Computers & Graphics (2024)

Contents lists available at ScienceDirect

Computers & Graphics

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

On the shape description of general solids using Morse theory

Juan Pareja-Corcho[a,d,*], Diego Montoya-Zapata[b], Aitor Moreno[a], Carlos Cadavid[c], Jorge Posada[a], Ketzare Arenas-Tobon[b], Oscar Ruiz-Salguero[a,b]

[a] *Vicomtech Foundation, Basque Research and Technology Alliance (BRTA), Mikeletegi 57, 20009 Donostia-San Sebastian, Spain*
[b] *CAD CAM CAE Laboratory, Universidad EAFIT, Cr 49 Cl 7-Sur 50, 050022 Medellin, Colombia*
[c] *Mathematics and Applications, Universidad EAFIT, Cr 49 Cl 7-Sur 50, 050022 Medellin, Colombia*
[d] *Faculty of Informatics, University of the Basque Country (UPV/EHU), Manuel Lardizabal 1, 20018 Donostia-San Sebastian, Spain*

ARTICLE INFO

Article history:
Received June 14, 2024

Keywords: Shape description, Handle decomposition, Morse theory, Solid geometry

ABSTRACT

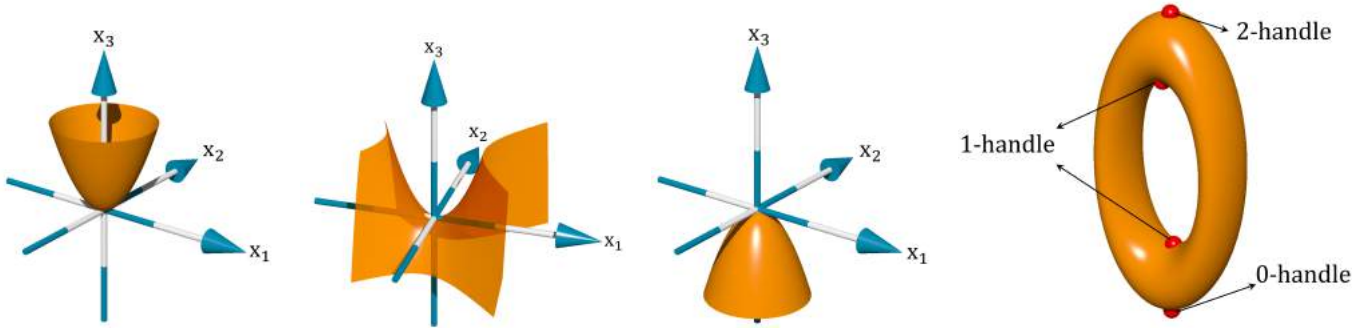
The automatic shape description of solids is a problem of interest in manufacturing engineering, amongst other related areas. This description can be either geometrical or topological in nature and can be applied to either surfaces or solids (embedded manifolds). Topological descriptions are specially interesting for the problem of shape comparison and retrieval, where one wants to know if a given shape resembles some other known shape. Some popular topological descriptions use Morse theory to study the topology of manifolds and encode their shape characteristics. A Morse function f is defined on the manifold and the manifold's shape is indirectly studied by studying the behaviour of the critical points of f. This family of methods is well defined for surfaces but doesn't consider the case of solids. In this paper we address the topological description of solids using Morse theory. Our methodology considers three cases: solids without internal boundaries, solids with internal boundaries and thin-walled solids. We present an algorithm to identify topological changes on these solids using the principle of shape decomposition by Morse handles. The presented algorithm deals with Morse functions that produce parallel planar level sets. Future endeavours should consider other candidate functions.

1. Introduction

Many tasks in computational-aided engineering require the shape description of solid pieces (or geometries in general) in a condensed and efficient way. Tasks such as surface matching [1], feature detection [2], collision detection [3] or fabrication planning [4] use condensed shape descriptions to perform calculations efficiently. These descriptors use mainly two sources of information: geometry and topology. Geometrical descriptors focus on splitting the pieces into regions that share some common geometrical trait such as curvature [5, 6] or geodesic distance from a given point [7].

Other approaches describe the model by considering its topological features [8, 9]. Amongst these methods, the Reeb graph is the prime example [9, 10]. It studies the topological evolution of a collection of level sets defined by a real-valued function on the manifold. The Reeb graph descriptor is suitable to describe surface-based pieces [11, 12] but ill suited to describe solids in a general sense. Other family of methods combine both geometrical and topological description [13] by using semantic information to describe subsets of the piece that have a special significance in the interpretation of the model (e.g. the decomposition of the human body into head, torso, legs, arms). These, however, are methods specifically tailored to certain shapes and are not suitable for general models.

*Corresponding author
e-mail: jcpareja@vicomtech.org (Juan Pareja-Corcho)



(a) 0-handle (b) 1-handle (c) 2-handle (d) Critical points on the standing torus.

Fig. 1: Local shape of function f according to the Morse handle classification.

This paper is based on a previous work [14] that addressed the shape decomposition of solids with inner voids. The presented algorithm used Morse handle identification in a level set sequence to encode the shape of such solids. In this work we extend this approach by considering new cases of topological transitions that might appear in the level set sequence and their treatment in the algorithm. Additionally, new datasets, including thin-walled solids, are presented to showcase the application of our methodology.

2. Morse theory on 2-manifolds

Let's consider first the case of surfaces (i.e. 2-manifolds). Let M be a smooth manifold and $f : M \to \mathbb{R}$ be a smooth function defined on M. We have the following definitions:

Level sets: The level set of f on M at a value $a \in \mathbb{R}$ is $\{p \in M : f(p) = a\}$. We denote the level set of f at value a as $\Pi_{f,a}$. Connected parts of a level set are called *level set components*.

Critical points and values: A critical point of the function f is a point p where $dF(p) = 0$. A real c is called a critical value of f if the pre-image $f^{-1}(c)$ contains a critical point of f. Additionally, a critical point is said to be non-degenerate if it's Hessian is non-degenerate.

Morse functions: Function f is said to be a Morse function if it satisfies the following conditions [15]:

1. all critical points of f are non-degenerate.

$$\forall p \in \mathcal{M} : \nabla_{\mathcal{M}} f(p) = 0 \to \det\left(\mathbf{H}_{\mathcal{M}} f(p)\right) \neq 0 \qquad (1)$$

2. for all pairs (p, q) of different critical points of f, $f(p) \neq f(q)$.

Morse lemma: Assume p to be a non-degenerate critical point of f. There is a *local* coordinate system $\{X_1, \ldots, X_n\}$ on a neighborhood N_p of p, such that on N_p [15]:

$$f(X_1, \ldots, X_n) = -X_1^2 - \ldots - X_k^2 + X_{k+1}^2 + \ldots + X_n^2 \qquad (2)$$

Value k is known as the *index* of f at p. In the case of surfaces the critical points of a Morse function can be classified as a minimum ($k = 0$), a saddle ($k = 1$) or a maximum ($k = 2$), as shown in Figure 1. A three-dimensional manifold has four types of non-degenerate critical points: minimum ($k = 0$), 1-saddle ($k = 1$), 2-saddle ($k = 2$) and maximum ($k = 3$). If the manifold is closed then the number of non-degenerate critical points is always finite [16] and, as a corollary of the Morse lemma, they are isolated. These facts allow us to encode the shape of any manifold by tracking the appearance of these critical points and their handle classifications. Up to this point, however, critical points are individually considered. To introduce an order and link them together we introduce the concept of handle decomposition.

