Computers & Graphics 73 (2018) 47–58

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Hybrid geometry / topology based mesh segmentation for reverse engineering☆

Daniel Mejia[a,b], Oscar Ruiz-Salguero[a], Jairo R. Sánchez[b,*], Jorge Posada[b], Aitor Moreno[b], Carlos A. Cadavid[c]

[a] Laboratorio de CAD CAM CAE, Universidad EAFIT, Cra 49 N 7 Sur-50, Medellín, Colombia
[b] Vicomtech, Paseo Mikeletegi 57, Parque Científico y Tecnológico de Gipuzkoa, Donostia / San Sebastián, España 20009 Spain
[c] Matemáticas y Aplicaciones, Departamento de Ciencias Matemáticas, Universidad EAFIT, Cra 49 N 7 Sur-50, Medellín, Colombia

ARTICLE INFO

Article history:
Received 10 July 2017
Revised 15 March 2018
Accepted 16 March 2018
Available online 28 March 2018

Keywords:
Mesh segmentation
Heat transfer
Reverse engineering
CAD/CAM/CAE

ABSTRACT

Mesh segmentation and parameterization are crucial for Reverse Engineering (RE). Bijective parameterizations of the sub-meshes are a sine-qua-non test for segmentation. Current segmentation methods use either (1) topologic or (2) geometric criteria to partition the mesh. Reported topology-based segmentations produce large sub-meshes which reject parameterizations. Geometry-based segmentations are very sensitive to local variations in dihedral angle or curvatures, thus producing an exaggerated large number of small sub-meshes. Although small sub-meshes accept nearly isometric parameterizations, this significant granulation defeats the intent of synthesizing a usable Boundary Representation (compulsory for RE). In response to these limitations, this article presents an implementation of a hybrid geometry / topology segmentation algorithm for mechanical workpieces. This method locates heat transfer constraints (topological criterion) in low frequency neighborhoods of the mesh (geometric criterion) and solves for the resulting temperature distribution on the mesh. The mesh partition dictated by the temperature scalar map results in large, albeit parameterizable, sub-meshes. Our algorithm is tested with both benchmark repository and physical piece scans data. The experiments are successful, except for the well - known cases of topological cylinders, which require a user - introduced boundary along the cylinder generatrices.

1. Introduction

In the context of Computer Aided Design / Manufacturing / Engineering (CAD/CAM/CAE) and the emerging Industry 4.0 framework, RE encompasses (re-)design, manufacturing, simulation, etc. [1]. Typical RE processes (Fig. 1): (1) tessellate the point cloud of the scanned model, (2) clean the raw triangular mesh (smoothing, filling, non-manifold repair, decimation, etc.), (3) build the Boundary Representation (B-Rep) of the mesh, (4) segment the mesh, (5) fit the resulting partition with parametric surfaces (analytic and / or freeform surfaces), (6) build the B-Rep of the reconstructed CAD model, and (7) conduct the engineering analysis. RE applications include (but are not limited to) Finite Element Analysis (FEA) [2,3], Structural Optimization [4,5] and Dimensional Analysis [6,7].

Mesh segmentation / parameterization plays a crucial role in RE (steps 4–5) for the adequate geometric modeling of the workpiece. The mesh segmentation / parameterization problem is defined as follows:

Given: A 2-manifold triangular mesh M (or simply, "mesh") embedded in $\mathbb{R}^3$. **Goal:** (i) to partition (i.e., segment) the triangle set M into a set of disjoint and connected sub-meshes $\{M_1, M_2, \ldots, M_k\}$ which together compose the original mesh, and (ii) to compute a bijective parameterization $\psi_i : M_i \rightarrow \mathbb{R}^2$ for each sub-mesh M_i. The segmentation step (i) must favor the parameterizability of the computed sub-meshes while retaining feature (functional) surfaces of the scanned workpiece.

Mesh segmentation algorithms can be classified depending on the surface features used to divide the mesh:

1. **Geometry-based segmentation** captures locally geometric features of the surface (sharp edges, principal curvatures, surface normals, etc.) and partitions the surface using this information. This type of segmentation is ideal for CAD meshes that present clear sharp transitions between sub-meshes. However, geometric criterion alone applied to noisy or imperfect meshes results in over-segmentation (Fig. 2). If the workpiece

☆ This article was recommended for publication by Jun-Hai Yong.

* Corresponding author.
E-mail address: jrsanchez@vicomtech.org (J.R. Sánchez).

https://doi.org/10.1016/j.cag.2018.03.004
0097-8493/

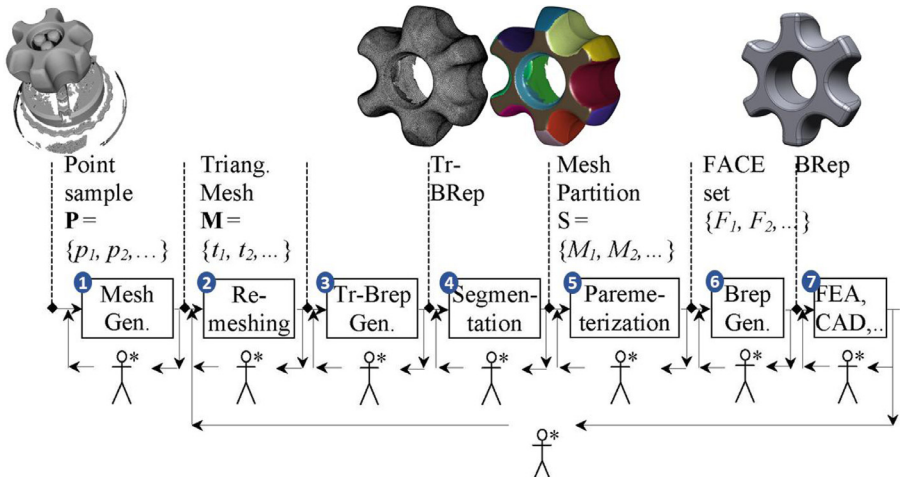
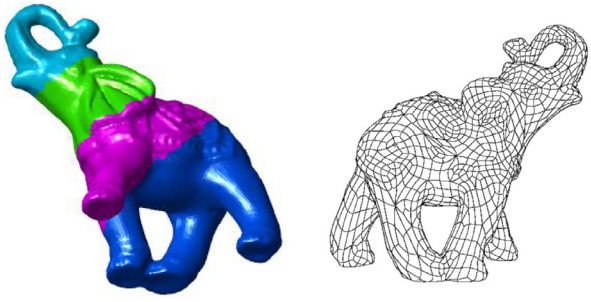


Fig. 1. The current RE workflow is user-intensive [8].

Fig. 2. Dihedral segmentation produces over-segmentation due to surface imperfections and surface blends.



(a) Topologic segmentation [8] (b) Interactive segmentation with Geomagic® for RE [8]

Fig. 3. Topology vs. user - based segmentations [8].

is smooth, geometric segmentation produces large and (likely) non-parameterizable sub-meshes.

2. **Topology-based segmentation** relies on the spectra (eigenpairs) of any Laplacian operator computed on the mesh graph. This type of segmentation is common in Computer Graphics applications. However, this segmentation usually results in non-parameterizable sub-meshes (Fig. 3(a)).
3. **Interactive segmentation** is the most common practice by RE software (such as Geomagic® and Polyworks®, Fig. 3(b)). The current state-of-the-art segmentation approaches still demand expensive user interaction in order to achieve suitable segmentations for parameterization and B-Rep reconstruction (Fig. 1).

Having the mesh segmented, the construction of the B-Rep becomes straightforward if a bijective parameterization of each sub-mesh is computed. A trimmed NURBS (Non-Uniform Rational B-Splines) surface can be fitted by Least Squares [9] or Radial Basis Functions (RBFs) [10] to each sub-mesh. The fitted surfaces and their trimming curves can be oriented and related to each other by their adjacency graph in order to produce the reconstructed B-Rep model [11].

This manuscript presents a hybrid mesh segmentation / parameterization algorithm for RE, as follows: (i) A set of heat transfer equations are defined on the mesh. The topology of the mesh is captured by the Laplace–Beltrami operator inherent in the differential equation for heat transfer. (ii) Temperature constraints are imposed on a subset of vertices (mesh seeds), acting as heat sources and sinks. The local geometry of the mesh is captured by choosing the mesh seeds according to a dihedral angle criterion. (iii) To avoid over-segmentation, seeds that produce small sub-meshes are ignored. The temperature fields are used to re-compute the segmentation without these small sub-meshes. (iv) The parameterization of each sub-mesh is thereafter computed by a Hessian-based parameterization [12].

The contribution of this manuscript resides in the mixed topology (temperature) / geometry (dihedral) nature of the segmentation algorithm. Our algorithm not only pursues mesh parameterizability but also a functional partition of scanned mechanical workpieces, without resorting to over-segmentation. The algorithm allows (almost) automatic processing of 3D meshes from scanned workpieces, improving the RE workflow.

