BLOWING BUBBLES FOR THE MULTI-SCALE ANALYSIS AND DECOMPOSITION OF TRIANGLE-MESHES

M. MORTARA⋆, G. PATANÉ⋆, M. SPAGNUOLO⋆, B. FALCIDIENO⋆, J. ROSSIGNAC♮

ABSTRACT. Tools for the automatic decomposition of a surface into shape features will facilitate the editing, matching, texturing, morphing, compression, and simplification of 3D shapes. Different features, such as flats, limbs, tips, pits, and various blending shapes that transition between them may be characterized in terms of local curvature and other differential properties of the surface or in terms of a global skeletal organization of the volume it encloses. Unfortunately, both solutions are extremely sensitive to small perturbations in the surface smoothness and to quantization effects when they operate on triangulated surfaces. Thus, we propose a multi-resolution approach, which not only estimates the curvature of a vertex over neighborhoods of variable size, but also takes into account the topology of the surface in that neighborhood. Our approach is based on blowing a spherical bubble at each vertex and studying how the intersection of that bubble with the surface evolves. We describe an efficient approach for computing these characteristics for a sampled set of bubble radii and for using them to identify features, based on easily formulated filters, that may capture the needs of a particular application.

keywords: Shape description, shape decomposition, multi-scale shape feature extraction, shape indexing.

1. INTRODUCTION

Shape analysis and coding are challenging problems in Computer Vision and Graphics. An ideal shape description should be able to capture and compute the main features of a given shape and organize them into an abstract representation which can be used to automate processes such as matching, retrieval or comparison of shapes. We have tackled the problem in the context of 3D objects represented by triangular meshes, having in mind that a good shape description should be able to distinguish between global and local features and should be based on geometric properties of the shape which are invariant under rotation, translation and scaling [6]. To characterize a shape we have used the paradigm of *blowing bubbles*: a set of spheres of increasing radius R_i, $i = 1, \dots, n$ is drawn, whose centers are at each vertex of the mesh, and whose radius represents the scale at which the shape is analyzed. The number of connected components of the intersection curve between each bubble and the surface gives a first qualitative characterization of the shape in a 3D neighborhood of each vertex. Then, the evolution of the ratio of the length of these components to the radius of the spheres is used to refine the classification and detect specific features such as sharp protrusions or wells, mounts or dips, blends or branching parts. For example, for a thin limb, that intersection will start simply connected and will rapidly split into two components. For a point on the tip of a limb, that intersection will usually simply remain connected, but the ratio of its length to the radius of the bubble will be decreasing.

[1]⋆Istituto di Matematica Applicata e Tecnologie Informatiche, Consiglio Nazionale delle Ricerche, Genova, Italia.
email: {falcidieno,michela,patane,spagnuolo}@ima.ge.cnr.it
♮ College of Computing and GVU Center, Georgia Institute of Technology, Atlanta.
email: jarek@cc.gatech.edu

For a point on a blend, that ratio will exceed 2π. An example of the resulting decomposition is given in Figure 1.

FIGURE 1. Shape decomposition using blowing bubbles.

The achieved description provides an insight on the presence of features together with their morphological type, persistence at scale variation, amplitude and/or size. The decomposition is algorithmically simple, independent on the orientation of the object in space and equally distributed in all directions. The multi-scale approach and the chosen descriptors define a view-independent decomposition and reduce the influence of noise on the shape evaluation. The number of radii and the interval given by the minimum and maximum radius define the scope of the description for the input surface and the step between consecutive scales. The choice of these parameters determines the final decomposition which is improved if a-priori knowledge about the size of the features to be extracted is available. In this paper we focus on the method adopted for the segmentation, while a possible application of the results can be found in [13], where a skeleton describing a shape from the point of view of its sharpest protrusions is presented.

The paper is organized as follows: in Section 2 previous work relevant for the described method is briefly reviewed. Basic concepts on differential geometry delineate, in Section 3, the theoretical background of the used geometric descriptors. The approach to shape classification is presented in Section 4. The algorithm description and details of the method are described in Section 5, while the mesh decomposition strategy is analyzed in Section 6. Finally, Section 7 includes critical considerations and remarks.

2. PREVIOUS WORK

An abstract description of a shape usually combines a set of primitives, that are relevant to the specific context, and is defined in terms of their type and intrinsic shape parameters.

