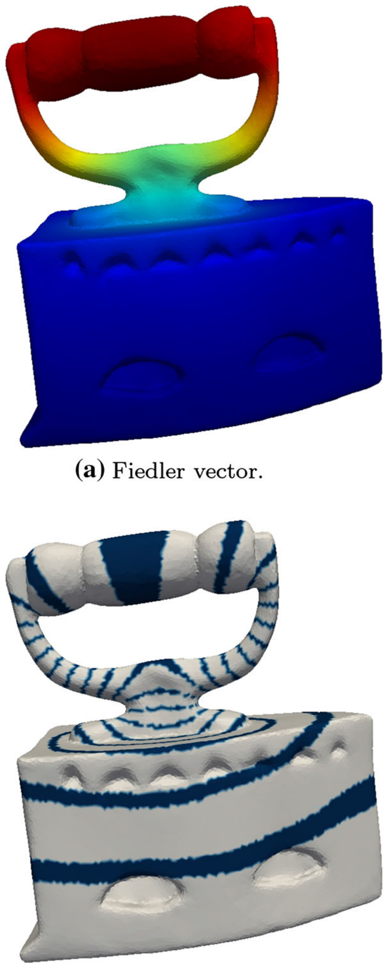


Fig. 6 Homogeneous triangular mesh of the iron model

g is the metric tensor of $\mathcal{M}$, $g^{ij} = (g_{ij})^{-1}$ are the components of the inverse of the metric tensor, $\boldsymbol{Y} \in TM$ is a vectorial field defined on the manifold (local tangent planes) and Y^i are its corresponding components in local coordinates $\frac{\boldsymbol{\partial}}{\boldsymbol{\partial y}^i}$.

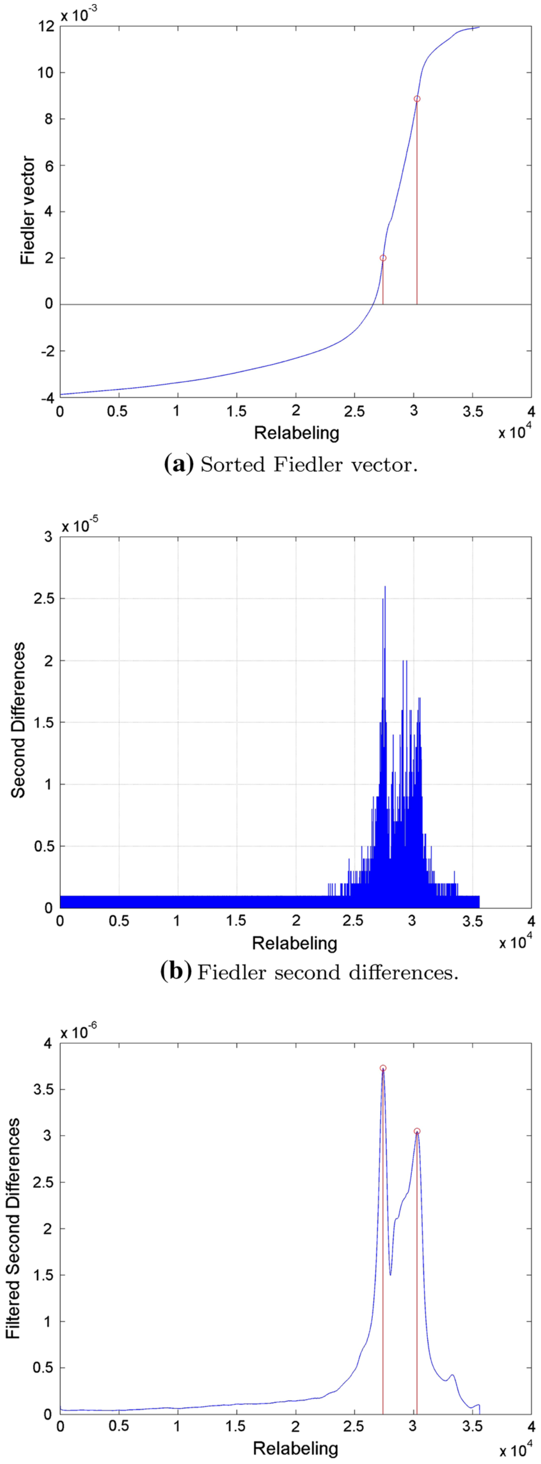
The Laplace-Beltrami operator is a generalization of the standard Laplacian taken to manifolds. From this point of view, the differential equation $\Delta\phi = \lambda\phi$ on the manifold can be seen as an analogue of the Helmhotz equation. The Helmhotz equation models the time-independent component of the wave equation on a given domain. Schemes for discretization of such operator on triangular surfaces have been presented in the literature [13,38] resulting in weighted Laplacians of the given mesh graphs. This result motivates our work since the consideration of an homogeneous mesh leads to a scaled solution of Eq. (3).