2.1. Handle decomposition of 2-manifolds

In simple terms, any compact manifold can be expressed as an ordered sum of submanifolds called handles [17]. Let M be a smooth manifold and $f : M \to \mathbb{R}$ be a Morse function defined on M. We define the following:

Lower level sets: The lower level set M_t of f in M at value t is the set $M_t = \{x \in M : f(x) \leq t\}$.

Theorem 1: For two reals a, b with $a < b$ if f has no critical values in the interval $[a, b]$, then the manifolds M_a and M_b are diffeomorphic.

The previous theorem implies that the topology of M_t does not change as parameter t passes through regular (non-critical) values of f.

Theorem 2: Let p be a critical point of f and $f(p) = t$ be its critical value. Let ϵ be a number small enough so that f does not have critical values in $[t - \epsilon, t + \epsilon]$. The manifold $M_{t+\epsilon}$ can be obtained by *gluing* a handle (a contractible smooth manifold) to the manifold $M_{t-\epsilon}$.

Theorem 2 implies that the topology of M_t changes as t passes through a critical value of f. This change is incorporated into M_t by adding a new submanifold according to the nature of the critical point encountered. See [17] for an extensive elaboration on Thereoms 1 and 2 and handle decomposition in general. Since a Morse function defined on a compact manifold has only finitely many critical points, the topology of M_t changes a finite number of times and its shape can be described by describing the sequence of *handles* glued together as parameter t increases:

Handle decomposition: A handle decomposition of a compact manifold M is a finite sequence of manifolds $W_0, ..., W_l$ such that:

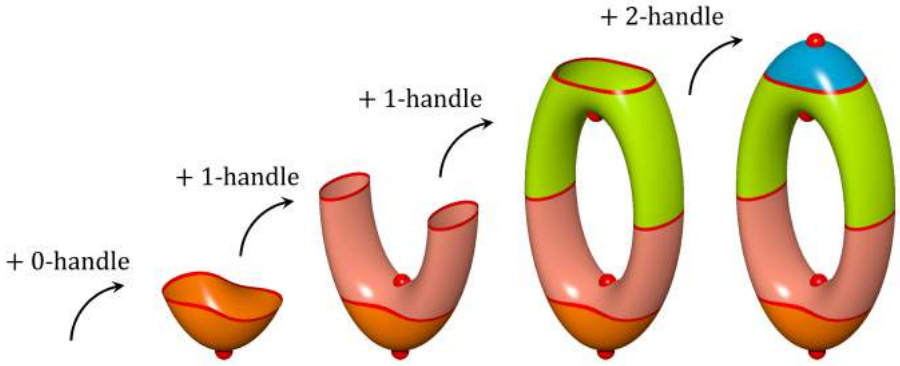
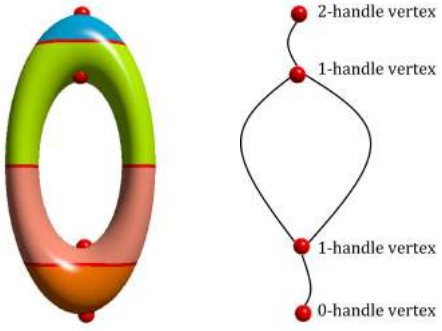


Fig. 2: Handle decomposition of the standing torus.

1. $W_0 = \emptyset$,
2. W_l is diffeomorphic to M,
3. W_i is obtained from W_{i-1} by attaching a handle.

The shape of the handle necessary to obtain $W_i = M_{t+\epsilon}$ from $W_{i-1} = M_{t-\epsilon}$ depends on the index of the critical point in the interval $[t-\epsilon, t+\epsilon]$. Therefore, to describe the shape of a surface we only need to consider a sequence of 0-handles, 1-handles and 2-handles. See Figure 2 for an example. As we will detail later, the case for solids is not as simple. The reason for this is that, in the case of solids embedded in $\mathbb{R}^3$, changes in topology arise from both changes in the number of connected components in a level set and changes in the topology of the level set itself (i.e. changes in the genus of isosurfaces). Most algorithms that describe the shapes of 2-manifold by their handle sequence exploit a data structure known as the *Reeb graph*.

2.2. Reeb graphs



(a) Critical points. (b) Reeb Graph $R(f)$.

Fig. 3: Critical points and Reeb Graph $R(f)$ of the standing torus.

Reeb graph [18]: Two points $p, q \in \mathcal{M}$ are equivalent if they belong to the same connected component of $f^{-1}(c)$ with $c = f(p) = f(q)$. The Reeb Graph of f, $R(f) = \mathbb{X}_{\sim}$, is the quotient space defined by this equivalence relation.

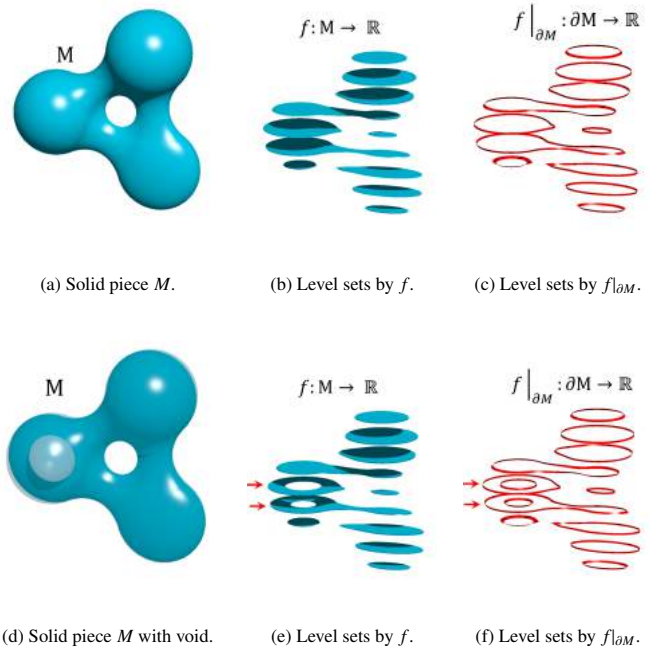
A node in the Reeb graph represents a critical point of f on M and the degree of the node is related to the index of the critical point it represents. Minimums and maximums ($k = 0, 2$) are shown in the Reeb graph as nodes of degree 1 and saddles ($k = 1$) are shown as nodes of degree 3. The arcs of the graph correspond to regions of M in which the topology of M does not change. See Figure 3 for an example of the Reeb graph of the standing torus. Intuitively, the nodes of the Reeb graph are built by contracting critical level sets to a single point and connecting those points according to the number of connected components in subsequent level sets. Various algorithms exist for the automated extraction of Reeb graphs [19]. Some of these algorithms for the case of surfaces can be found in [20, 21, 22, 23, 24, 25]. The most efficient algorithms complexity-wise can be found in [26, 27]. Reeb graphs are popular tools to decompose a surface into its handle constituents [10]. Some theoretical works exists on generalizations of Reeb graphs for multivariate continuous functions [28].