The remainder of this manuscript is organized as follows: Section 2 reviews the relevant literature. Section 3 describes the mesh segmentation algorithm. Section 4 discusses the implementation details of the algorithm. Section 5 presents and discusses results of the conducted experiments. Section 6 concludes the paper and introduces what remains for future work.

2. Literature review

Mesh segmentation algorithms can be classified depending on the mesh properties used to partition the mesh as follows:

2.1. Geometry-based segmentation

Geometry-based segmentation approaches compute local geometric properties (e.g., dihedral angle, curvature, frequency, [8,13]) and use region-growing algorithms to lump property - homogeneous regions (Fig. 2).

Shape recognition algorithms partition the surface by matching analytic shapes to the mesh [14–16]. One of these analytic shapes (plane, sphere, cylinder or cone) is registered to each mesh vertex according to the local geometric information (such as curvature).

A clustering or region growing algorithm is finally applied to compute the mesh segmentation.

Geometry-based segmentation algorithms (1) require several post-processing due to over-segmentation, (2) do not favor functional or feature segmentation, and (3) are highly sensitive to noise. Mesh smoothing may be used to reduce noise previous to segmentation [13].

2.2. Topology-based segmentation

In spectral analysis, a mesh topology operator matrix (e.g., adjacency or Laplacian) is estimated on the mesh graph in order to extract and analyze its spectra (eigenpairs) [17]. A partition of the first non-constant Laplacian (Fiedler) eigenvector reflects a possible segmentation of the mesh [8,18]. A central pre-condition for spectral methods is the edge length homogeneity through the mesh. To improve the robustness of the spectral segmentation, Refs. [19,20] segment similar meshes simultaneously by introducing edge correspondences between meshes, while Ref. [21] captures images of the same mesh from different perspectives in order to correlate the mesh edges.

Ref. [22] computes an edge weighted Laplacian which includes information about concave regions. Chosen Laplacian eigenvectors are merged into a single scalar field whose partition segments the mesh. Ref. [23] introduces Secondary Laplacian and Giaquinta–Hildebrant operators which locally capture geometric properties (e.g., principal curvatures), thus allowing to infer 3D concavities / convexities. Ref. [24] computes the spectra of a weighted dual graph Laplacian. The dual Laplacian encodes the topology of the mesh in terms of the connectivity of the triangles (instead of the points connectivity). The weighting scheme incorporates dihedral angles, which improves the sub-mesh definition.

Heat-based algorithms are an alternative approach for topologic segmentation, defining and solving different heat transfer equations on the mesh. The topology of the mesh is captured by the Laplace–Beltrami operator, present in the heat equation. The resulting segmentation is obtained from the computed temperature fields on the mesh. Ref. [25] presents an interactive segmentation algorithm where the user draws lines perpendicular to potential sub-mesh boundaries. The algorithm defines temperature constraints according to these user strokes. The algorithm computes the constrained temperature fields and produces the segmentation based on the temperature contours.

Heat kernels are specific solutions to the heat transfer problems with unique point sources. These heat kernels can be computed by means of the eigenvectors of the Laplace–Beltrami operator [26,27]. Refs. [28,29] compute the heat potential (tendency to attract heat) of each mesh point in order to identify crucial heat sources which are then used to compute the heat kernels and the underlying segmentation.

In general, topology-based methods present several shortcomings: (1) they produce large sub-meshes which are non-parameterizable, and (2) they usually require heavy user interaction in selection of eigenpairs (spectral) or heat sources (heat-based) on the mesh, critical for the quality of the segmentation [30].

2.3. Mesh segmentation in RE

RE workflow currently requires intensive, costly user input (Fig. 1). Commercial tools include PolyWorks® [4], RapidWorks® [6] and Geomagic® [8]. Refs. [2,31] apply RE to run FEA on scanned turbine blades. The turbine blades are manually divided into sections prior to digitizing. Ref. [11] uses the dihedral angle and curvature scalar fields on the mesh to segment it, seeking to optimally fit analytic shapes (sphere, cylinder, cone, etc.). Refs. [9,32] fit freeforms to growing sub-meshes, with Ref. [32] favoring rectangular ones. A common approach to represent an unknown model is to fit rectangular NURBs patches to the whole mesh [32]. These small NURBs patches have the advantage to produce low-distortion parameterizations, even in the case of complex geometries where such parameterization can be optimized to produce the smallest distortion [33,34]. However, such patches usually lack from the functional information of the source CAD model (see Fig. 3(b)).

2.4. Conclusions of the literature review

Current state-of-the-art segmentation algorithms are not fully suitable for RE applications. Geometry-based segmentation algorithms produce over-segmentation on scanned workpieces due to surface imperfections and surface blends between sub-meshes. On the other hand, topology-based algorithms result in parameterization - hostile segmentations. Therefore, the current RE workflow demands massive user input in order to produce usable B-Reps, requiring between 25–150 h of interactive work for a single scanned workpiece [4,8].

To overcome these problems, this article presents an automatic mesh segmentation algorithm for RE: (1) Our algorithm defines several constrained heat transfer problems on the mesh for segmentation. Temperature constraints are located automatically using a dihedral criterion. To avoid over-segmentation, constraints that produce small sub-meshes are removed. Therefore, our algorithm favors sub-meshes parameterizability by capturing local geometric features (dihedral angle) and avoids over-segmentation by capturing topological mesh features (temperature fields). (2) The sub-meshes are parameterized with a Hessian-based parameterization algorithm [12]. Results are presented for meshes collected from a 3D optical scanner and public benchmarks.

3. Methodology

To compute the segmentation of the mesh M, we extend the heat-based approach presented in Ref. [25], making the segmentation procedure completely automatic (in the sense that it does not require user interaction) as follows (Fig. 4): (1) instead of manually selecting a heat source and a heat sink to split the mesh into two sub-meshes, our segmentation algorithm locates simultaneously the set of all heat sources and sinks S. Such heat sources and sinks (mesh seeds) are located automatically by a dihedral angle criterion which captures the local geometry of the mesh. (2) These heat sources / sinks are used to define a set of heat transfer differential equations on the whole mesh. Therefore, for each heat source $S_i \subset S$, a mesh temperature field u_i is found. (3) The computed temperature fields are compared for each vertex in order to define a unique pre-segmentation of M. (4) Seeds that produce small sub-meshes are removed to avoid over-segmentation, resulting in a new set of temperature fields. (5) Finally, these new temperature fields define the final segmentation of M. In addition, an almost automatic parameterization algorithm proceeds as follows: (6) Artificial boundaries are manually (interactively) introduced only in the case of cylinder-like sub-meshes to allow their parameterization. (7) The parameterization of each sub-mesh is computed with a Hessian-based parameterization algorithm [12].

3.1. Automatic placement of mesh seeds

One of the crucial requisites in mesh parameterization resides in the parameterizability of the resulting segmentation. Such parameterizability is hindered by high frequency zones and favored by low frequency zones. Our algorithm locates a set of mesh seeds S in the low frequency neighborhoods of the mesh. These mesh seeds will expand the different sub-meshes of the segmentation

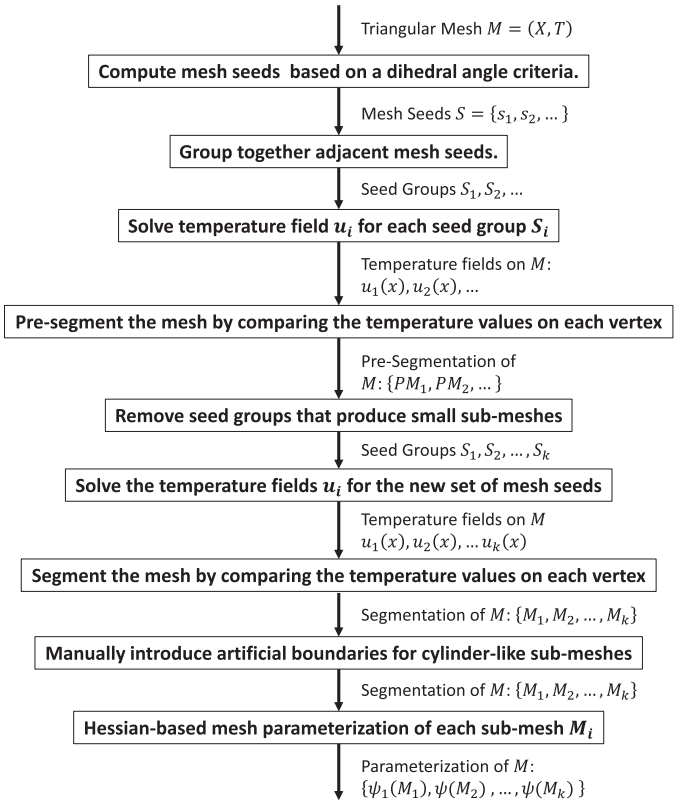


Fig. 4. Overall scheme of the segmentation algorithm.

by propagating heat through the whole mesh (discussed in subsequent sections). We identify such low frequency zones by a dihedral angle criterion as follows:

1. Set a dihedral angle threshold $\theta_{threshold} \to 0$
2. For each vertex $x_i \in M$:
   (a) Compute the incident edges $E_i = e_1, e_2, \dots$ on x_i.
   (b) Compute the dihedral angle θ_j of each incident edge $e_j \in E_i$.
   (c) If $\pi - \theta_j > \theta_{threshold}$ (for any incident edge e_j), then skip the current vertex.
   (d) Else, insert the current vertex x_i in the list of the mesh seeds S.