(a) Fiedler vector.

(b) Fiedler isolines.

Fig. 7 Fiedler vector on the iron mesh: **a** the field goes from *blue* (lower values) to *red* (higher values) and **b** isolines of such of field are drawn in *blue* (colour figure online)



(a) Sorted Fiedler vector.

(b) Fiedler second differences.

(c) Filtered differences.

Fig. 8 Re-labeling of the vertices (x axis) and corresponding function on the iron mesh: **a** Fiedler vector, **b** second differences and **c** filtered second differences. *Lines* in *red* show the cut points for segmentation (colour figure online)

3.4 Algorithm description

The idea of our approach consists in dividing the mesh into components with uniform connectivity. As we illustrated in Fig. 5, low connectivity areas present higher changes of the Fiedler vector with respect to some labeling of the vertices of the graph. These changes can be described by discrete differences on the Fiedler vector. Also, components with uniform connectivity have uniform differences and high changes in the discrete differences imply relevant topologic changes in the mesh graph. Therefore, high absolute differences of second order on the Fiedler vector with respect to a specific labeling of the vertices define good cut points for segmentation.

The vertices of the mesh must be re-labeled such that the Fiedler vector is an increasing function in terms of the re-labeling since the segmentation will be defined by cutting thresholds on the Fiedler vector. Below the proposed algorithm (Fig. 3) is described:

1. **Construction of the Laplacian matrix:** Build the sparse Laplacian matrix L of the graph representation G of the connected mesh M as defined in Sect. 3.1.