3. Using Morse Theory on Solids

Even though the Morse shape description is well defined for the case of surfaces, very little attention has been put into applying it in general solids. The main difference is that solids in general can be described by more than one surface boundary components (i.e. solids with internal voids). As we will show later, even in the case of solids with only one boundary component, the genus of the boundary surface might introduce ambiguities in the interpretation of the critical points from a level set representation.

Easiest solids to deal with, at least in principle, are those with only one boundary component. This is because these solids can be defined as the volume enclosed by a single surface and the topological structure of the surface boundary will reflect the topology of the solid itself. In simple terms, one could encode the shape of single-boundary solids by studying the shape of its boundary component using 0-, 1- and 2-handle critical point classification without ambiguities. As an example of this, consider the case shown in Figure 4a.

Let M be a solid (3-manifold with boundary) embedded in $\mathbb{R}^3$ with a single boundary component and $f_M : M \to \mathbb{R}$ a Morse function defined on M. Consider only functions f that do not possess critical points in the interior of the solid. Let function $f|_{\partial M}$ be the restriction of f to the boundary of M. In this case, for every non-critical value t the level sets $\Pi_{f,t}$ and $\Pi_{f|_{\partial M},t}$ are



(a) Solid piece M. (b) Level sets by f. (c) Level sets by $f|_{\partial M}$.

(d) Solid piece M with void. (e) Level sets by f. (f) Level sets by $f|_{\partial M}$.

Fig. 4: Example with a solid piece with and without inner void. Level sets produced by height function f and its restriction to the border $f|_{\partial M}$.

closely related. The level set components of $\Pi_{f|_{\partial M},t}$ consist of closed curves on the boundary ∂M. These curves on ∂M bound regions of M that match with the level set components of $\Pi_{f,t}$. This means that the number of level set components in $\Pi_{f,t}$ and $\Pi_{g,t}$ is the same (Figures 4b and 4c) and therefore the topology of M can be studied by studying the topology of its boundary ∂M. This is exploited to produce shape descriptions of single-boundary solids by considering the topology of its boundary surface [29, 30, 31, 16].

Now consider the case of solids with multiple boundary components (Figure 4d). In this case the boundary ∂M of M is composed by two distinct surfaces: an interior boundary and an exterior boundary. The level set components of $\Pi_{f|_{\partial M},t}$ (curves on the boundaries) do not bound a level set component of $\Pi_{f,t}$ in a one-to-one correspondence. Notice that both components of $\Pi_{f|_{\partial M},t}$ bound the same component of $\Pi_{f,t}$, one from the inside and the other from the outside. This means that the unequivocal one to one relationship between the solid level sets and the boundary level sets is lost and therefore additional information is required to study these phenomena. Count, for example, the number of connected components in the highlighted level sets in Figures 4e and 4f. In some level set values, the solid slicing has a lower component count than its surface counterpart. Notice, however, that the nature of the Morse handle classification is not violated: topological events on the boundary components are still well described by the Morse theory but the general shape interpretation requires to distinguish between events in the internal and external borders.

This distinction is important to encode the shape of solids with internal voids. Some works in the extraction of Reeb graphs using level set/contouring approaches for solid decomposition exist, see for example the works in [20], [2] and [9]. These works, however, do not consider the internal void situation. The oldest approach by Shinagawa and Kunii [20] connects contours between consecutive level sets using a distance criteria but do not consider the topological structure of each level set (e.g. its genus). In our approach we substitute the distance criteria by a 2-dimensional similarity criteria for enhanced stability. The work by Takahashi et al. [2] performs a tetrahedralization of a volumetric region and extracts the critical points of a field function defined on the region represented by its discrete samples (level sets/isosurfaces). This differs from our method that we do not consider functions over volumetric regions but only functions defined on a 2d manifold embedded in 3d space. A recent work by Strodthoff and Juttler [9] presents a methodology to extract Reeb graphs for solids with straight tunnels. Notice that these tunneled-solids are not strictly solids with internal voids yet present similar challenges for their decomposition. This tunneled-solids are similar to the special 1-handle transition we discuss in Section 3.2.

The encoding of topological changes related to the genus of the level sets requires, as we show, some additional information to that provided by the Reeb graph. One could consider the solid's boundaries separately and compute a Reeb graph for each boundary using the previously cited algorithms. However, reconstructing a single Reeb graph (or any similar encoding) from these two separated and unrelated sets of critical points is a non-trivial task or could not be possible at all (e.g. the Reeb graph definition does not allow such level set genus-related changes). Some works other introduce the concept of *augmented* or *extended* Reeb graphs (e.g. [32]). Most of these, however, augment or extend the Reeb graph with information not related to the pure shape description but to other aspects such as textures and local geometric measurements. One important work by Strodthoff and Juttler [33] introduces *layered* Reeb graphs, which are discrete representations of Reeb spaces [28]. These structures can encode the kind of topological changes that we address but require a double slicing (given that one deals now with two functions instead of one) that our proposed methodology does not require.

3.1. Augmented handle set

To describe the shape of solids with multiple internal boundaries we propose an **augmented** collection of handles that describe the shape of a general solid M. This is equivalent to finding the sequence of manifolds $W_0, ..., W_k$ such that

1. $W_0 = \emptyset$,
2. W_k is homeomorphic to M,
3. W_i is obtained from W_{i-1} by attaching a **solid handle**,
4. Each **solid handle** attached incarnates a critical point of f in ∂M.

Notice that the problem statement requires that the critical values of f are unique (i.e. only one topological change is allowed between consecutive level sets). This means that for any two critical points p, q of f then $f(p) \neq f(q)$. Therefore, it is always possible to order the critical points of f in ascending order $f(p_0) < ... < f(p_k)$. The proposed set of solid handles is depicted in Figure 5, as well as the effect of each piece in the level set collection of the solid.

The solid handles presented in Figures 5a, 5b, 5c and 5d correspond to topological events of the external boundary. The solid handles presented in Figures 5e, 5f, 5g and 5h describe topological events on the internal boundaries. All solid handles are volume versions of the local shape of function f on either internal or external boundary surfaces. Before considering the automatic identification of the proposed solid handles we detail a type of transition not considered in our previous work [14] and that could introduce ambiguity in the identification of which solid handle to use to model a topological change.

3.2. A special case of the 1-handle

Some problems may arise when the solid has tunnels with axis not parallel to the slicing plane [9]. This is specially critical in the consideration of solids where the external boundary surface have a genus greater than zero. We present one of such solids in Figure 6a and its level set representation in Figure 6b. In the level set collection a topological event takes place between slices Π_2 and Π_3. If we consider this event using the component count effect defined by the solid handle set in Figure 5 we identify the event as modeled by a *internal 0-handle*, but we see in the solid this is not the case, as the *internal 0-handle* indicates the appearance of an strictly internal border whereas the presented solid has no internal border. This ambiguity is undesirable for the shape description pipeline.