As suggested in [12], methods for shape description can be classified into two broad categories: those considering only the local properties of the boundary of the shape, and those measuring properties of the enclosed volume. Typically, boundary-based methods evaluate accurate and mathematically well-defined local characteristics, such as critical points or curvature. They may also identify specific loci on the surface, such as curvature extrema or ridges, but they generally lack in providing a global view of the shape. Furthermore, they typically work at a single resolution and thus do not organize features into a hierarchy of global and local details.

Conversely, interior-based methods, which assume that the surface is the boundary of a solid, generally provide descriptions which better highlight the global structure of the shape. Skeletons, such as the medial axis or the Reeb graph [1, 16] belong to this class of descriptors. The great advantage of skeletons is that they provide an abstract representation by idealized lines that retain the connectivity of the original shape, thus reducing the complexity of the representation. Usually, each arc is associated with a portion of the original shape that corresponds to a feature. For example, in 2D the medial axis is constructed using the paradigm of the maximal enclosed disks, whose centers define a locus of points which describes, together with the associated radius, the width variation of the shape. The medial axis induces a decomposition of the shape into protrusion-like features, while concavities of the shape are not directly identified by the medial axis of the interior. Unfortunately, the medial axis of a 3D shape is not any longer a one-dimensional graph, but it is made of surface pieces as well. Moreover, the instability of the medial axis with respect to noise has prevented its use in many application areas. Approaches to construct and store the medial axis at different scales have been also proposed, which implicitly address the problem of noise reduction as well [16, 3, 15].

Another notable example of topology-driven skeleton is given by the Reeb graph [18, 1]. The Reeb graph is a topological structure which codes a given shape by storing the evolution of criticalities of a mapping function defined on the boundary surface. In particular, when the height function with respect to a predefined direction is chosen, the Reeb graph describes the evolution of the contours obtained by intersecting the shape with constant planes. The decomposition induced by the Reeb graph corresponds to a segmentation of the solid into slices and the corresponding branches of the Reeb graph identify the connected components of the surface. The description obtained using a Reeb graph approach is suitable for matching purposes especially if the mapping function is chosen in order to provide invariance under affine transformations. Such orientation-independent approaches have been proposed in [10, 13]; however, they are computationally intensive and offer little control over the scale at which the shape is analyzed. We propose here an alternative and more efficient approach that gives us more flexibility to formulate the filters for shape analysis, and captures the more representative properties in a more detailed description.

3. THEORETICAL BACKGROUND

This section provides definitions and concepts [4, 8, 11, 14] useful for describing our approach. Let $x : D \subseteq \mathbf{R}^2 \mapsto \mathbf{R}^3$ be a C^2-parameterization of the surface

$$\Sigma := \{x(u,v) : (u,v) \in D\}.$$

The classification of local properties of Σ is traditionally based on the study of the *mean* and the *Gaussian* curvature, which can be respectively defined as the average and the product of the maximum and minimum principal curvatures [11].

Let us consider the normal n to the surface Σ at a point p, and the *normal sections* of the surface around the normal vector, that is, the set of curves originated by intersecting the surface with planes containing the normal n. For each of these planar curves the curvature is classically defined as the inverse of curvature radius. If we call κ_1 the maximum curvature of the normal sections, and κ_2 the minimum, then the mean curvature $\overline{\kappa}$ is defined as $\overline{\kappa} := (\kappa_1 + \kappa_2)/2$ and the Gaussian curvature as $K := \kappa_1\kappa_2$. The directions along which the extrema of curvature are assumed are called principal directions. This definition formalizes the relation between the surface shape and its position with respect to the tangent plane. For example, for elliptic-shaped surfaces, the centers of curvature of all the normal sections will lie on the same side of the tangent plane, with positive values for the minimum and maximum of curvature. For hyperbolic-shaped surfaces, the centers of curvature will move from one side of the surface to the other, with a negative minimum value and a positive maximum value assumed at opposite sides with respect to the tangent plane. Finally, for parabolic-shaped surfaces, one of the principal directions will have curvature equal to zero, that is, along that direction the normal section will be a straight line. This is the case, in general, of ruled surface which are also said to have no double curvature. The planar case is obvious.