A vertex is considered as a low frequency vertex if and only if none of its incident edges is sharp (Fig. 5). An edge is sharp (non-planar) if $\pi - \theta_j$ is larger than the dihedral threshold $\theta_{threshold}$. Therefore, $\theta_{threshold} \to 0$ can be seen as the maximum non-coplanarity between two adjacent triangles in a low frequency zone.

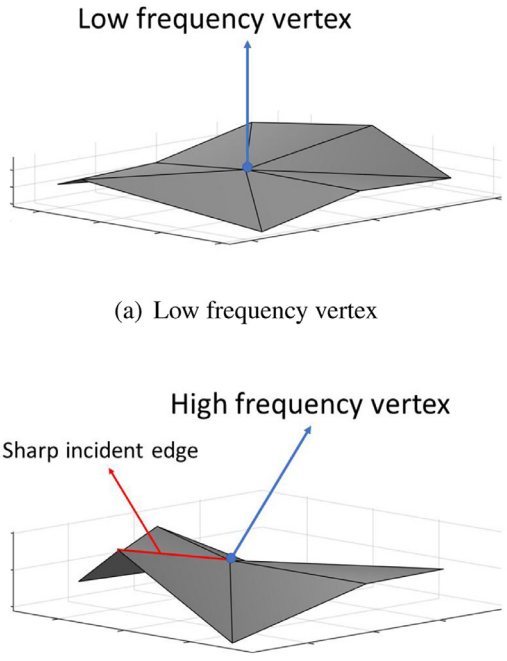
Our algorithm ensures that adjacent low frequency mesh vertices lie in the interior of a common sub-mesh by grouping them into a subset of mesh seeds $S_i \subset S$ (Fig. 6(a)).

3.2. Heat transfer with temperature constraints

The following partial differential equation describes the steady heat transfer phenomenon without heat sources on the mesh M:

$$\Delta u(x) = 0, \tag{1}$$

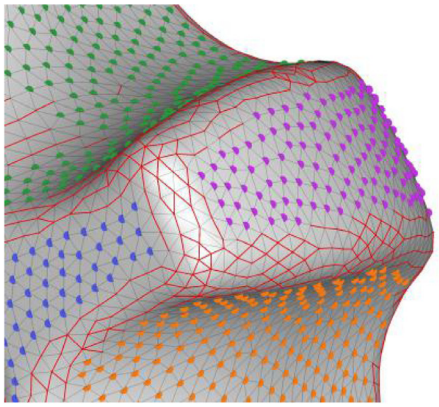
where Δ is the Laplace–Beltrami operator and $u(x)$ is the temperature distribution along the surface. We impose a temperature value $(u(S_i) = 1)$ on a subset of mesh seeds (heat sources) $S_i \subset S$, and temperature value $(u(S_j) = 0)$ at the remaining seed sets (heat sinks) $S_j \subset M$, $i \neq j$. Each subset of sources S_i will define a sub-mesh M_i of the segmentation. Therefore, for each sub-mesh M_i, the



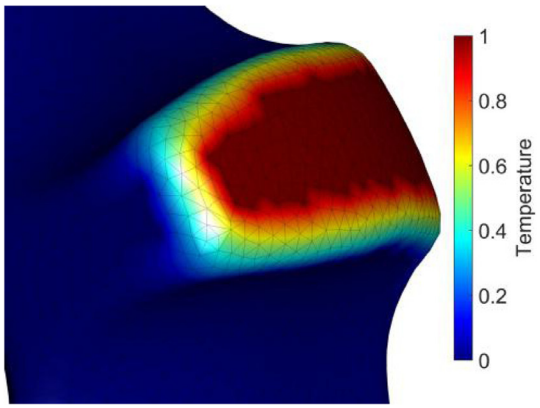
(a) Low frequency vertex

(b) High frequency vertex with a sharp incident edge

Fig. 5. Examples of low and high frequency vertex for selection of mesh seeds based on a dihedral criterion.



(a) Mesh seeds on low frequency zones grouped by color



(b) Temperature solution for one seed group

Fig. 6. Mesh seeds are located at low frequency zones. Each seed group defines a temperature field on the mesh.

following constrained heat problem arises:

$$\begin{aligned} &\Delta u_i(x) = 0 \\ &s.t. \\ &u_i(S_i) = 1, \\ &u_i(S_j) = 0, \qquad i \neq j \end{aligned} \tag{2}$$

For each heat source S_i, its corresponding temperature field $u_i(x)$ is obtained by propagating the thermal energy through the whole mesh M (Fig. 6(b)). The temperature solution $u_i(x)$ is directly related to the sub-mesh M_i, achieving maximum value ($u = 1$) at the defined heat sources S_i and minimum value ($u = 0$) at the remaining heat sinks S_j, $i \neq j$.

The FEA Method is implemented to estimate Δ numerically. Therefore, Δ is approximated by the FEA matrix $\mathbf{L}$, defined as [35,36]:

$$\mathbf{L}_{ij} = \begin{cases} \frac{3}{A_i} w_{ij}, & \text{if } (x_i, x_j) \text{ is an edge of } M \\ -\frac{3}{A_i} \sum_{x_k \in N_i} w_{ik}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}, \tag{3}$$

where N_i is the neighborhood of x_i, $w_{ij} = \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{2}$ is the cotangent weight of edge (x_i, x_j), α_{ij} and β_{ij} are the angles opposite to edge (x_i, x_j), and A_i is the area of all the triangles incident to vertex x_i. An $n \times n$ linear system of equations $\mathbf{A}U_i = B_i$ arises for each heat source S_i, with:

$$\mathbf{A} = \begin{bmatrix} \mathbf{L} \\ \mathbf{I_{S_i}} \\ \mathbf{I_{S_j}} \end{bmatrix}, \text{ and } B_i = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix}, \tag{4}$$

where $\mathbf{L}$ is the FEA matrix for all the nodes with unknown temperature in M (i.e., rows associated to mesh sources and sinks are excluded from $\mathbf{L}$), $\mathbf{I_{S_i}}, \mathbf{I_{S_j}}$ are the constraints matrices for the current heat sources S_i and heat sinks S_j, respectively (i.e., entry kl of matrix $\mathbf{I_S}$ is 1 if constraint k fixes the temperature for the heat source / sink x_l, 0 otherwise).

Our algorithm simultaneously solves several heat transfer problems (one for each group of seeds S_i). The matrix $\mathbf{A}$ is common to all of them and it is computed and prefactored once. The linear system defined by Eq. (4) is then solved using sparse Cholesky factorization, which in most cases can be solved in nearly linear time $\mathcal{O}(n)$ [37,38]. The following section describes how to combine the different temperature fields to obtain a single segmentation field.

3.3. Heat-based mesh segmentation

At this point, each vertex x_i in the mesh has an associated set of temperature values $u_1(x_i), u_2(x_i), \ldots$ from the temperature fields generated by each set of mesh seeds $S_1, S_2, \ldots$. The segmentation of M is achieved by computing the maximum temperature value at each vertex and its corresponding seed group. Thus, each sub-mesh M_i is composed by the subset of vertices whose maximum temperature is $u_i(x)$:

$$M_i = \left\{ x_k \in M \mid u_i(x_k) > u_j(x_k), i \neq j \right\} \tag{5}$$

This construction guarantees that the set of heat sources S_i belongs to the sub-mesh M_i, assigning low frequency areas to the same sub-mesh. Heat propagates smoothly from these zones to higher frequency zones, defining the sub-mesh boundaries.

3.4. Discarding small seed groups

In RE, the mesh M presents surface imperfections due to manufacturing imperfections and / or RE pre-processing results (such as data acquisition, surface meshing, mesh filtering, mesh decimation, etc). Such imperfections and mesh noise produce small groups of seeds that lead the heat algorithm to an over-segmentation of the surface. A second heat - based segmentation is then executed excluding noise - originated seeds. An overview of the method follows:

1. Locate the initial heat seeds on low frequency neighborhoods.
2. Find the mesh temperature fields and segment accordingly.
3. Compute the area of each sub-mesh.
4. Given the sub-mesh with the largest area $A_{largest}$, locate all the sub-meshes with an area below $\varepsilon \cdot A_{largest}$ (small sub-meshes).
5. Discard seeds in small sub-meshes.
6. Re-compute the temperature fields with the surviving seeds.
7. Re-compute the segmentation with the new temperature fields.