Taking a closer look on the event described (Figure 7), we see that the mechanism that produces contour B from contour A is of the type *1-handle* (contour A *glues* to itself). To correctly classify this event as a 1-handle event and not as an internal 0-handle (using only its level set effect) two geometrical observations can be used:

1. For this special kind of 1-handle events the contours after separation are very close to each other, therefore a minimum distance criteria could be useful to separate them from internal 0-handles events, which can happen everywhere inside a contour.
2. Shape similarity between the external contour before and after the event should be considerably lower than in the case of internal 0-handle events which barely modify the shape of external contours.

These observations are used to enhance the algorithm presented in our previous work [14]. We describe these enhancements in the following section.

4. The identification algorithm

In this section we present the algorithm to identify the solid handles necessary to describe the shape of general solids. Figure 8 shows a summary of the proposed algorithm. In the following subsections we detail each step and provide the pseudocode necessary to complete each task.

4.1. Contour orientation and inclusion

The slicing function f extracts spatial curves (contours) from all boundaries of the solid. For the sake of efficiency, as the slicing proceeds we record the information about the origin of each contour (whether it comes from an internal or an external surface). This is straightforward (by storing the normal vector with respect to the surface) if the Boundary Representation is well defined. In the fashion of [34], the orientation of each contour is defined by the projection of the surface normal vector to the slicing plane. Consider the slicing plane intersects a triangle $t \in M$, then:

$$\vec{n}_{xy}(t) = Proj_{xy}(\vec{n}(t)) \tag{3}$$

A connected component of a level set $f^{-1}(c)$ can be: (i) a closed 1-manifold (closed curve), (ii) a 0-manifold (point) or (iii) a 2-manifold region (the triangle itself when it coincides with the slicing plane). In the degenerate cases (ii and iii) no orientation can be produced and these components can be safely ignored as they will appear in a non-degenerate case in following level sets. Every contour must have *coherent* normal behaviour if the mesh is well defined (i.e. all projected normal vectors must point outwards or inwards with respect to the polygon defined by the closed curve of the contour). The contour forest F_i for a level set Π_i should be constructed from an algorithm like Algorithm 1. This contour forest contains the information about the inclusion relationships between contours.

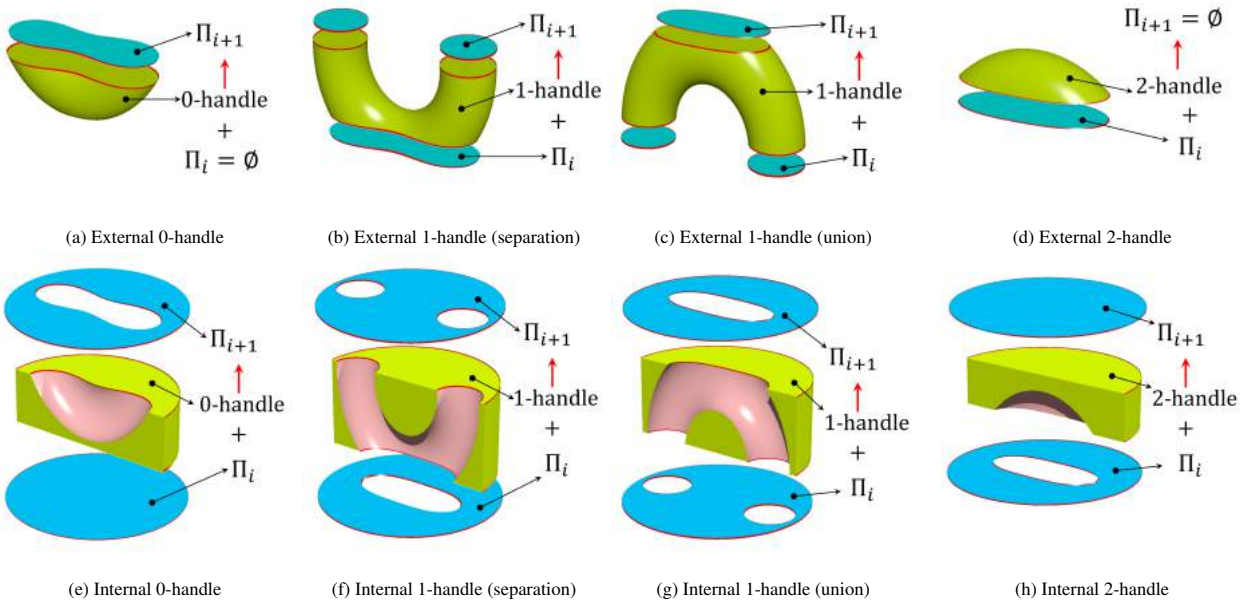
Algorithm 1 Build contour forest F_i for every level set Π_i

```
Π_ext ← external(C ∈ Π_i)
Π_int ← internal(C ∈ Π_i)
F_i ← new Forest
for all contour C_k ∈ Π_ext do
    F_i ← addParent(C_k)
end for
for all contour C_k ∈ Π_int do
    for all contour C_j ∈ Π_ext do
        included ← testInclusion(C_k, C_j)
        if included then
            F_i ← addChild(C_k)
            break
        end if
    end for
end for
```

Algorithm 1 creates the contour forest of a level set assuming external contours as parent nodes (since no external contour can contain other external contour) and internal contours as child nodes. The inclusion test between two polygons (defined by their contours) is a very well studied problem and numerous solutions exist [35]. Every forest constructed this way will have depth no larger than 1. This is because no internal contour can contain other contour as we are not considering the case of annidated solid bodies. Figure 9 shows an example of the contour forest of a level set with several contours.

4.2. Calculation of mapping groups

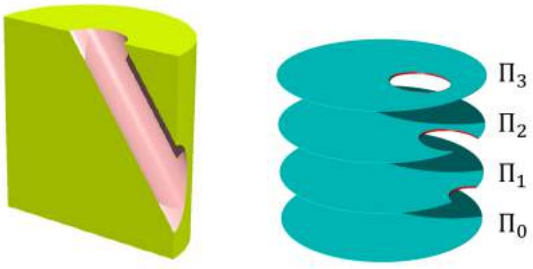
Mapping groups allow us to relate one level set to the next one by identifying contours as natural evolutions of previous contours.This amounts to: given a contour $C_k \in \Pi_i$, find a contour $C_j \in \Pi_{i+1}$ that most resembles a feasible geometrical



(a) External 0-handle (b) External 1-handle (separation) (c) External 1-handle (union) (d) External 2-handle

(e) Internal 0-handle (f) Internal 1-handle (separation) (g) Internal 1-handle (union) (h) Internal 2-handle

Fig. 5: Augmented handle set and their effect on the level sets component population. The *hollow* handles volumes are sliced for visualization purposes.



(a) Sliced version of the solid. (b) Level set collection.

Fig. 6: Example of single-boundary solid with genus > 0. Sliced version is presented for visualization purposes.

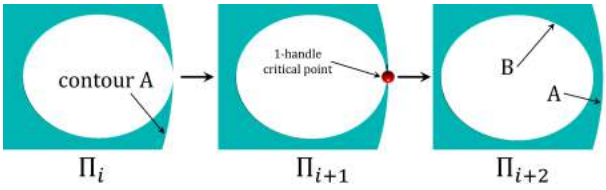


Fig. 7: Special 1-handle event detail.