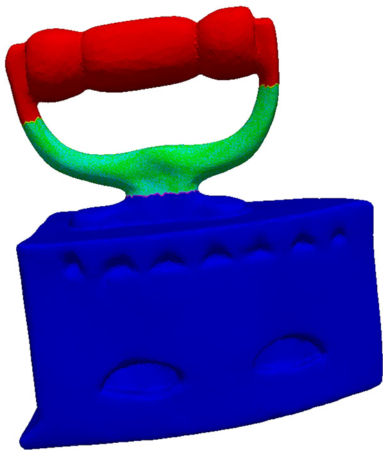


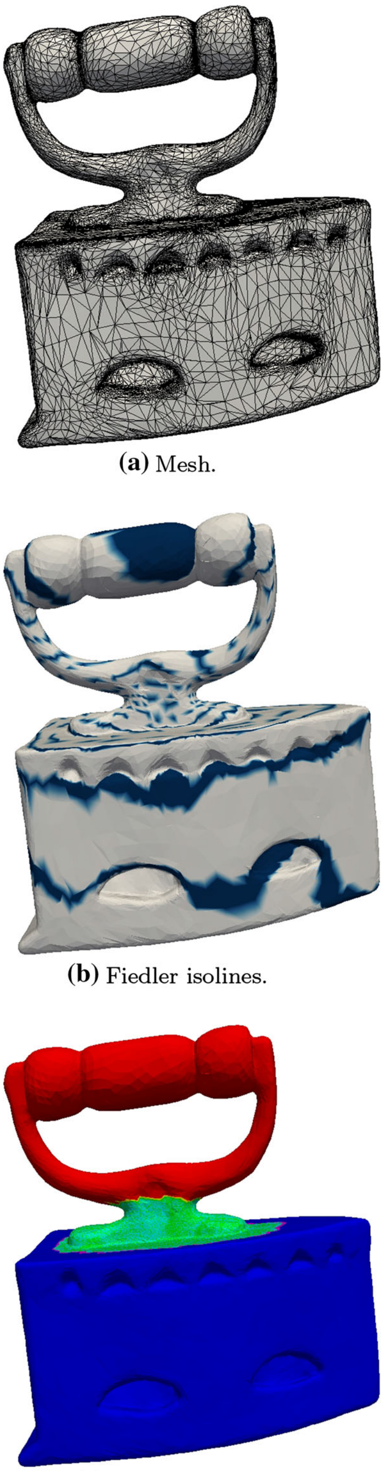
Fig. 9 Segmentation of the iron mesh

2. **Computation of the Fiedler vector:** Compute the Fiedler vector $\boldsymbol{u_2}$ of L by using some eigensolver.
3. **Sorting of the Fiedler vector:** Compute $\boldsymbol{u'_2}$ as a sorting of $\boldsymbol{u_2}$ in ascending order.
4. **Re-labeling of the vertices of the mesh:** Compute V where $V = \{v_1, v_2, \ldots, v_n\}$ is a re-labeling of X such that $v_i = x_k, u'_2(v_i) \leq u'_2(v_{i+1})$ and $v_i \neq v_j, \forall i \neq j$.
5. **Computation of the Fiedler second differences:** Compute the absolute second differences $d(v')$ of $\boldsymbol{u'_2}$ with respect to v'. This is equivalent to compute $d_i = |2u'_2(v'_i) - u'_2(v'_{i-1}) - u'_2(v'_{i+1})|$.
6. **Application of a filter:** A low-pass filter must be applied to d since the sampling of the mesh can lead to noise in the computed second differences. Compute the filtered second differences d'.
7. **Finding local maxima:** Compute $t_1, t_2, \ldots, t_{k-1}$ where t_1 is the global maximum of $d'(v')$ and each t_i is a subsequent (local) maximum (recall that k is the desired number of components to partition the mesh).
8. **Setting the segmentation thresholds:** The segmentation of the mesh is achieved by setting the thresholds $u'_2((d')^{-1}(t_i))$ on the Fiedler vector.

4 Results

We now present some results of our algorithm applied to several datasets. In each case the Fiedler vector was computed by the Implicitly Restarted Arnoldi Iteration algorithm which comes implemented in the ARPACK package [39]. Also, a moving average was implemented for the filtering of the second order differences with window sizes ranging between 1–10 %.

Figure 6 shows the triangular mesh of the iron model. This mesh consists of 35,582 points and 71,164 triangles. Figure 7 shows the Fiedler vector of the iron mesh. Topology of the graph of such (homogeneous) mesh still preserves many geometric properties of the surface. Isolines of the Fiedler vector (Fig. 7b) show how the highly connected areas present low changes in the Fiedler vector with respect to geometry while the lower ones present high changes.



(a) Mesh.

(b) Fiedler isolines.

(c) Segmentation.