The area percentage parameter $0 \leq \varepsilon \leq 1$ measures the minimum sub-mesh size (relative to the largest sub-mesh) allowed by the segmentation. Triangles belonging to small sub-meshes in the over-segmentation are appended to the largest sub-meshes by temperature propagation as discussed in Section 3.3.

(a) Parameterization - hostile cylinder-like sub-mesh (b) Parameterizable cylinder-like sub-mesh

Fig. 7. To parameterize cylinder-like sub-meshes, an artificial boundary (red) is manually introduced using a cylinder generatrix. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).

3.5. Segmentation of cylinder-like sub-meshes

In the well known case of cylinder-like sub-meshes, our segmentation algorithm produces parameterization - hostile surfaces. However, such surfaces can be made parameterizable by manually making a generatrix of the cylinder a mesh boundary (Fig. 7), as follows:

1. The user selects the start and end vertices of the generatrix.
2. A shortest path (Dijkstra) algorithm computes the path that links the start and end vertices.
3. Our algorithm generates a new B-Rep of the sub-mesh introducing the computed trajectory as sub-mesh boundary.

Other authors have addressed the problem of fitting closed cylinders using least squares minimization [9,11]. However, our approach comprises not only standard cylinders but also their topological equivalents (with and without holes).

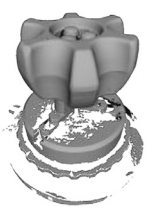
3.6. Hessian-based mesh parameterization

To compute the parameterization $\psi \subset \mathbb{R}^2$ of M, a Hessian-based mesh parameterization algorithm [12] is applied on each sub-mesh M_i. This Hessian mesh parameterization algorithm applies the main concepts of Hessian Locally Linear Embedding (HLLE) [39] (a Dimensional Reduction algorithm) on triangular meshes.

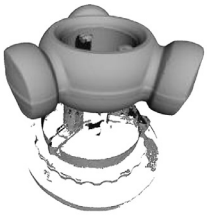
According to [12,39], a parameterization of M_i is given by the first 2 non-constant eigenvectors of the Hessian functional $\mathcal{H}$, defined as:

$$\mathcal{H}f = \int_{M_i} \|\mathbf{H}_{\mathbf{x}}^{\tan} f\|_F^2 \mathrm{dA} \approx \mathbf{f}^{\mathbf{T}} \mathbf{K} \mathbf{f}, \tag{6}$$

Fig. 8. Acquisition of 3D point cloud data through an optical 3D scanner.



(a) Knob (17.3m points)



(b) Tripod joint (23.6m points)

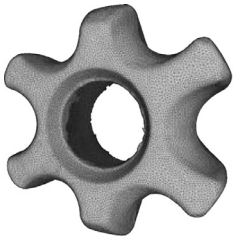
(c) Rocker arm base (3.1m points)

Fig. 9. Datasets (point clouds) obtained with an optical scanner.

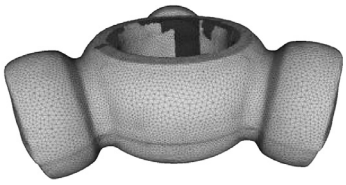
where $f \in \mathcal{C}^2(M_i)$ is a smooth function defined on M_i, $\|\cdot\|_F$ is the Frobenius norm, dA is the surface differential, $\mathbf{f} = \{f_1, f_2, \ldots, f_n\}$ are the values of $f(x_j)$ at each vertex $x_j \in M_i$, and $\mathbf{K} = (\mathbf{K}_1 + \mathbf{K}_2 \ldots, \mathbf{K}_n)$ is the discrete Hessian estimator (matrix). This matrix is semidefinite-positive [12,39]. Therefore, the parameterization $\psi_i(M_i)$ is extracted by computing the first two eigenvectors of $\mathbf{K}$ with the smallest non-zero eigenvalue. Such eigenvectors correspond to an orthogonal basis for all linear functions (and as a consequence, a basis for all parameterizations) defined on M_i.

4. Implementation of the algorithm

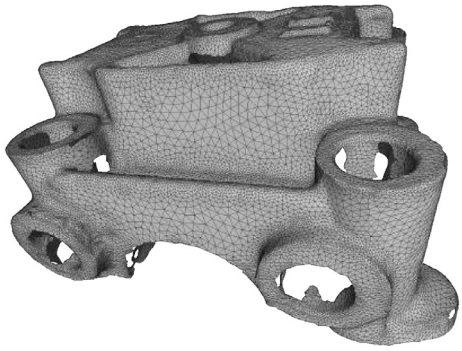
To test our algorithm in a real RE context, different engineering pieces have been scanned with an optical 3D scanner (Fig. 8). The RE result for these pieces is used in real engineering contexts. The optical 3D scanner produces point cloud data for each workpiece. Fig. 9 plots the datasets obtained by scanning (a) a knob, (b) a tripod joint, and (c) a rocker arm base. These datasets were user - processed in Geomagic® Design™ to ensure manifold properties (pre-condition for segmentation and parameterization).



(a) Knob (57.8 faces)

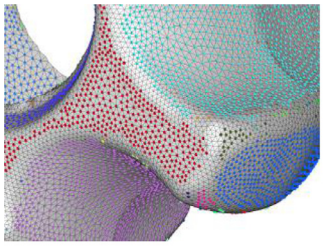


(b) Tripod joint (61.0k faces)

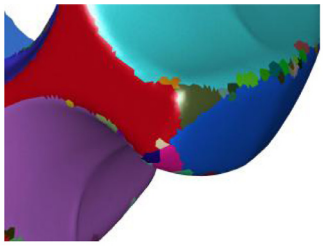


(c) Rocker arm base (36.1k faces)

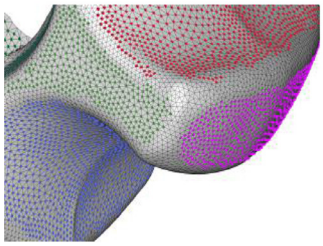
Fig. 10. Input meshes for our segmentation algorithm. These meshes are the result of manual preprocessing with commercial software (Geomagic®).



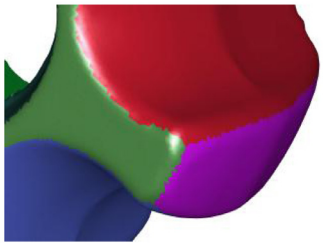
(a) (Raw) Knob seeds



(b) Knob temperature-based segmentation (raw seeds)



(c) (Processed) Knob seeds



(d) Knob temperature-based segmentation (processed seeds)

Fig. 11. Discarding mesh seeds. The initial seed groups produce over-segmentation due to surface imperfections (a–b). After discarding small seed groups, such over-segmentation is removed (c–d).

Fig. 10 plots the resulting meshes after the interactive processing. Large holes have been left in the mesh. These meshes are the actual input for our segmentation algorithm.

Fig. 11 plots the seed groups processing for the Knob mesh. The initial mesh seeds are computed with a $\theta_{threshold} = \frac{1}{20}\pi$ radians (see Table 1), since this value has shown to consistently capture flat zones in all of our experiments. Several small seed groups arise due to isolated low frequency points inside high fre-

Table 1
Default parameter values for our segmentation algorithm.

Parameter	Value
$\theta_{threshold}$	$\frac{1}{20}\pi$ radians
ε	5%

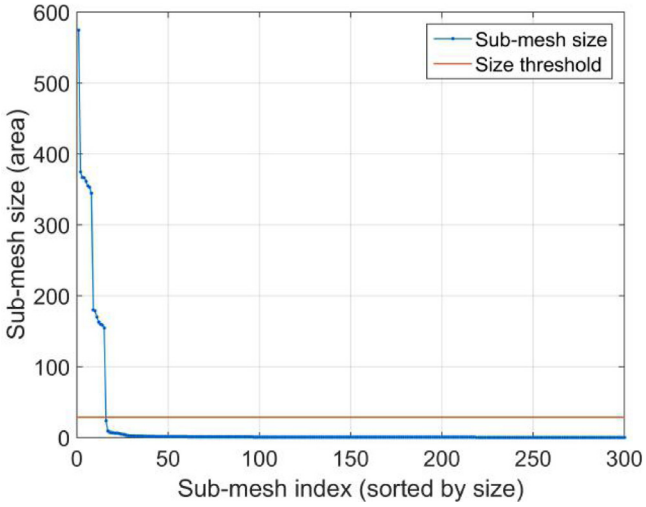
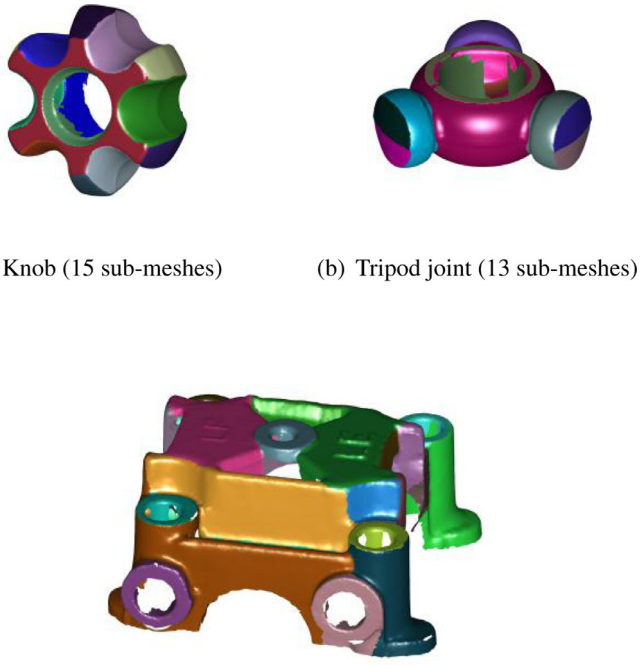


Fig. 12. Sub-mesh sizes for the initial segmentation of the knob mesh. The red line plots the area threshold ($\varepsilon = 5\%$) used to discard small sub-meshes from the final segmentation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.).