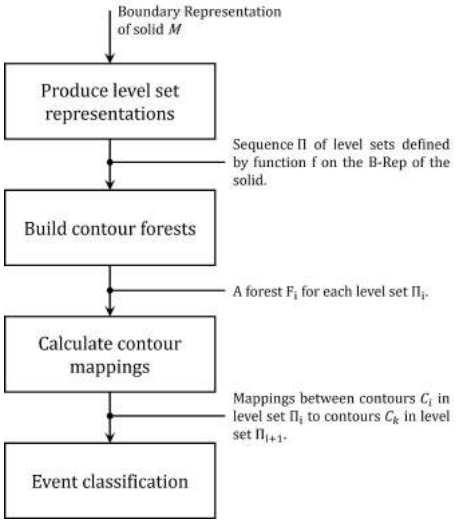


Fig. 8: Summary of the proposed algorithm.

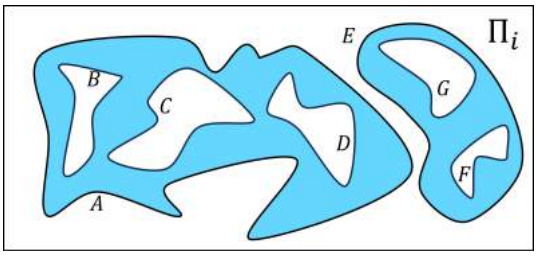
and topological evolution of C_k. Algorithm 2 describes our approach. We use the information from the contour forests from the previous step to avoid unnecessary tests between contours and use a similarity criteria to match two contours. Figure 10 shows an example of a contour mapping. Notice that:

1. External contours in Π_i need only be tested against external contours in Π_{i+1}. No *external to internal* match can exist.
2. Let $C_1 \in \Pi_i$ be an external contour containing an internal contour $C_2 \in \Pi_i$. Let $C_a \in \Pi_{i+1}$ be the matching contour of C_1 in Π_{i+1}. Internal contour $C_2 \in \Pi_i$ can only produce matchings with contours that are contained by C_a. This reduces the number of matching operations needed.

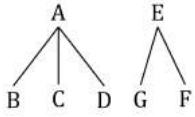
Algorithm 2 builds the contour mappings between consecutive level sets. First, all external contours of level set Π_i are tested against external contours of level set Π_{i+1} and mappings are produced according to a 2-dimensional geometrical similarity criteria. If a contour in Π_i can't find a match in Π_{i+1} a *contour to void* mapping is produced. Every time a contour in Π_{i+1} is added to a map it is marked as already mapped. After all combinations are tested, any contour in Π_{i+1} left unmarked is added to a *void to contour* mapping. For every internal contour in Π_i, we seek the parent external contour in Π_i and find its correspondent external contour in Π_{i+1}. Then the internal contour in Π_i is tested against the children of the correspondent external contour in Π_{i+1}.

Algorithm 2 Calculation of mappings between Π_i and Π_{i+1}

$\Pi_{ext,i} \leftarrow external(C \in \Pi_i)$
$\Pi_{int,i} \leftarrow internal(C \in \Pi_i)$
$\Pi_{ext,i+1} \leftarrow external(C \in \Pi_{i+1})$
$\Pi_{int,i+1} \leftarrow internal(C \in \Pi_{i+1})$
for all contour $C_j \in \Pi_{ext,i}$ **do** ▷ External matchings
  match ← False
  for all contour $C_k \in \Pi_{ext,i+1}$ **do**
    match ← testSimilarity(C_k, C_j)
    if match **then**
      newMap(C_j, C_k)
      markAsMatched(C_k)
    end if
  end for
  if not match **then**
    newMap($C_j, \emptyset$)
  end if
end for
if any $C_k \in \Pi_{ext,i+1}$ not matched **then**
  newMap($\emptyset, C_k$)
end if
for all contour $C_j \in \Pi_{int,i}$ **do** ▷ Internal matchings
  $C_{prnt} \leftarrow$ getParentContour(C_j)
  $C_m \leftarrow$ getMatchInMap(C_{prnt})
  $\Pi_{m,i+1} \leftarrow$ getChildrenContours(C_m)
  match ← False
  for all $C_k \in \Pi_{m,i+1}$ **do**
    match ← testSimilarity(C_k, C_j)
    if match **then**
      newMap(C_j, C_k)
      markAsMatched(C_k)
    end if
  end for
  if not match **then**
    newMap($C_j, \emptyset$)
  end if
end for
if any $C_k \in \Pi_{ext,i+1}$ not matched **then**
  newMap($\emptyset, C_k$)
end if



(a)



(b)

Fig. 9: Contour forest F_i for level set Π_i. (a) Example level set Π_i. (b) Resulting contour forest F_i.

The core of the mapping algorithm is the shape similarity test. The goal of this test is to calculate the degree of geometric resemblance between two contours. This is done by considering the orthogonal projection of the contours to a common plane and calculating a similarity index (Figure 11). Consider contours $C_j \in \Pi_i$ and $C_k \in \Pi_{i+1}$. Their respective orthogonal projections to a common parallel plane are denoted as $C_{j\perp}$ and $C_{k\perp}$. The similarity index is calculated as:

$$similarity(C_j, C_k) = \min\left(\frac{Area\left(C_{j\perp} \cap C_{k\perp}\right)}{Area(C_{j\perp})}, \frac{Area\left(C_{j\perp} \cap C_{k\perp}\right)}{Area(C_{k\perp})}\right) \quad (4)$$

Two contours in consecutive level sets are matched if: a) they are both external or both internal and b) their similarity index is more than a threshold value. In Equation 4 two quantities are calculated: the portion of C_j inside C_k and the portion of C_k inside C_j. The similarity index is defined as the smallest of those two quantities. Some authors define the similarity index as the largest of the two quantities [36] but this results in a less strict test and false positives can be wrongly matched. The intersection approach has proven to be stable and robust [37]. The canonical approach to establish connectivity between contours of subsequent level sets is the distance criteria [20]. This approach establishes a weighted distance function between contours in different level sets and searches for the maximum value of the function for each contour. The similarity approach advantages the distance criteria because the it is independent of the distance between the level sets $d(\Pi_i, \Pi_{i+1})$. This independence is important to preserve the robustness of the matching when changing the slicing density. The slicing density should not, however, be reduced much. A very low density slicing could make the contour shape change too much between consecutive level sets making the similarity criteria unfeasible for the establishment of geometric continuity.