Fig. 10 Segmentation of a non-homogeneous mesh of the iron model

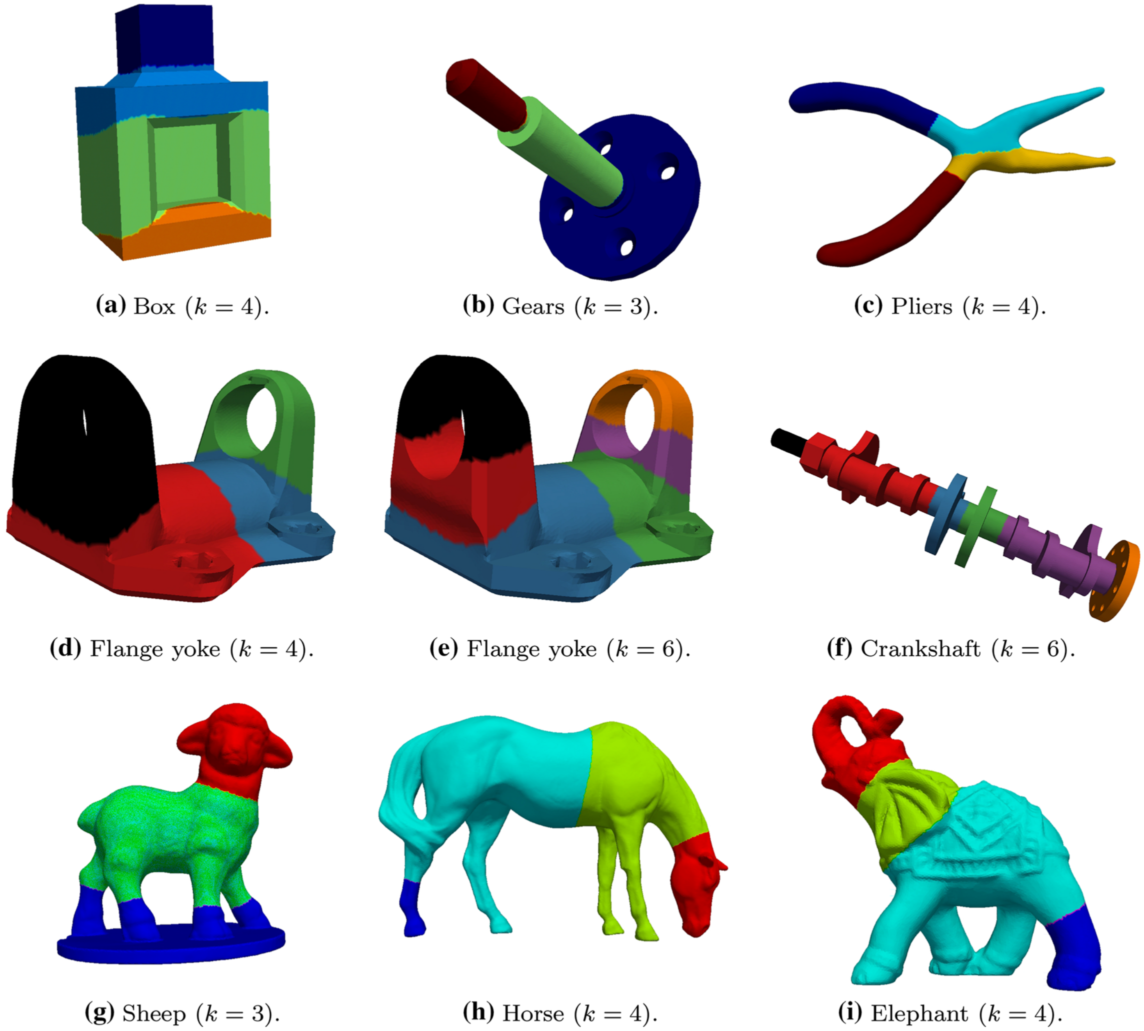
Table 1 Appraisal of the results of the segmentation algorithm applied to several datasets