(a) Knob (15 sub-meshes) (b) Tripod joint (13 sub-meshes)

(c) Rocker arm base (27 sub-meshes)

Fig. 13. Temperature-based segmentation. The dihedral criterion captures the local mesh geometry while the temperature approach produces smooth transitions between sub-meshes.

quency zones (Fig. 11(a)). The temperature-based segmentation using these seeds results in an over-segmentation of the surface (Fig. 11(b)). Small (noise-generated) sub-meshes are then discarded by the algorithm (as discussed in Section 3.4), no longer receiving heat seeds and therefore being absorbed in natural form by large meshes when the heat algorithm is run again. After applying seeds processing, the remaining seeds capture the local flat geometry of the mesh and the high frequency seeds disappear (Fig. 11(c)). The final temperature-based segmentation preserves the geometric properties from the dihedral criterion in low frequency zones while producing a smooth transition between sub-meshes, avoiding mesh over-segmentation (Fig. 11(d)). Fig. 12 plots the distribution of the sub-mesh sizes (sorted by surface area) and the area threshold used to discard small sub-meshes. In all our conducted experiments we choose an area threshold parameter of $\varepsilon = 5\%$ (see Table 1) as we have identified that it consistently differentiates large (albeit parameterizable) sub-meshes (Fig. 11(d)) from small (noise-generated) ones (Fig. 11(b)). The initial segmentation of the knob produces 300 sub-meshes while the final segmentation produces only 15 sub-meshes. Decreasing the value of ε in Fig. 12 would increase the likelihood of over-segmentation. On the other hand, increasing its value could lead the algorithm to merge large sub-meshes and produce non-parameterizable segmentations. The user may, of course, change the cutting value (upon examination of the distribution exemplified in Fig. 12), reinforcing or decreasing the absorption of small sub-meshes into the larger ones.

Fig. 13 plots the final segmentation results for each mesh. The computed sub-meshes present low frequencies while sub-mesh boundaries are located in high frequency zones. The segmentation is controlled by the area percentage parameter ε (taken as $\varepsilon = 5\%$ in all our experiments), discarding noise - related sub-meshes as discussed in Section 3.4. Our algorithm produces parameterization - friendly segmentations while keeping a relatively low number of

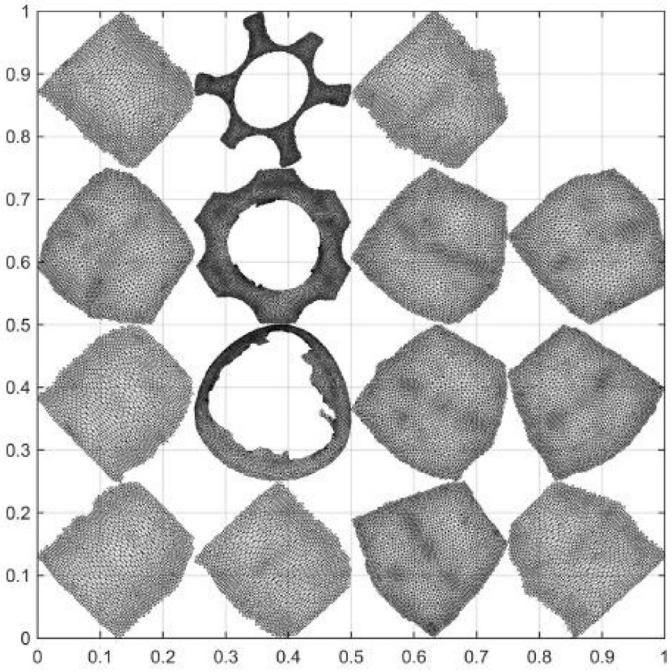


Fig. 14. Knob Hessian parameterization.

sub-meshes (15 sub-meshes for the knob - Fig. 13(a), 13 for the tripod joint - Fig. 13(b), and 27 for the rocker arm base - Fig. 13(c), respectively.).

Hessian parameterization is then applied on each sub-mesh. Fig. 14 plots the 2D parameterization of each of the knob sub-meshes. Such parameterization is completely bijective (i.e., no triangle flips nor surface overlaps occur in the parametric space).

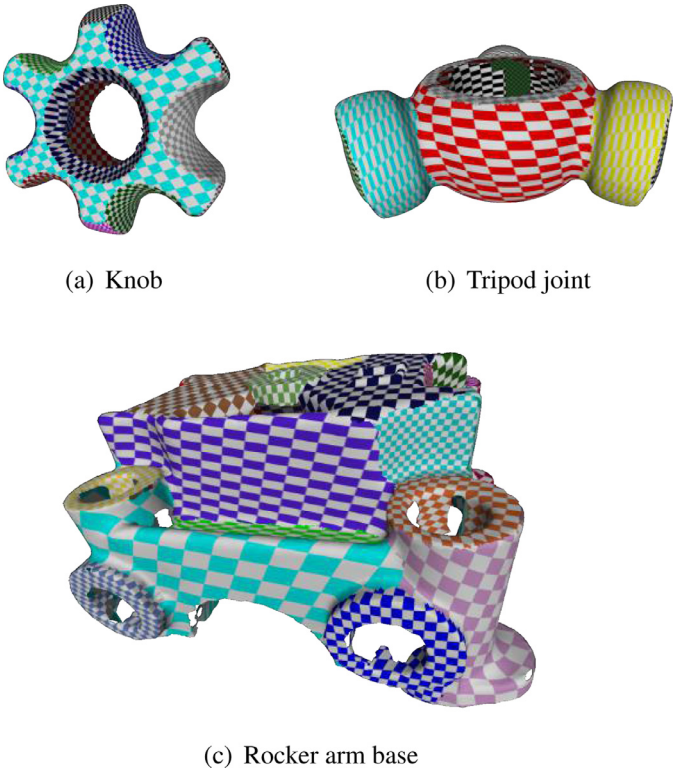


Fig. 15. Chessboard texture applied on each sub-mesh. The resulting parameterization is bijective for all the scanned models.

Fig. 15 plots the chessboard textures applied on the resulting segmentation using the computed Hessian parameterization. The distortion of the chessboard squares represents the distortion of the computed parameterization. In the case of the tripod joint (Fig. 15(b)), artificial boundaries have been introduced manually on the cylinder-like sub-meshes as discussed in Section 3.5. Fig. 15 displays (using chessboard textures) the computed bijective Hessian parameterizations of the sub-meshes. The special segmentation case of topological cylinders (e.g., tripod joint, Fig. 15(b)) currently requires the manual creation of a boundary along a cylinder generatrix (discussion in Sect. 3.5).

5. Results and benchmarking

This section presents a comparison of our segmentation algorithm against several state-of-the-art algorithms and commercial CAD software. Section 5.1 presents a standard benchmarking using datasets and algorithms from the National Design Repository [40] and the Princeton Benchmark Repository [41], which are standard in the mesh segmentation literature. Afterwards, Section 5.2 compares our algorithm against recent algorithms from the literature and some commercial software using our in-house scanned pieces (introduced in Section 4).

5.1. Standard benchmarking

Fig. 16 plots our segmentation results compared with the Cross Boundary Brushes algorithm results [25] for some CAD models from the National Design Repository [40]. Both methods use a heat-based approach to capture geometric features of each CAD model. However, Cross Boundary Brushes is completely interactive, requiring user input for each computed sub-mesh. CAD models usually present several geometric features which require moderate segmentation sizes (>10 sub-meshes). Therefore, interactive user input may become unreliable in such cases. In contrast, our algorithm produces similar segmentation results and parameterizable sub-meshes without requiring any user input.

Table 2
Number of sub-meshes for the segmentation results of the Princeton Benchmark.

Algorithm \Dataset	Flange	Cup
Random cuts [42]	7	4
Random walks [43]	5	3
Fitting primitives [44]	8	6
Our (Temp-Geom) algorithm	21	5

Table 3
Number of sub-meshes for the segmentation results of our scanned models using state-of-the-art algorithms and commercial software.