There is no formula to calculate the correct sampling density, however one should have in mind two principles when sampling the objective solid: (a) a greater number of samples will increase the execution time of the algorithm (see Section 5); (b) the sampling must be dense enough to make sure that no

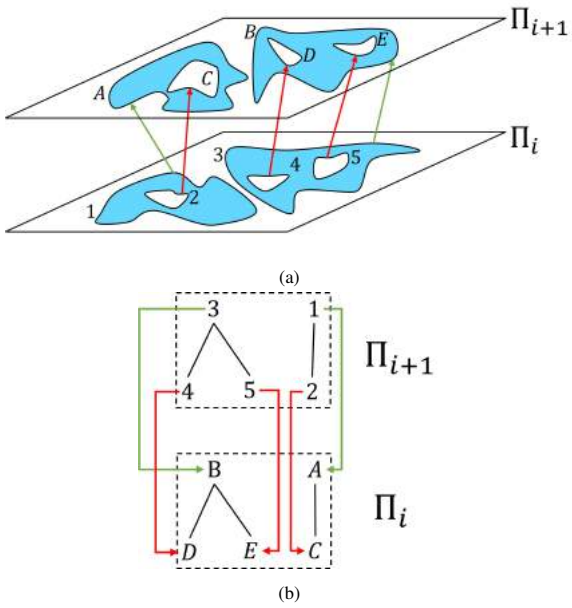


Fig. 10: Example of contour mapping. (a) Level set sequence (Π_i, Π_{i+1}). (b) Maps produced between the forests.

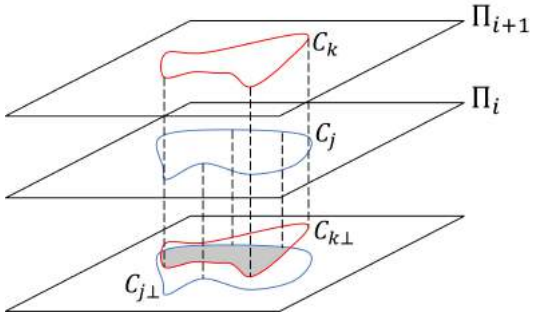


Fig. 11: Projection of contours to calculate the similarity index.

topological changes are missed, this means that between two consecutive slices *at most* one topological change is allowed; (c) even when the slicing is *topologically* correct in the sense of (b), it must be dense enough so that the shape of two consecutive slices does not change too much.

In this case, the selected function is the height function $f(x, y, z) = z$. This choice results in perfectly planar and parallel level sets. Other choices for the Morse functions might produce non-planar level sets, see for example the use of Laplace-Beltrami eigenmaps [38], heat diffusion maps [39] or geodesic maps [16] to produce Reeb graphs and volumetric skeletonizations. Since our methodology relies on the orthogonal projection of the contours to a parallel plane, the proposed algorithm will fail to correctly grasp the geometry of non-planar level sets. A common drawback of the choice of the height function is that the resulting encoding is pose-dependent, unlike those resulting from the use of heat diffusion or eigenvector maps. We believe, however, that the choice of the height function is a reasonable compromise between simplicity, computational efficiency and widespread applications (e.g. additive manufacturing).

4.2.1. Event classification

The calculated maps reveal the existence of topological changes. If all contours in level set Π_i found one (and only one) match in level set Π_{i+1} we say there is no topological change between these level sets. Otherwise, three options appear:

1. A *contour to void* map is present.
2. A *void to contour* map is present.
3. A contour in Π_i or Π_{i+1} is present in more than one map.

Contour to void maps mean that a contour in level set Π_i can't find a match in the next level set. This the case when the surface the contour represents disappears as depicted by Figures 5d and 5h. This event is modeled by the internal and external 2-handles and to distinguish between internal and external events one only needs to consider if the disappearing contour comes from an external or internal boundary. *Void to contour* maps mean that a contour in level set Π_{i+1} was not matched to a preceding contour in level set Π_{i+1}. This is the case for the birth of new surfaces (either external or internal). These are modeled by the 0-handles depicted in Figures 5a and 5e. *Multiple matchings* between contours in consecutive level sets mean the surfaces splitted into various branches. These are modeled by the 1-handle transitions depicted in Figures 5b, 5c, 5f and 5g.

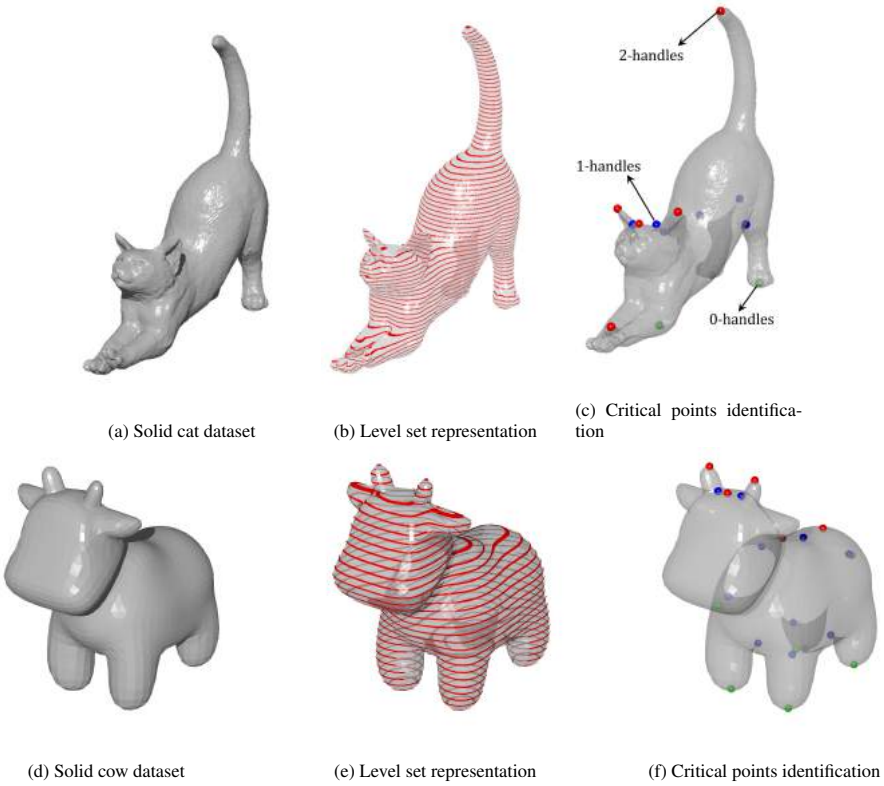
The special case of 1-handle transitions explained in Section 3.2 requires special consideration. This event cannot be distinguished from the internal 0-handle event (or 2-handle depending on the direction of the slice sweep) purely on terms of the mapping contours. Therefore we need to include the geometrical criteria explained in Section 3.2:

1. The minimum distance between contours is calculated and a minimum distance threshold ϵ_{min} defined. All internal 0- and 2-handles events whose minimum distance to their parent external contours is less than this threshold are considered candidates to be identified as special 1-handles.
2. For these candidate events an additional check is made to discard false positives: the similarity index between their parent external contour is evaluated and compared to a threshold that should of course be larger than the matching threshold defined in the matching algorithm. Internal 0- and 2-handle events usually take place with very little effect on the parent external contour. This is not the case for special 1-handle transitions.

These geometrical criteria are, of course, imperfect ones. False negatives might still take place when the tunnel in the external boundary is very small compared to the size of the external parent contour, but as we show in the Experiments section these criteria work well for thin-walled solids. Without the distinction between the internal 0- and 2-handles and the special 1-handle event, the correct description of the shape of these solids would be impossible, specially when the slicing plane is not strictly orthogonal to the tunnel axis.