Figure	Dataset	# of points	Commentary
Figure 11a	Box	31945	The segmentation correctly finds high topological changes but segmentation boundaries are not aligned with sharp edges
Figure 11b	Gears	24624	The algorithm achieves an intuitive segmentation given the fact that high topological changes happen at sharp edges
Figure 11c	Pliers	14250	The segmentation separates the grip from the jaws. Each jaw is also separated as well as the grip
Figures 11d, e	Flange yoke	16522	A segmentation of the mesh into four sub-meshes is enough to capture the topological changes. Segmenting into more sub-meshes may present irrelevant information
Figure 11f	Crankshaft	45185	Some of the segmentation boundaries are not aligned with the sharp edges of the surface given the fact that only topological information is taken into account
Figure 11g	Sheep	68741	The base and the head are separated from the rest of the body
Figure 11h	Horse	99463	The body is separated into two pieces and the head is separated from the whole body. A leg is separated from the rest of the body since it represents a high topological change in the direction of the Fiedler vector
Figure 11i	Elephant	32707	Alike the horse dataset, a leg is separated as well as the head from the body. Additionally, The algorithm separates the ears from the rest of the head

The Fiedler vector is then sorted as described in Sect. 3.4 and re-labeling of the vertices is made. Figure 8 shows the re-labeling results for the iron. As we pointed in Sect. 3 the sorted Fiedler vector shows higher changes in specific regions (Fig. 8a) which correspond to relative changes of the Fiedler isolines in Fig. 7b. The second differences plot (Fig. 8b) shows a concentration of several peaks near desired cut points. Selecting thresholds from this signal would lead to several disconnected components near the same isoline which is not desired. Therefore this signal must be filtered first as described in Sect. 3.4. The filtered signal is presented in Fig. 8c. Such filtering allows the algorithm to automatically detect good cutting isolines. By setting $k = 3$, global maxima are computed in the filtered differences resulting in two thresholds (red lines) for the Fiedler vector which will divide the surface by the corresponding isolines. These thresholds coincide with high changes in the slope of the sorted Fiedler vector as seen in Fig. 8a. The resulting segmentation is presented in Fig. 9. Notice how the resulting partition divided the mesh into two high connectivity regions (blue and red) and a low connectivity region (green).

In contrast with the segmentation of a homogeneous mesh, Fig. 10 presents results for a non-homogeneous mesh of the same model. The Fiedler vector of the graph mesh does not follow adequately the geometry of the surface (Fig. 10a) due to the non-homogeneous sampling as seen in Fig. 10a and as a consequence, the connectivity of the graph does not follow correctly our intuition of the geometry of the mesh anymore. Additionally, isolines present undesired behaviours along the surface which would result into components with more irregular boundaries. The segmentation of this mesh under our approach is presented in Fig. 10c. Notice that the segmentation is less intuitive since the topology of the mesh graph is more discordant with the geometry of the iron model and boundaries of disconnected components are located at more geometrically random places. These results illustrate the importance of the homogeneity of the mesh for adequate results of the proposed algorithm.

Table 1 presents the results of our algorithm applied to several datasets usually used in CAD applications or Computer Graphics. Since the algorithm only takes into account the topology of the mesh, it is not expected to partition the mesh by its sharp edges. However, the achieved segmentation agrees with some of these sharp edges as can be seen in the gears and crankshaft datasets showing the importance of the topology even in these cases. It is important to emphasize the importance of the user interaction since different results can arise depending on the selection of the parameter k as seen in the flange yoke dataset. The segmentation of the rest


(a) Box ($k = 4$). **(b)** Gears ($k = 3$). **(c)** Pliers ($k = 4$).

(d) Flange yoke ($k = 4$). **(e)** Flange yoke ($k = 6$). **(f)** Crankshaft ($k = 6$).

(g) Sheep ($k = 3$). **(h)** Horse ($k = 4$). **(i)** Elephant ($k = 4$).

Fig. 11 Results of our algorithm for several datasets. Princeton datasets [11 (**a**), 11 (**b**), 11 (**c**)]. Laboratorio de CAD CAM CAE U. EAFIT datasets [11 (**g**), 11 (**h**), 11(**i**)]

of the meshes presents intuitive results which illustrates the fact that high changes in the Fiedler vector can be used as boundaries between different shapes.