The Gaussian curvature represents a measurement at any point p of Σ which is the excess per unit area of a small patch of the surface, i.e., how *curved* it is. An interesting result is due to the Gauss-Bonnet theorem, which is introduced as follows. First of all, given a closed curve γ on a surface Σ, let T_γ be the total turning that the unit tangent t undergoes when it is carried along γ, defined as the sum of the local turnings, i.e. *exterior angles* [11] (see Figure 2). Then, the quantity $I_\gamma = 2\pi - T_\gamma$ is called the *angle excess* of the curve γ and it is related to the curvature of Σ within γ, as described by the Gauss-Bonnet formula.

Gauss-Bonnet Formula 1. *Let γ be a curvilinear polygon of class C^2 on a surface patch of class C^k, $k \geq 3$. Suppose γ has a positive orientation and its interior on the patch is simply connected. Then*

$$\int_\gamma \kappa_g ds + \iint_\Omega K dS = 2\pi - \sum_i \alpha_i = I_\gamma \tag{1}$$

where κ_g is the geodesic curvature along γ, Ω is the union of γ and its interior, K is the Gaussian curvature, α_i the exterior angles of γ, ds and dS are the curve and line elements respectively.

Among the properties of the angle excess the following ones have a particular interest for our approach:

- I_γ is independent of the chosen starting point on γ,
- I_γ is additive,
- for any topological disk on an arbitrary surface at p, the angle excess around the boundary is equal to the total curvature of the interior.

Starting from the Gauss-Bonnet formula and defining the total curvature of Σ as the integral:

$$T_\Sigma := \iint_\Sigma K dS \tag{2}$$

it can be proved that the last one is a topological invariant of compact, orientable surfaces as described by the following theorem.

Gauss-Bonnet Theorem 1. *If Σ is an orientable, compact surface of class C^3, then*

$$\iint_\Sigma K dS = 2\pi\chi(\Sigma) \tag{3}$$

where $\chi(\Sigma)$ is the Euler characteristic of Σ.

The definition of the curvature at each point of a triangulation is not trivial because a triangular mesh is parameterized by a piecewise continuous function whose second derivatives are, almost everywhere, null. More precisely, the curvature on a triangulation is concentrated along edges and at vertices, since every other point has a neighborhood homeomorphic to a planar Euclidean domain whose Gaussian curvature is null.

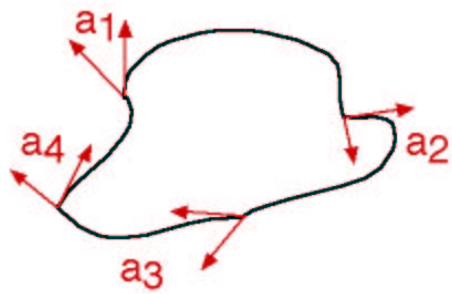


FIGURE 2. Exterior angles $\{a_i\}$ on a closed path γ.

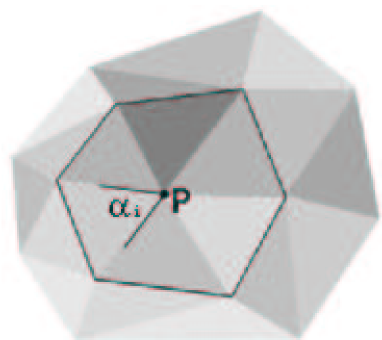


FIGURE 3. $Star(p)$.

The angle excess can be used to evaluate the Gaussian curvature at mesh vertices [5, 17]. Let us consider the region $Star(p)$ on the surface defined by the triangles incident in a vertex p (see Figure 3). The boundary of $Star(p)$ defines a closed path on the mesh, to which we may apply the Gauss-Bonnet formula (1). Since the geodesic curvature along the boundary is obviously zero (edges are straight), the total curvature at p is simply quantified by the sum of the exterior angles. To understand better the geometry of the situation, we can imagine to locally cut $Star(p)$ along any of the edges incident in p, and to develop the $Star(p)$ onto the plane without shrinking the surface. The sum of the exterior angles corresponds to the sum of the angles at p in the $Star(p)$. This result is consistent with the intrinsic nature of the Gaussian curvature since the angle excess only depends on the angles, that is, this value does not change if the mesh is deformed preserving the distance between points. Also, the computation of the angle excess can be performed without resorting to any coordinate system, as the angles may be obtained using only the edge length and not the vertex coordinates.