5. Time complexity

Our approach can be divided in four different processes: (a) slicing, (b) forest build, (c) contour matching, (d) surface retrieval. We will only consider the time complexity of the *forest*



(a) Solid cat dataset (b) Level set representation (c) Critical points identification

(d) Solid cow dataset (e) Level set representation (f) Critical points identification

Fig. 12: Single-boundary solid examples. We show the solid model, its level set representation and the identified critical points of the manifold.

build algorithm and the *contour matching* algorithm as the other two processes are standard and well studied and their theoretical time complexity has been widely reported.

Consider Algorithm 1 with K being the number of level sets in the model, N being the maximum number of internal contours in a single level set and M the maximum number of external contours in a single level set. The worst-case complexity of Algorithm 1 is $O(K * N * M)$. In Algorithm 2, we have a worst-case complexity of $O(K(M^2+N^2))$. Therefore, the overall worst-case complexity of the algorithm in terms of the number of contours and level sets is:

$$O(K(NM + M^2 + N^2)) \tag{5}$$

We believe it is more meaningful to consider complexity with respect to the number of contours and the density of the slicing than with respect to the size of the mesh given that the slicing reduces the mesh (whatever its size) to a number of contours that represent the topological changes of the solid and therefore its handle decomposition. Additionally, this consideration allows us to assess the algorithm's performance for applications in which the contour population does not come from a mesh (e.g. CT scans).

6. Experiments

In this section we present the application of our methodology to additional datasets with respect to our previous work [14]. In these examples we present three types of solids: i) single-boundary solid, ii) thin-walled solid and iii) multi-boundary solid.

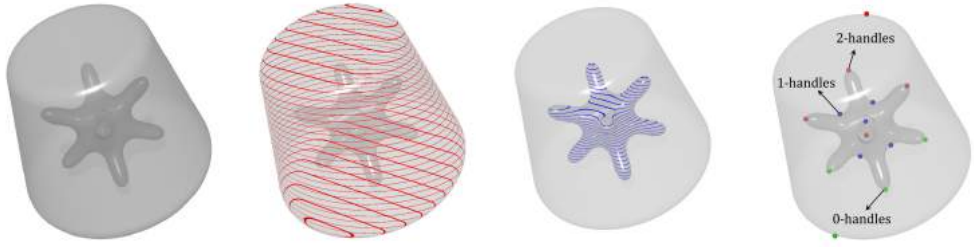
The single-boundary example is shown to show the capabilities of our method to produce a correct critical point identification in the case of surface models. This case is depicted in Figure 12. Figure 12d shows the solid model used. Figure 12e shows the level set representation obtained by slicing the external boundary of the solid model. As explained before, for the case of single-boundary solids describing the critical points of the function f on the boundary surface is equivalent to describing the shape of the solid itself. Figure 12f shows the identified critical points on the example dataset.

The thin-walled solid example is used to showcase the identification of the special 1-handle event. The model is tilted with respect to the horizontal plane to force the appearance of such events. The model used is depicted in Figure 13a and its slicing is depicted in Figure 13b. Figure 13c shows the identified critical points. Thin-walled solids are ubiquitous in industrial applications (e.g. exhaust systems in automotive industry) and the correct description of their shape is crucial in areas like Additive Manufacturing [40]. Our approach correctly identifies key shape features such as the appearance of the sheet flaps and the opening of the internal channel of the piece. The identification of these areas is an important element to feed automated manufacturing planning algorithms.

The third example considers a complex multi-boundary solid (i.e. solid with internal voids). This kind of solid is found in applications such as mold design and manufacturing and their

(a) Thin-walled model (b) Level set representation (c) Critical points identification

Fig. 13: Thin-walled solid example. We show the solid model, its level set representation and the identified critical points of the manifold.



(a) Multi-boundary solid model (b) Level set representation (external boundary) (c) Level set representation (internal boundary) (d) Critical points identification

Fig. 14: Multi-boundary solid example. We show the solid model, the level set representation of its boundary components and the identified critical points of the manifold.

automated shape description is of interest in the area of additive manufacturing of mold components [41]. Figure 14a shows the model used. The model consists of an external rounded cone and an internal star-shaped void. Figures 14b and 14c show the level set representations of the external and internal boundary surfaces respectively. Figure 14d shows the identified critical points on the solid. Our approach correctly identifies the bifurcation, start and ending points of the boundary surfaces. Notice that, in this case, most of the identified critical points are on the internal boundary component.

7. Conclusions

The automatic shape description of solid pieces is a problem of interest for many areas of computer-aided engineering, specially in manufacturing planning. Existing approaches to describe the shapes of pieces are driven by geometrical and/or topological criteria. Most topological approaches study the topological evolution of a collection of level sets defined by a real-valued function on the manifold. With this function one can describe the shape of a given manifold by keeping track of the topological changes that take place in the level set representation. The prime example of these methods is the Reeb graph. However popular, the Reeb graph is ill-suited for the shape description of general solids. In this paper we present a methodology to identify the critical points (and therefore describing the shape) of general solids. The presented methodology is based on a previous work [14] but enhanced with some additional considerations on special 1-handle events, their algorithmical implementation and examples of application on thin-walled solids. The results show that our methodology can produce valid shape descriptions for solids both with and without inner voids and thin-walled solids. The presented experiments where selected to showcase applications of industrial relevance, as the automated shape description of thin-walled solids and mold-like pieces are key in the Additive Manufacturing planning of these elements. Our approach, however, as all approaches guided by Morse theory, is sensitive to the choice of the function f that defines the critical points in the manifold. Future research should include: (i) the evaluation of our method's viability and performance with Morse functions other than the height function such as the eigenvalues of the manifold, (ii) analysis of performance with bigger and more complex data and (iii) robustness to noise analysis.

8. Acknowledgements

We thank students Samuel Martinez-Londono and Jefferson Salcedo-Chavez from Universidad EAFIT for their help in the production of the experiments. This work was partly funded by the Basque Government/Eusko Jarlitza project BIO4CURE (program ELKARTEK, ref- KK-2022/00019).

References

[1] Li, X, Gu, X, Qin, H. Surface mapping using consistent pants decomposition. IEEE Transactions on Visualization and Computer Graphics 2009;15(4):558–571.

[2] Takahashi, S, Takeshima, Y, Fujishiro, I. Topological volume skeletonization and its application to transfer function design. Graphical Models 2004;66(1):24–49.

[3] Mamou, K, Ghorbel, F. A simple and efficient approach for 3d mesh approximate convex decomposition. In: 2009 16th IEEE international conference on image processing (ICIP). IEEE; 2009, p. 3501–3504.

[4] Guo, H, Xu, J, Zhang, S, Zhang, Y, Tan, J. Multi-orientation optimization of complex parts based on model segmentation in additive manufacturing. Journal of Mechanical Science and Technology 2023;37(1):317–331.

[5] Sander, PV, Snyder, J, Gortler, SJ, Hoppe, H. Texture mapping progressive meshes. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. 2001, p. 409–416.

[6] Wang, J, Yu, Z. Surface feature based mesh segmentation. Computers & Graphics 2011;35(3):661–667.