Algorithm \Dataset	Knob	Tripod joint	Rocker arm base
Contour Based Segmentation [13]	6	9	11
Autodesk® 3ds Max®	6216	5633	11,772
Geomagic® Design™	40	26	115
Our (Temp-Geom) algorithm	15	13	27

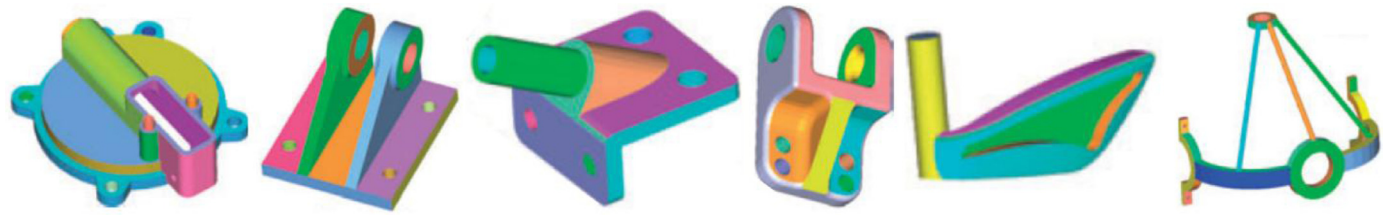
Fig. 17 plots segmentation results of our algorithm and some automatic algorithms from the Princeton Benchmark [41]. Our algorithm is able to capture the geometric features of the surface for the flange dataset (Fig. 17(d)) while other algorithms struggle to capture such features, grouping different surfaces (such as the cylinders, cones and the plane on the flange orifices) into the same sub-mesh (Figs. 17(a)–17(c)). As a consequence, our algorithm produces more sub-meshes (21) than the benchmark algorithms (<10), which in a RE context is preferable to allow easy parameterization of each of the flange sub-meshes (see Table 2). On the other hand, our segmentation of the cup dataset (Fig. 17(d)) results in a similar number of sub-meshes (see Table 2), and it is in agreement with the rest of the benchmarking algorithms (Figs. 17a–17(c)), correctly segmenting the cup model into its meaningful parts (Fig. 17(d)).

Our segmentation algorithm is designed to work on scanned meshes of mechanical pieces. As a consequence, our algorithm behaves unexpectedly if applied to organic meshes. Fig. 18 illustrates this fact by applying our algorithm to a human mesh. The result is a bad segmentation with features not being characterized by our algorithm (such as head, hands or leg), and also each sub-mesh is non-parameterizable. Despite of the topology (heat-based) component of the algorithm, such a result is mainly due to the dihedral-criterion used to place the temperature seeds on the mesh (see Section 3.1). This problem can be addressed by changing the approach to define these seeds, which is left for future work.

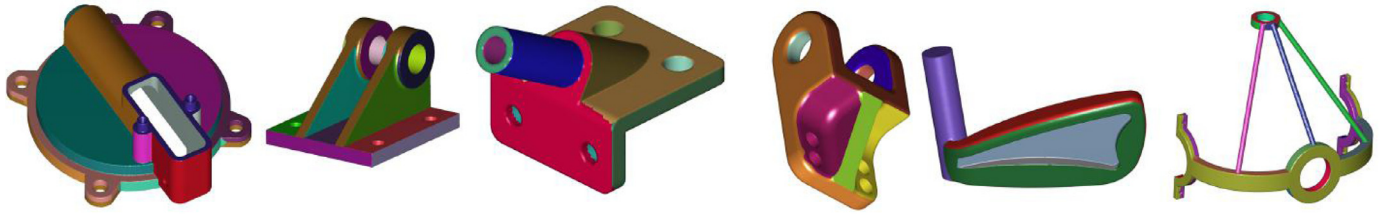
5.2. RE benchmarking

Fig. 19 plots the segmentation results of the scanned mechanical workpieces (introduced in Section 4) using state-of-the-art segmentation techniques. Fig. 19a plots the segmentation result using our implementation of the Contour Based automatic algorithm [13]. The resulting segmentation captures some of the surface features of the tripod joint and rocker arm meshes. However, sub-mesh boundaries are non-smooth and do not capture the real boundaries of the workpiece surfaces. The number of sub-meshes is relatively low (see Table 3) for each segmented piece, grouping several feature surfaces of the workpiece in the same sub-mesh, which difficults the parameterization step of the RE process. The segmentation of the knob mesh is undesirable in the context of RE.

Figs. 19(b)–19(c) plot the automatic segmentation results of the scanned workpieces using commercial CAD software. The Autodesk® 3ds Max® result is able to locate the different fea-

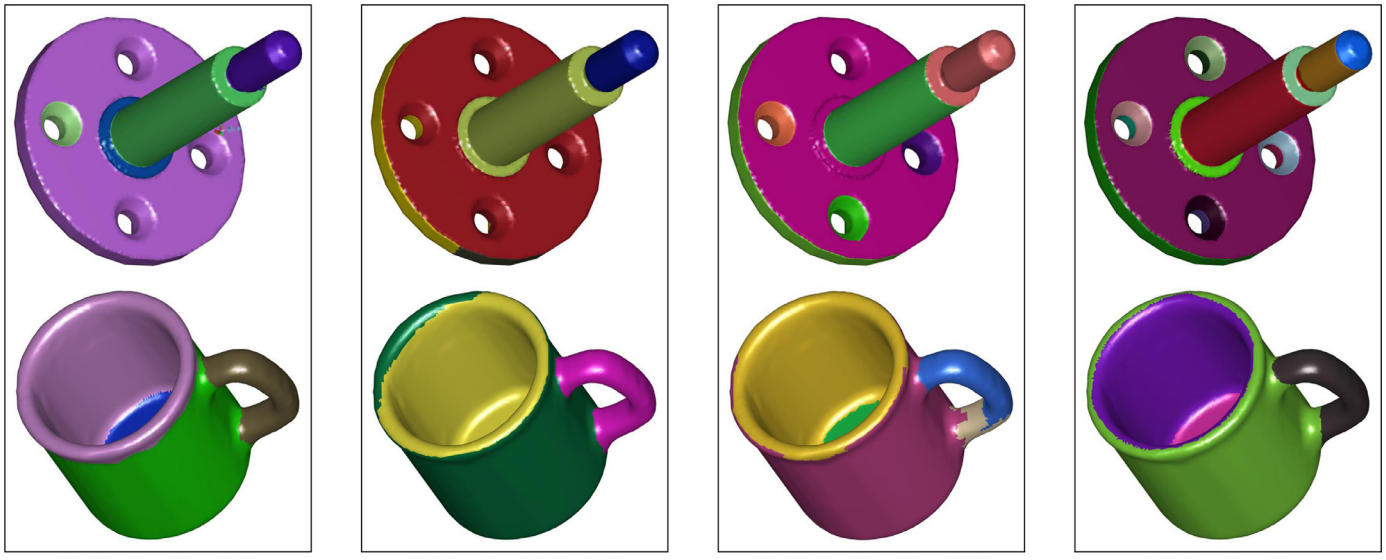


(a) Cross Boundary Brushes [25] (fully interactive)



(b) Our method (automated)

Fig. 16. Standard benchmarking. Segmentation results of the Cross Boundary Brushes method [25] (above) vs. our automated method (below). Meshes from the National Design Repository [40].