5 Conclusions

We illustrated how the Fiedler vector encodes some features of the surface concerning to the connectivity of the mesh graph. These features allowed us to develop a simple algorithm for automatic segmentation of homogeneous meshes. Keeping the discussion at the topologic level allowed us to intuitively show several important characteristics of the second eigenvector of the mesh graph Laplacian.

For the segmentation a re-labeling of the vertices of the mesh is made in such a way the vector Fiedler is increasing in terms of the re-labeled vertices. The algorithm is still applicable to general graphs given the fact that geometry is not taken into account while using constant weights makes the algorithm faster and more stable computationally. However, this advantage comes with a cost requiring the mesh to be homogeneous as we illustrated in Sect. 4. Adequate results were presented for many complex datasets usually used in the context of CAD CAM CAE and Computer Graphics applications. These results proved to be in concordance with user intuitive partitions which is very important since only topologic aspects are considered. Possible future work could be the adaptation of the method to non-homogeneous meshes. Also, future consideration of the geometry could lead to better results.

The algorithm presented by us would typically work within a Divide— and—Conquer or Iterative scheme. In this manner, each run would subtract some sub-meshes of the source mesh, gradually reducing the size of the problem. Each iteration would take the input of a human user to steer each removal iteration and/or to set the values of discontinuities in the Fiedler vector permutation, which would in turn cause mesh-fragmentation.

Acknowledgements The box, gears and pliers datasets were taken from the Princeton benchmark [40]. The iron, sheep, horse and elephant datasets were scanned at Laboratorio de CAD CAM CAE at Universidad EAFIT (Colombia).