[7] Wang, Z, Bhatia, KK, Glocker, B, Marvao, A, Dawes, T, Misawa, K, et al. Geodesic patch-based segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2014: 17th International Conference, Boston, MA, USA, September 14-18, 2014, Proceedings, Part I 17. Springer; 2014, p. 666–673.

[8] Ma, C, Gao, Y. Study on mesh segmentation of topology optimization results using Reeb graph. In: 2021 International Conference on Artificial Intelligence and Electromechanical Automation (AIEA). IEEE; 2021, p. 277–280.

[9] Strodthoff, B, Jüttler, B. Automatic decomposition of 3D solids into contractible pieces using Reeb graphs. Computer-Aided Design 2017;90:157–167.

[10] Hajij, M, Dey, T, Li, X. Segmenting a surface mesh into pants using Morse theory. Graphical Models 2016;88:12–21.

[11] Agathos, A, Pratikakis, I, Perantonis, S, Sapidis, N, Azariadis, P. 3d mesh segmentation methodologies for cad applications. Computer-Aided Design and Applications 2007;4(6):827–841.

[12] He, C, Wang, C. A survey on segmentation of 3d models. Wireless Personal Communications 2018;102:3835–3842.

[13] Ho, TC, Chuang, JH, et al. Volume based mesh segmentation. Journal of Information Science and Engineering 2012;28(4).

[14] Pareja-Corcho, J, Montoya-Zapata, D, Moreno, A, Cadavid, C, Posada, J, Arenas-Tobon, K, et al. An Approach to the Decomposition of Solids with Voids via Morse Theory. In: Banterle, F, Caggianese, G, Capece, N, Erra, U, Lupinetti, K, Manfredi, G, editors. Smart Tools and Applications in Graphics - Eurographics Italian Chapter Conference. The Eurographics Association. ISBN 978-3-03868-235-6; 2023,doi:10.2312/stag.20231302.

[15] Matsumoto, Y. An introduction to Morse theory; vol. 208. American Mathematical Soc.; 2002.

[16] Biasotti, S, Giorgi, D, Spagnuolo, M, Falcidieno, B. Reeb graphs for shape analysis and applications. Theoretical Computer Science 2008;392(1):5–22. Computational Algebraic Geometry and Applications.

[17] Bohm, N. Morse theory and handle decompositions. University of Chicago Mathematics REU 2019;.

[18] Reeb, G. Sur les points singuliers d'une forme de pfaff completement integrable ou d'une fonction numerique [on the singular points of a completely integrable pfaff form or of a numerical function]. Comptes Rendus Acad Sciences Paris 1946;222:847–849.

[19] Theologou, P, Pratikakis, I, Theoharis, T. A review on 3d object retrieval methodologies using a part-based representation. Computer-Aided Design and Applications 2014;11(6):670–684.

[20] Shinagawa, Y, Kunii, TL. Constructing a Reeb graph automatically from cross sections. IEEE Computer Graphics and Applications 1991;11(06):44–51.

[21] Brandolini, L, Piastra, M. Computing the Reeb graph for triangle meshes with active contours. ICPRAM (2) 2012;12:80–89.

[22] Cole-McLaughlin, K, Edelsbrunner, H, Harer, J, Natarajan, V, Pascucci, V. Loops in Reeb graphs of 2-manifolds. In: Proceedings of the Nineteenth Annual Symposium on Computational Geometry. New York, NY, USA: Association for Computing Machinery; 2003, p. 344–350.

[23] Biasotti, S. Reeb graph representation of surfaces with boundary. In: Proceedings Shape Modeling Applications, 2004. IEEE; 2004, p. 371–374.

[24] Hajij, M, Rosen, P. An efficient data retrieval parallel Reeb graph algorithm. Algorithms 2020;13(10):258.

[25] Pascucci, V, Scorzelli, G, Bremer, PT, Mascarenhas, A. Robust on-line computation of Reeb graphs: Simplicity and speed. ACM Transactions on Graphics 2007;26(3):58.

[26] Parsa, S. A deterministic o (m log m) time algorithm for the reeb graph. In: Proceedings of the twenty-eighth annual symposium on Computational geometry. 2012, p. 269–276.

[27] Harvey, W, Wang, Y, Wenger, R. A randomized o (m log m) time algorithm for computing reeb graphs of arbitrary simplicial complexes. In: Proceedings of the twenty-sixth annual symposium on Computational geometry. 2010, p. 267–276.

[28] Edelsbrunner, H, Harer, J, Patel, AK. Reeb spaces of piecewise linear mappings. In: Proceedings of the twenty-fourth annual symposium on Computational geometry. 2008, p. 242–250.

[29] Doraiswamy, H, Natarajan, V. Output-sensitive construction of Reeb graphs. IEEE Transactions on Visualization and Computer Graphics 2011;18(1):146–159.

[30] Tai, CL, Shinagawa, Y, Kunii, TL. A reeb graph-based representation for non-sequential construction of topologically complex shapes. Computers & Graphics 1998;22(2-3):255–268.

[31] Araújo, C, Cabiddu, D, Attene, M, Livesu, M, Vining, N, Sheffer, A. Surface2volume: Surface segmentation conforming assemblable volumetric partition. arXiv preprint arXiv:190410213 2019;.

[32] Tung, T, Schmitt, F. Augmented reeb graphs for content-based retrieval of 3d mesh models. In: Proceedings Shape Modeling Applications, 2004. IEEE; 2004, p. 157–166.

[33] Strodthoff, B, Jüttler, B. Layered Reeb graphs for three-dimensional manifolds in boundary representation. Computers & Graphics 2015;46:186–197.

[34] Mejia-Parra, D, Ruiz-Salguero, O, Cadavid, C, Moreno, A, Posada, J. Level sets of weak-Morse functions for triangular mesh slicing. Mathematics 2020;8(9).

[35] Feito, F, Torres, JC, Urena, A. Orientation, simplicity, and inclusion test for planar polygons. Computers & Graphics 1995;19(4):595–600.

[36] Lim, CW, Su, Y, Yeo, SY, Ng, GM, Nguyen, VT, Zhong, L, et al. Automatic 4d reconstruction of patient-specific cardiac mesh with 1-to-1 vertex correspondence from segmented contours lines. PloS one 2014;9(4):e93747.

[37] Ruiz, OE, Cadavid, CA, Granados, M, Peña, S, Vásquez, E. 2d shape similarity as a complement for voronoi–delone methods in shape reconstruction. Computers & Graphics 2005;29(1):81–94.

[38] Shi, Y, Lai, R, Krishna, S, Sicotte, N, Dinov, I, Toga, AW. Anisotropic laplace-beltrami eigenmaps: bridging reeb graphs and skeletons. In: 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. IEEE; 2008, p. 1–7.

[39] Hachani, M, Zaid, AO, Puech, W. Kinematic Reeb graph extraction based on heat diffusion. In: 2014 22nd International Conference on Pattern Recognition. IEEE; 2014, p. 3981–3986.

[40] Khatkar, J, Clemon, L, Fitch, R, Mettu, R. A reeb graph approach for faster 3d printing. In: 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE). IEEE; 2022, p. 277–282.

[41] Tuteski, O, Kočov, A. Mold design and production by using additive manufacturing (am)–present status and future perspectives. Industry 40 2018;3(2):82–85.