The methods proposed in the literature for curvature evaluation can be classified in different ways but a global comparison among them is still lacking as underlined in [5, 7, 20]. These methods can be divided into two main groups: *continuity-based* and *property-based* algorithms.

The first ones are developed transforming the discrete case to the continuous one by using a local fitting of the surface which enables to apply standard definitions. For example in [9] an approximation is derived at each vertex by applying the continuous definition to a least-square paraboloid fitting its neighboring vertices, while in [19] it is evaluated by estimating its tensor curvature. The second class of algorithms defines equivalent descriptors starting from basic properties of continuous operators but directly applied to the discrete settings. The methods proposed in [2, 17] are based on the Laplace-Beltrami operator, the Gauss map and the Gauss-Bonnet theorem guaranteeing the validity of differential properties such as area minimization and mean curvature flow [8]. The discrete Gaussian curvature at a vertex p of the mesh is evaluated as

$$K_G = \frac{2\pi - \sum_{i=1}^{num.faces} \alpha_i}{A} \tag{4}$$

, that is, the local angle excess in p weighted by the area A of a small patch of surface around p given by $Star(p)$ or some subset of it (see Figure 3).

In spite of the introduction of a multi-resolution structure, the mentioned approaches are usually sensitive to noise and small undulations, requiring smoothness conditions on the input mesh. Furthermore, the smoothing process used to get stable and uniform curvature estimations introduces a deficiency in the magnitude evaluation and, consequently, difficulties in the accurate distinction between planar patches and curved surfaces with low curvature. The local dependency of the curvature estimation (4) with respect to the $Star(p)$ of each vertex is shown in Figure 4.

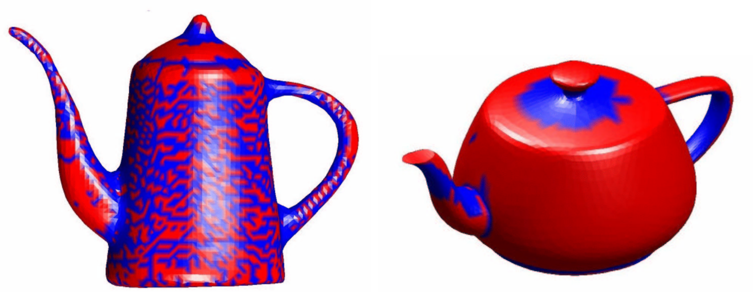
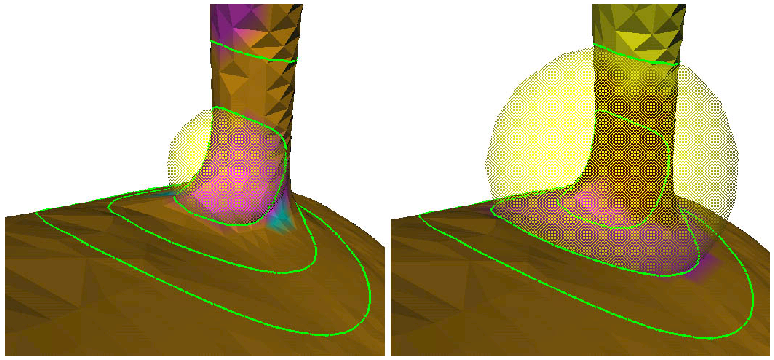


FIGURE 4. Gaussian curvature and sensibility to local noise; red and blue vertices represent elliptic and hyperboloic points.