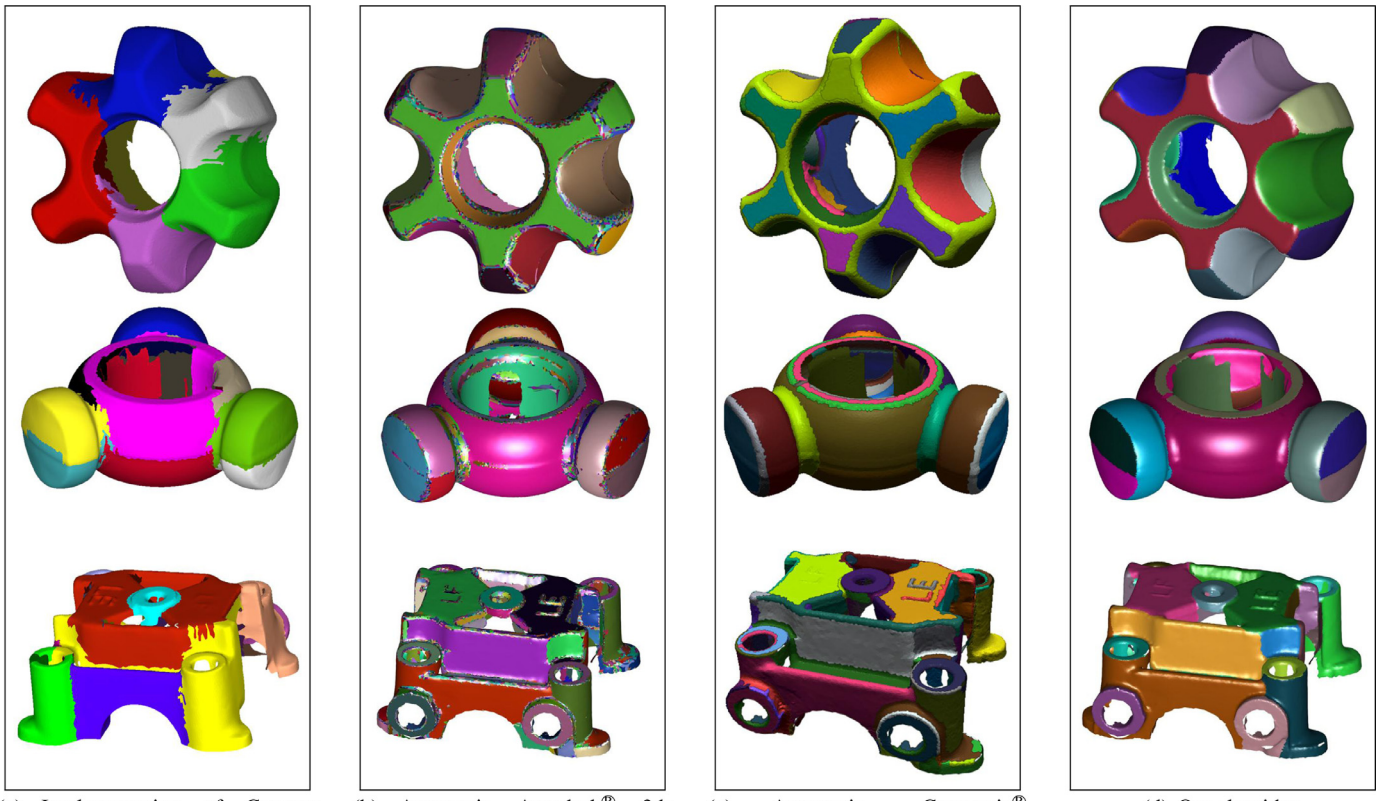
(a) Random Cuts [42] (b) Random Walks [43] (c) Fitting Primitives [44] (d) Our algorithm

Fig. 17. Standard benchmarking. Segmentation results for the flange and cup models. Meshes from the Princeton Benchmark [41].

Fig. 18. Non-parameterizable segmentation of an organic mesh with our algorithm. Human mesh from the Princeton benchmark [41].

ture surfaces of the CAD meshes. However, it produces an excessive amount of sub-meshes (>1000, see Table 3) which are for the most part product of mesh noise and blending surfaces. On the other hand, the Geomagic® Design™ result captures not only feature surfaces but also blending surfaces (which dictate smooth transitions between feature surfaces) while ignoring the mesh noise. Such a result is highly desirable in a RE context to reconstruct the analytic surfaces of the scanned model. However, these blending surfaces can produce over-segmentation at some degree as illustrated in the rocker arm of Fig. 19(c), which has 115 sub-meshes. Our algorithm solves this problem by merging the blending surfaces into the feature surfaces (Fig. 19(d)), reducing this number to 27 [10] while keeping the segmentation parameterizable.

Table 4 presents the main advantages and disadvantages of all the segmentation algorithms used in this manuscript. Our algorithm provides an automatic alternative to mesh segmentation of mechanical pieces for RE, avoiding over-segmentation even in the presence of blending surfaces and mesh noise (natural to scanning devices and manufacture defects). It is worth to note that in the general context of mesh segmentation, an algorithm is considered



(a) Implementation of Contour Based Segmentation [13] (b) Automatic Autodesk® 3ds Max® segmentation (c) Automatic Geomagic® Design™ segmentation (d) Our algorithm

Fig. 19. Comparison of results: (a) state-of-the-art competitor [13], (b–c) commercial tools, (d) our algorithm. Datasets: in-house scanned mechanical pieces.

Table 4
Advantages and disadvantages of each segmentation algorithm.

Algorithm	Advantages	Disadvantages
Cross Boundary Brushes [25]	1. Works on both mechanical and organic meshes 2. Smooth sub-mesh boundaries	1. Non-automatic (requires heavy user interaction)
Random Cuts [42]	1. Avoids over-segmentation 2. Smooth sub-mesh boundaries	1. Non-automatic (requires user interaction) 2. Non-parameterizable sub-meshes
Random Walks [43]	1. Automatic segmentation 2. Works on both mechanical and organic meshes 3. Avoids over-segmentation 4. Smooth sub-mesh boundaries	1. Non-parameterizable sub-meshes
Fitting Primitives [44]	1. Automatic segmentation 2. Smooth sub-mesh boundaries 3. Parameterizable sub-meshes for RE	1. Does not work properly on organic meshes 2. Does not work properly on mechanical meshes composed by several freeform surfaces
Contour Based Segmentation [13]	1. Works on both mechanical and organic meshes 2. Avoids over-segmentation	1. Non-parameterizable sub-meshes 2. Non-smooth sub-mesh boundaries
Autodesk® 3ds Max®	1. Automatic segmentation 2. Parameterizable sub-meshes	1. Does not work properly on organic meshes 2. Over-segmentation
Geomagic® Design™	1. Automatic segmentation 2. Smooth sub-mesh boundaries 3. Parameterizable sub-meshes for RE	1. Does not work properly on organic meshes 2. Over-segmentation on meshes with a lot of small features (such as blending surfaces)
Our (Temp-Geom) algorithm	1. Automatic segmentation 2. Avoids over-segmentation 3. Smooth sub-mesh boundaries 4. Parameterizable sub-meshes for RE	1. Does not work properly on organic meshes 2. Ignores small feature surfaces

to be automatic if it does not require interactive input of the user to compute the result. However, it is very common for automatic algorithms (including ours, see Table 1) to require the use of at least one input parameter (prior to segmentation) which is used by the algorithm to internally perform numerical decisions during the segmentation.

6. Conclusions and future work

This manuscript presents an algorithm for automatic mesh segmentation of 3D meshes of digitized mechanical pieces for RE applications. The implemented algorithm articulates a dihedral / heat transfer-based segmentation with a Hessian-based parameterization.

Compared to similar approaches, our method improves the RE workflow with an automatic hybrid geometry / topology approach which segments triangular meshes acquired from scanned mechanical models. The geometric component of the algorithm (i.e., dihedral criterion) favors the parameterizability of the resulting partition. On the other hand, the topologic component (captured by the temperature fields) favors smooth transitions between submeshes and avoids over-segmentation. The experiments were conducted on data acquired by a 3D optical scanner and from public repositories, and yet resulted in sets of fully parameterizable submeshes.

Ongoing work addresses: (1) Detection of cylinder-like submeshes and automatic computation of virtual boundaries as our algorithm currently requires user guidance to parameterize such cases. (2) Design of an alternative method to compute temperature seeds in the mesh to allow the segmentation of organic meshes. (3) Triangle negotiation / splitting between adjacent submeshes in order to produce smoother sub-mesh boundary curves, in preparation of cleaner B-Reps. (4) Definition of a consistent topology (SHELL, FACES, LOOPS, EDGES and VERTICES) and geometry (freeform curves and surfaces) which together compose the final B-Rep of the reconstructed model.

Acknowledgments

We thank our colleagues from GKN Legazpia who provided material for the experiments. We also want to thank B. Zabala for contributing with the segmentation using commercial CAD software.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] Posada J, Toro C, Barandiaran I, Oyarzun D, Stricker D, de Amicis R, Pinto EB, Eisert P, Döllner J, Vallarino I. Visual computing as a key enabling technology for industrie 4.0 and industrial internet. IEEE Comput Graph Appl 2015;35(2):26–40. doi:10.1109/MCG.2015.45.

[2] Chintala G, Gudimetla P. Optimum material evaluation for gas turbine blade using reverse engineering (re) and fea. Proc Eng 2014;97:1332–40. doi:10.1016/j.proeng.2014.12.413.

[3] Waggoner T. Exporting solid models into finite element analysis (fea) for reverse engineering of electrical components. In: Proceedings of the electrical insulation conference and electrical manufacturing expo; 2005. p. 231–5. doi:10.1109/EEIC.2005.1566297.

[4] García MJ, Boulanger P, Henao M. Structural optimization of as-built parts using reverse engineering and evolution strategies. Struct Multidiscip Optim 2008;35(6):541–50. doi:10.1007/s00158-007-0122-6.

[5] D. Vučina, I. Pehnec, Enhanced Reverse Engineering Using Genetic-Algorithms-Based Experimental Parallel Workflow for Optimum Design, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 172–183. doi:10.1007/978-3-642-20520-0_18.

[6] Asiabanpour B, Ardis A, Andrade AA. A systematic use of reverse engineering in evaluating the overall accuracy of the fabricated parts. Int J Rapid Manuf 2014;4(2–4):165–78. doi:10.1504/IJRAPIDM.2014.066006.