References

1. Vasilakis, A.A., Fudos, I.: Pose partitioning for multi-resolution segmentation of arbitrary mesh animations. Comput. Graph. Forum **33**(2), 293–302 (2014). doi:10.1111/cgf.12327
2. Arcila, R., Cagniart, C., Hétroy, F., Boyer, E., Dupont, F.: Segmentation of temporal mesh sequences into rigidly moving components. Graph. Models **75**(1), 10–22 (2013). doi:10.1016/j.gmod.2012.10.004
3. Pal, P.: Fast freeform hybrid reconstruction with manual mesh segmentation. Int. J. Adv. Manuf. Technol. **63**(9–12), 1205–1215 (2012). doi:10.1007/s00170-012-3986-6
4. Miao, Y.W., Feng, J.Q., Xiao, C.X., Peng, Q.S., Forrest, A.: Differentials-based segmentation and parameterization for point-sampled surfaces. J. Comput. Sci. Technol. **22**(5), 749–760 (2007). doi:10.1007/s11390-007-9088-5
5. Luo, G., Cordier, F., Seo, H.: Compression of 3d mesh sequences by temporal segmentation. Comput. Anim. Virtual Worlds **24**(3–4), 365–375 (2013). doi:10.1002/cav.1522
6. Cheng, S.C., Kuo, C.T., Wu, D.C.: A novel 3d mesh compression using mesh segmentation with multiple principal plane analysis. Pattern Recognit. **43**(1), 267–279 (2010). doi:10.1016/j.patcog.2009.05.016
7. Yi, B., Liu, Z., Tan, J., Cheng, F., Duan, G., Liu, L.: Shape recognition of cad models via iterative slippage analysis. Comput. Aided Des. **55**, 13–25 (2014). doi:10.1016/j.cad.2014.04.008
8. Tao, S., Huang, Z., Ma, L., Guo, S., Wang, S., Xie, Y.: Partial retrieval of cad models based on local surface region decomposition. Comput. Aided Des. **45**(11), 1239–1252 (2013). doi:10.1016/j.cad.2013.05.008
9. Wang, H., Lu, T., Au, O.K.C., Tai, C.L.: Spectral 3d mesh segmentation with a novel single segmentation field. Graph. Models **76**(5), 440–456 (2014). doi:10.1016/j.gmod.2014.04.009
10. Wang, L., Dong, M.: Multi-level low-rank approximation-based spectral clustering for image segmentation. Pattern Recognit. Lett. **33**(16), 2206–2215 (2012). doi:10.1016/j.patrec.2012.07.024
11. Wang, X., Wu, Z., Luo, P., Wu, H.: An image segmentation method combining L1-sparse reconstruction with spectral clustering. Procedia Eng. **29**, 1387–1391 (2012). doi:10.1016/j.proeng.2012.01.145
12. Miao, Y., Feng, J., Wang, J., Jin, X.: User-controllable mesh segmentation using shape harmonic signature. Progress Nat. Sci. **19**(4), 471–478 (2009). doi:10.1016/j.pnsc.2008.06.024
13. Reuter, M., Biasotti, S., Giorgi, D., Patanè, G., Spagnuolo, M.: Discrete Laplace - Beltrami operators for shape analysis and segmentation. Comput. Graph. **33**(3), 381–390 (2009). doi:10.1016/j.cag.2009.03.005
14. Tsuchie, S., Hosino, T., Higashi, M.: High-quality vertex clustering for surface mesh segmentation using student-t mixture model. Comput. Aided Des. **46**, 69–78 (2014). doi:10.1016/j.cad.2013.08.019
15. Wang, J., Yu, Z.: Surface feature based mesh segmentation. Comput. Graph. **35**(3), 661–667 (2011). doi:10.1016/j.cag.2011.03.016
16. Yan, D.M., Wang, W., Liu, Y., Yang, Z.: Variational mesh segmentation via quadric surface fitting. Comput. Aided Des. **44**(11), 1072–1082 (2012). doi:10.1016/j.cad.2012.04.005
17. Golovinskiy, A., Funkhouser, T.: Randomized cuts for 3d mesh analysis. ACM Trans. Graph. **27**(5), 145:1–145:12 (2008). doi:10.1145/1409060.1409098
18. de Castro, P.M.M., de Lima, L.A.P., Lucena, F.L.A.: Invariances of single curved manifolds applied to mesh segmentation. Comput. Graph. **38**, 399–409 (2014). doi:10.1016/j.cag.2013.12.002
19. Liu, X., Zhang, J., Liu, R., Li, B., Wang, J., Cao, J.: Low-rank 3d mesh segmentation and labeling with structure guiding. Comput. Graph. **46**, 99–109 (2015). doi:10.1016/j.cag.2014.09.019
20. Lv, J., Chen, X., Huang, J., Bao, H.: Semi-supervised mesh segmentation and labeling. Comput. Graph. Forum **31**(7), 2241–2248 (2012). doi:10.1111/j.1467-8659.2012.03217.x
21. Kalogerakis, E., Hertzmann, A., Singh, K.: Learning 3d mesh segmentation and labeling. ACM Trans. Graph. **29**(4), 102:1–102:12 (2010). doi:10.1145/1778765.1778839
22. Eppstein, D., Goodrich, M.T., Kim, E., Tamstorf, R.: Motorcycle graphs: Canonical quad mesh partitioning. In: Proceedings of the Symposium on Geometry Processing, SGP '08, pp. 1477–1486. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2008)
23. Gunpinar, E., Moriguchi, M., Suzuki, H., Ohtake, Y.: Motorcycle graph enumeration from quadrilateral meshes for reverse engineering. Comput. Aided Des. **55**, 64–80 (2014). doi:10.1016/j.cad.2014.05.007
24. Gunpinar, E., Suzuki, H., Ohtake, Y., Moriguchi, M.: Generation of bi-monotone patches from quadrilateral mesh for reverse engineering. Comput. Aided Des. **45**(2), 440–450 (2013). doi:10.1016/j.cad.2012.10.027
25. Brown, S., Morse, B., Barrett, W.: Interactive part selection for mesh and point models using hierarchical graph-cut partitioning. In: Proceedings of Graphics Interface 2009, GI '09, pp. 23–30. Canadian Information Processing Society, Toronto, Ont., Canada (2009)
26. Berretti, S., Bimbo, A.D., Pala, P.: 3d mesh decomposition using reeb graphs. Image Vis. Comput. **27**(10), 1540–1554 (2009). doi:10.1016/j.imavis.2009.02.004. Special Section: Computer Vision Methods for Ambient Intelligence
27. Tierny, J., Vandeborre, J.P., Daoudi, M.: Topology driven 3d mesh hierarchical segmentation. In: Shape Modeling and Applications, 2007. SMI '07. IEEE International Conference on, pp. 215–220 (2007). doi:10.1109/SMI.2007.38
28. Patane, G., Spagnuolo, M., Falcidieno, B.: Reeb graph computation based on a minimal contouring. In: Shape Modeling and Applications, 2008. SMI 2008. IEEE International Conference on, pp. 73–82 (2008). doi:10.1109/SMI.2008.4547953
29. Gunpinar, E., Moriguchi, M., Suzuki, H., Ohtake, Y.: Feature-aware partitions from the motorcycle graph. Comput. Aided Des. **47**, 85–95 (2014). doi:10.1016/j.cad.2013.09.003
30. Zhang, J., Zheng, J., Wu, C., Cai, J.: Variational mesh decomposition. ACM Trans. Graph. **31**(3), 21:1–21:14 (2012). doi:10.1145/2167076.2167079
31. Zheng, Y., Tai, C.L.: Mesh decomposition with cross-boundary brushes. Comput. Graph. Forum **29**(2), 527–535 (2010). doi:10.1111/j.1467-8659.2009.01622.x
32. Zou, B.J., Liu, S.J., Liao, S.H., Ding, X., Liang, Y.: Interactive tooth partition of dental mesh base on tooth-target harmonic field. Comput. Biol. Med. **56**, 132–144 (2015). doi:10.1016/j.compbiomed.2014.10.013
33. Fang, Y., Sun, M., Kim, M., Ramani, K.: Heat-mapping: A robust approach toward perceptually consistent mesh segmentation. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pp. 2145–2152 (2011). doi:10.1109/CVPR.2011.5995695