4. GEOMETRIC AND TOPOLOGICAL CLASSIFICATION

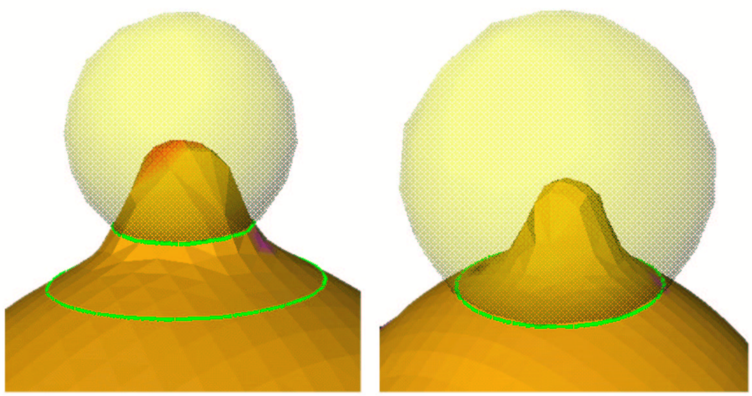
The approach proposed here for describing a 3D shape integrates boundary and interior information of the shape finalized at defining a complete multi-scale vertex classification. The link between closed paths and curvature has suggested to specialize its study to the family of closed paths built by intersecting the surface with spheres centered in each of its points. The study of the evolution of these curves and the geometric characterization of the mesh areas intersected by the spheres are the core of the proposed method. The topology of the intersection curves changes according to the object shape: in Figure 5(a), the highlighted sphere intersects the surface only at one curve, while in (b) the boundary of the intersection area splits into two connected components. This is likely to happen, for example, near handles and branches, or around deep pits. Therefore, the variation in the boundary suggests that the vertex is in the proximity of a feature, which becomes relevant at the scale, or radius, at which the change occurs.



(a) (b)

FIGURE 5. The evolution of intersection for increasing radii.

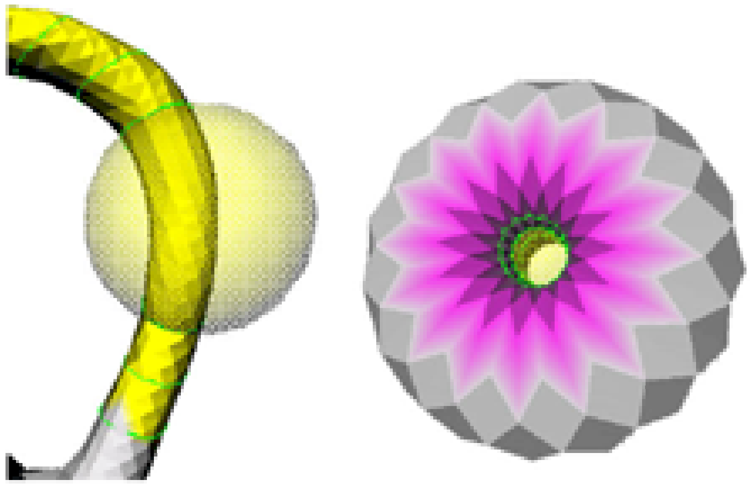
Given a set of radii $R_i, i = 1, \ldots, n$, each vertex of the mesh will eventually be classified with a n-dimensional vector of morphological labels, each corresponding to its type at the related scale. Shape features of the mesh are then identified by connected regions of vertices with the same label at a given scale, and the geometric parameters computed to assign the label will characterize the feature. For example, a tip and a mount are both characterized by one intersection curve, but they can be distinguished measuring the curvature induced by the intersection curve on the surface (see Figure 6).



(a) (b)

FIGURE 6. A tip measured for a small radius (a) becomes a mount at a larger radius (b).

Features which are identified by two intersection curves are further characterized by measuring the relative curve length and by checking if they define a volume which is inside or outside the shape (see Figure 7). These parameters, together with the persistence of type through the scale values, can be used to distinguish global and local features with respect to the scale range.



(a) (b)

FIGURE 7. A handle (a) is distinguished by a narrow pit or through hole (b).