[7] Mejia D, Sánchez JR, Segura Á, Ruiz-Salguero O, Posada J, Cadavid C. Mesh segmentation and texture mapping for dimensional inspection in web3d. In: Proceedings of the twenty-second international conference on 3d web technology. In: Web3D '17. New York, NY, USA: ACM; 2017. 3:1–3:4. doi: 10.1145/3055624.3075954.

[8] Orozco S, Formella A, Cadavid CA, Ruiz-Salguero O, Osorno M. Geometry and topology-based segmentation of 2-manifold triangular meshes in r3. Br J Appl Sci Technol 2017;21(1):1–14. doi:10.9734/BJAST/2017/32827.

[9] Wang J, Gu D, Yu Z, Tan C, Zhou L. A framework for 3d model reconstruction in reverse engineering. Comput Ind Eng 2012;63(4):1189–200. doi:10.1016/j.cie.2012.07.009.

[10] Ruiz OE, Mejia D, Cadavid CA. Triangular mesh parameterization with trimmed surfaces. Int J Interact Des Manuf (IJIDeM) 2015;9(4):303–16. doi:10.1007/s12008-015-0276-1.

[11] Bénière R, Subsol G, Gesquière G, Breton FL, Puech W. A comprehensive process of reverse engineering from 3d meshes to cad models. Comput Aided Des 2013;45(11):1382–93. doi:10.1016/j.cad.2013.06.004.

[12] Mejia D, Ruiz-Salguero O, Cadavid CA. Hessian eigenfunctions for triangular mesh parameterization. In: Proceedings of the eleventh joint conference on computer vision, imaging and computer graphics theory and applications (VISIGRAPP); 2016. p. 75–82. doi:10.5220/0005668200730080.

[13] Rodrigues RS, Morgado JF, Gomes AJ. A contour-based segmentation algorithm for triangle meshes in 3d space. Comput Graph 2015;49:24–35. doi:10.1016/j.cag.2015.04.003.

[14] Xiao D, Lin H, Xian C, Gao S. Cad mesh model segmentation by clustering. Comput Graph 2011;35(3):685–91. doi:10.1016/j.cag.2011.03.020.

[15] Wang J, Yu Z. Surface feature based mesh segmentation. Comput Graph 2011;35(3):661–7. doi:10.1016/j.cag.2011.03.016.

[16] Yi B, Liu Z, Tan J, Cheng F, Duan G, Liu L. Shape recognition of cad models via iterative slippage analysis. Comput Aided Des 2014;55:13–25. doi:10.1016/j.cad.2014.04.008.

[17] Zhang H, Van Kaick O, Dyer R. Spectral mesh processing. Comput Graph Forum 2010;29(6):1865–94. doi:10.1111/j.1467-8659.2010.01655.x.

[18] Mejia D, Ruiz-Salguero O, Cadavid CA. Spectral-based mesh segmentation. Int J Interact Des Manuf (IJIDeM) 2017;11(3):503–14. doi:10.1007/s12008-016-0300-0.

[19] Golovinskiy A, Funkhouser T. Consistent segmentation of 3d models. Comput Graph 2009;33(3):262–9. doi:10.1016/j.cag.2009.03.010.

[20] Liu X, Zhang J, Liu R, Li B, Wang J, Cao J. Low-rank 3d mesh segmentation and labeling with structure guiding. Comput Graph 2015;46:99–109. doi:10.1016/j.cag.2014.09.019.

[21] Le T, Bui G, Duan Y. A multi-view recurrent neural network for 3d mesh segmentation. Comput Graph 2017;66(Supplement C):103–12. Shape Modeling International 2017. https://doi.org/10.1016/j.cag.2017.05.011"

[22] Wang H, Lu T, Au OK-C, Tai C-L. Spectral 3d mesh segmentation with a novel single segmentation field. Graph Model 2014;76(5):440–56. doi:10.1016/j.gmod.2014.04.009.

[23] Liao T, Li X, Xu G, Zhang YJ. Secondary laplace operator and generalized Giaquinta–Hildebrandt operator with applications on surface segmentation and smoothing. Comput Aided Des 2016;70:56–66. doi:10.1016/j.cad.2015.07.009.

[24] Zhang J, Zheng J, Wu C, Cai J. Variational mesh decomposition. ACM Trans Graph 2012;31(3). 21:1–21:14. doi: 10.1145/2167076.2167079.

[25] Zheng Y, Tai C-L. Mesh decomposition with cross-boundary brushes. Comput Graph Forum 2010;29(2):527–35. doi:10.1111/j.1467-8659.2009.01622.x.

[26] Gěbal K, Bærentzen JA, Aanæs H, Larsen R. Shape analysis using the auto diffusion function. Comput Graph Forum 2009;28(5):1405–13. doi:10.1111/j.1467-8659.2009.01517.x.

[27] Sun J, Ovsjanikov M, Guibas L. A concise and provably informative multi-scale signature based on heat diffusion. In: Proceedings of the symposium on geometry processing. In: SGP '09. Aire-la-Ville, Switzerland: Eurographics Association; 2009. p. 1383–92.

[28] Benjamin W, Polk AW, Vishwanathan S, Ramani K. Heat walk: robust salient segmentation of non-rigid shapes. Comput Graph Forum 2011;30(7):2097–106. doi:10.1111/j.1467-8659.2011.02060.x.

[29] Fang Y, Sun M, Kim M, Ramani K. Heat-mapping: a robust approach toward perceptually consistent mesh segmentation. In: Proceedings of the conference on computer vision and pattern recognition (CVPR); 2011. p. 2145–52. doi:10.1109/CVPR.2011.5995695.

[30] Meng M, Fan L, Liu L. A comparative evaluation of foreground/background sketch-based mesh segmentation algorithms. Comput Graph 2011;35(3):650–60. doi:10.1016/j.cag.2011.03.038.

[31] Vanco M, Brunnett G. Direct segmentation of algebraic models for reverse engineering. Computing 2004;72(1):207–20. doi:10.1007/s00607-003-0058-7.

[32] Soni K, Chen D, Lerch T. Parameterization of prismatic shapes and reconstruction of free-form shapes in reverse engineering. Int J Adv Manuf Technol 2009;41(9):948. doi:10.1007/s00170-008-1550-1.

[33] Yang Y-J, Zeng W, Chen J-F. Equiareal parameterizations of NURBS surfaces. Graph Models 2014;76(1):43–55. doi:10.1016/j.gmod.2013.10.007.

[34] Yang Y-J, Zeng W, Meng X-X. Conformal freeform surfaces. Comput Aided Des 2016;81:48–60. doi:10.1016/j.cad.2016.09.003.

[35] Pinkall U, Polthier K. Computing discrete minimal surfaces and their conjugates. Exp Math 1993;2(1):15–36. doi:10.1080/10586458.1993.10504266.

[36] Reuter M, Biasotti S, Giorgi D, Patanè G, Spagnuolo M. Discrete Laplace Beltrami operators for shape analysis and segmentation. Comput Graph 2009;33(3):381–90. IEEE International Conference on Shape Modelling and Applications 2009. http://doi.org/10.1016/j.cag.2009.03.005.

[37] Spielman DA, Teng S-H. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In: Proceedings of the thirty-sixth annual ACM symposium on theory of computing. In: STOC '04. New York, NY, USA: ACM; 2004. p. 81–90. doi:10.1145/1007352.1007372.

[38] Schmitz PG, Ying L. A fast direct solver for elliptic problems on general meshes in 2d. J Comput Phys 2012;231(4):1314–38 https://doi.org/10.1016/j.jcp.2011.10.013.

[39] Donoho DL, Grimes C. Hessian eigenmaps: locally linear embedding techniques for high-dimensional data. P Natl Acad Sci USA 2003;100(10):5591–6. doi:10.1073/pnas.1031596100.

[40] Regli WC, Foster C, Hayes E, Ip CY, Mcwherter D, Peabody M, Shapirsteyn Y, Zaychik V. National design repository project: A status report. In: Proceedings of the international joint conferences on artificial intelligence (IJCAI) and AAAI/SIGMAN workshop on AI in manufacturing systems; 2001.

[41] Chen X, Golovinskiy A, Funkhouser T. A benchmark for 3d mesh segmentation. ACM Trans Graph 2009;28(3). 73:1–73:12. doi: 10.1145/1531326.1531379.

[42] Golovinskiy A, Funkhouser T. Randomized cuts for 3d mesh analysis. ACM Trans Graph 2008;27(5). 145:1–145:12. doi: 10.1145/1409060.1409098.

[43] Lai Y-K, Hu S-M, Martin RR, Rosin PL. Fast mesh segmentation using random walks. In: Proceedings of the ACM Symposium on Solid and Physical Modeling. In: SPM '08. New York, NY, USA: ACM; 2008. p. 183–91. doi:10.1145/1364901.1364927.

[44] Attene M, Falcidieno B, Spagnuolo M. Hierarchical mesh segmentation based on fitting primitives. Vis Comput 2006;22(3):181–93. doi:10.1007/s00371-006-0375-x.