34. Gebal, K., Bærentzen, J.A., Aanæs, H., Larsen, R.: Shape analysis using the auto diffusion function. In: Proceedings of the Symposium on Geometry Processing, SGP ’09, pp. 1405–1413. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2009)
35. de Abreu, N.M.M.: Old and new results on algebraic connectivity of graphs. Linear Algebra Appl. **423**(1), 53–73 (2007). doi:10.1016/j.laa.2006.08.017
36. Spielman, D.A., Teng, S.H.: Spectral partitioning works: Planar graphs and finite element meshes. Linear Algebra Appl. **421**(2–3), 284–305 (2007). doi:10.1016/j.laa.2006.07.020. Special Issue in honor of Miroslav Fiedler
37. Rosemberg, S.: The Laplacian on a Riemannian Manifold. Cambridge University Press, United Kingdom (1997)
38. Dziuk, G.: Finite elements for the beltrami operator on arbitrary surfaces. In: S. Hildebrandt, R. Leis (eds.) Partial Differential Equations and Calculus of Variations, *Lecture Notes in Mathematics*, vol. 1357, pp. 142–155. Springer Berlin Heidelberg (1988). doi:10.1007/BFb0082865
39. Lehoucq, R., Sorensen, D., Yang, C.: ARPACK Users’ Guide. Soc. Ind. Appl. Math.(1998). doi:10.1137/1.9780898719628
40. Chen, X., Golovinskiy, A., Funkhouser, T.: A benchmark for 3d mesh segmentation. ACM Trans. Graph. **28**(3), 73:1–73:12 (2009). doi:10.1145/1531326.1531379