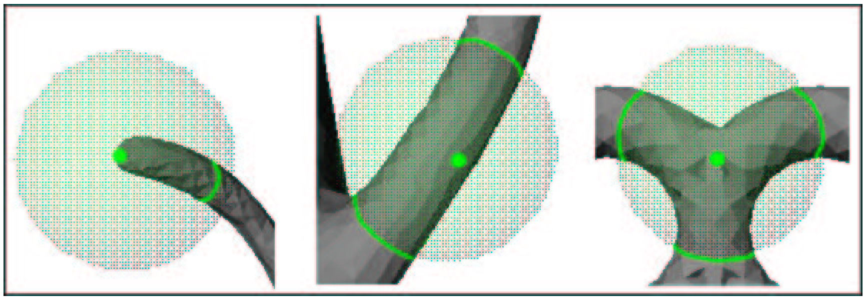
With reference to the properties of the obtained intersection curve $\gamma(p, R_i)$ at scale R_i, p is classified according to the number of connected components of $\gamma(p, R_i)$, curvature if $\gamma(p, R_i)$ has only one component, relative length if $\gamma(p, R_i)$ has two components, and a concavity/convexity check in all cases. These classification criteria will be treated separately in the following sections. The combination of these classifications leads to a complete characterization of vertices, which expresses both geometric and morphological properties of the surface. As far as this paper is concerned, only the decomposition is fully described with less emphasis on the construction of the region adjacency graph which encodes the segmentation. In the following, the main steps of the classification procedure are detailed. Then, the descriptors used to refine each class are introduced. We distinguish between geometric descriptors, which are the surface curvature and the relative length of the intersection curves, and the so-called *status* descriptors which distinguish between concave/convex or empty/full features.

4.1. **Classification based on intersections.** Given a 3D mesh Σ and a set of radii $R_i, i = 1, \ldots, n$, let $S(p, R_i)$ be the sphere of radius R_i and center p, and $\gamma(p, R_i)$ the boundary of the region of Σ containing p delimited by the intersection curves between the mesh and $S(p, R_i)$. Other regions of intersection might occur, but only the one containing p is taken into account. The first morphological characterization of the surface at a vertex p at scale R_i is given by the number of connected components of $\gamma(p, R_i)$.

We consider the following cases:

- 1 *component*: the surface around p can be considered topologically equivalent to a plane (see Figure 8(a)),
- 2 *components*: the surface around p is tubular-shaped (see Figure 8(b)),
- *3 or more components*: in a neighborhood of p a branching of the surface occurs (see Figure 8(c)).

In topological terms, two components identify a handle in the object, three or more components highlight a split. If $\gamma(p, R_i)$ is made by one component, the angles excess is computed and the vertex is classified as sharp, smooth or blend (see Section 4.2) . If $\gamma(p, R_i)$ is made by two components, their lengths are used to distinguish between conic and cylindrical shapes. For branching parts, no further geometric parameters are computed (see Table 1).



(a) (b) (c)

FIGURE 8. Different number of connected components in the intersection boundary.

To run the process, a set of radii must be selected for the computation of the intersections. The maximum and minimum radii (R_{max} and R_{min} respectively) determine an interval which is uniformly sampled according to the number of radii the user wants to use. This step produces the values of $R_i, i = 2, \ldots, n-1$ ($R_1 := R_{min}$ and $R_n := R_{max}$). Both R_{min} and R_{max} can be defined by the user by means of a slider in the GUI; otherwise, they are automatically set proportionally to the size of the object. More precisely, R_{min} is the minimum edge length and R_{max} the half of the diagonal bounding box of the object.

4.2. **Curvature characterization.** As described in Section 3, when $\gamma(p, R_i)$ has only one boundary component, the curvature at a point p, at scale R_i, is the angle excess of $\gamma(p, R_i)$. Instead of using the angle excess, we use the length of $\gamma(p, R_i)$ divided by the radius R_i, i.e. $L_{\gamma(p,R_i)} = length(\gamma(p, R_i))/R_i$. Note that this value has the dimension of an angle and it always assumes a positive value. Since we want to characterize the curvature of a surface, vertices will be labelled as sharp, smooth, or blend points according to their approximated curvature values by establishing some thresholds on the interval $[0, +\infty)$. We can distinguish the following cases:

- *sharp vertices*: let us consider a cone surface, with spike point p and $\alpha \in (0, \pi/2]$ the half amplitude of the cone. Intersecting the cone with a sphere centered in p and with radius R_i generates a circular curve of length $2\pi R_i \sin(\alpha)$ with $L_\gamma(p, R_i) = 2\pi \sin(\alpha)$, which is an increasing function of $\alpha \in (0, \pi/2]$: the lower the value of α, the more the surface around p tends to a cone shaped point. Intuitively, we consider p a sharp vertex if $\alpha \le \pi/4$ and consequently the curvature threshold is set to $T_s = \sqrt{2}\pi$.
- *smooth/blend*: to distinguish between these two situations we observe that the surface is smooth in a neighborhood of a point if its curvature continuously decreases, becomes at first *flat* and then *blend*. Now consider the intersection between the sphere and a plane; in this trivial case the length of the intersection curve is equal to $2\pi R_i$ and $L_\gamma(p, R_i) = 2\pi$; it follows that the threshold which discriminates between *smooth* and *blend* is set to $T_b = 2\pi$.

Summarizing, the characterization of a point p at scale R_i is set as follows:

- $0 \leq L_\gamma(p, R_i) \leq \sqrt{2}\pi$: p is *sharp*,
- $\sqrt{2}\pi < L_\gamma(p, R_i) \leq 2\pi$: p is *smooth*,
- $L_\gamma(p, R_i) > 2\pi$: p is *blend*.

For example, see Figure 9.

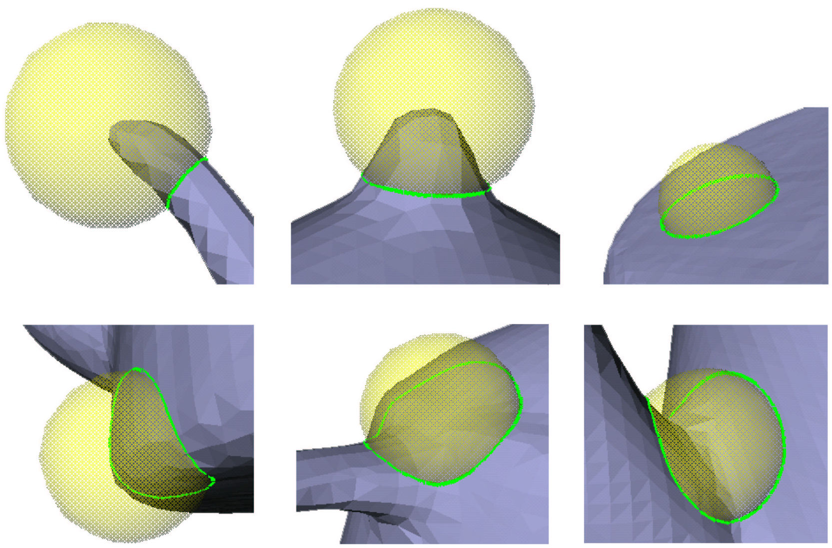
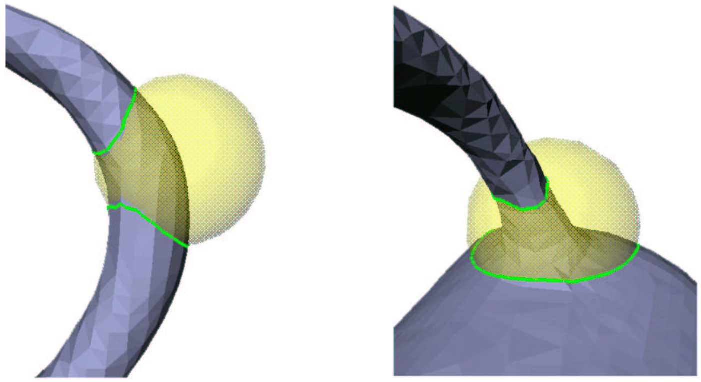


FIGURE 9. Several cases of one intersection curve: note the relation between the intersection curve lenght and the curvature of the surface in the neighborhood of the center of the sphere.

4.3. **Relative length characterization.** Now consider the case of two connected components in the intersection curve $\gamma(p, R_i)$. As mentioned above, this means that p lies on a region of the surface having an elongated shape, like a tubular protrusion or a handle around a hole in the object. We can specialize this remark as follows: if the length of the two intersection components is nearly the same, the shape at the scale R_i can be approximately considered cylindrical; if one is much longer than the other, it means that the shape may be seen as a conic part (see Figure 10). Let γ_1 and γ_2 be the two intersection components, and l_1, l_2 their lengths with $l_1 \geq l_2$. The shape is considered conical if $l_1 \geq 2l_2$, cylindrical otherwise. The related threshold is $T_c = 1/2$ thus guaranteeing that the amount l_2/l_1 (belonging to $[0, 1]$) uniquely determines whether the local shape of the surface around p is cylindrical or conic.

4.4. **Status characterization.** The extraction of morphological features on a surface is based on different operators each of them providing a specific, e.g. geometric, topological, approach to its description. For instance, in the case of one connected components in $\gamma(p, R_i)$, to discriminate between convex and concave vertices would lead to classify a *sharp* point as a peak or a pit, a *smooth* point as a mount or a dip. Obviously, the distinction between convex and concave does not make sense for blend points. For vertices with two or more connected components in $\gamma(p, R_i)$, it is checked if the surface intersected by the



(a) (b)

FIGURE 10. Example of conical (a) and cylindrical (b) parts of a triangular mesh.

sphere encloses a volume which is inside or outside the object. In other words, a criterion is used for distinguishing a handle from a deep tubular depression of the object.

Let us consider the case of one component first. As for curvature computation, concavity/convexity evaluation at a vertex p of a triangular mesh is strongly affected by noise and it depends on the local topology of the $Star(p)$. The local approach is depicted in Figure 11: a given edge e shared by triangles t_1, t_2 of a mesh is convex (resp. concave) if the angle formed by t_1, t_2, inside the object, is less (resp. more) than π.

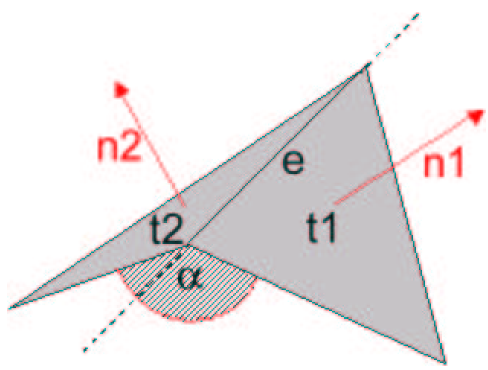


FIGURE 11. Edge concavity or convexity criterion.

Consequently, a given vertex p is defined strictly convex (resp. strictly concave) if all the incident edges in p are convex (concave). Because in most cases the incident edges in p are both convex and concave, the previous classification cannot be applied. Furthermore, point coordinates can be slightly affected by noise resulting in a complete different classification. For these reasons, the method adopted for assigning a *convex* or *concave* label to a vertex p at scale R_i again uses the intersection between the mesh and the sphere. In the case of one connected component of the intersection curve, the center of mass b of γ and the average normal N of the intersected triangles are computed. The vertex p is considered concave (convex) at scale R_i, if p lies below (resp. above) γ, that is $N \bullet (b - p) > 0$ (resp. < 0), where '$\bullet$' denotes the inner product. We refer to Figure 12 for an easier understanding.

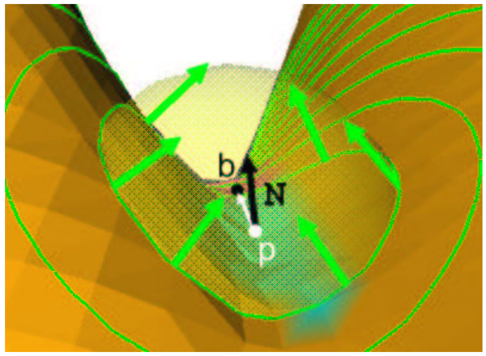


FIGURE 12. Configuration of the intersection curve normals around a concave point.

Suppose now that $\gamma(p, R_i)$ has two intersection components. Again we can distinguish between the case in which the local shape is a tubular protrusion or a tubular well of the surface (see Figure 13), in analogy with the property of convex/concave mentioned above for points generating one intersection curve.

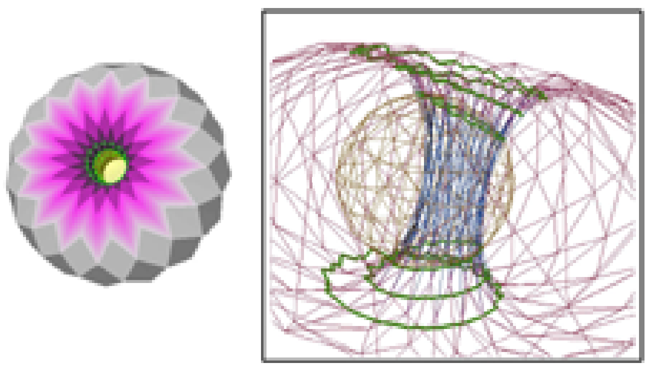
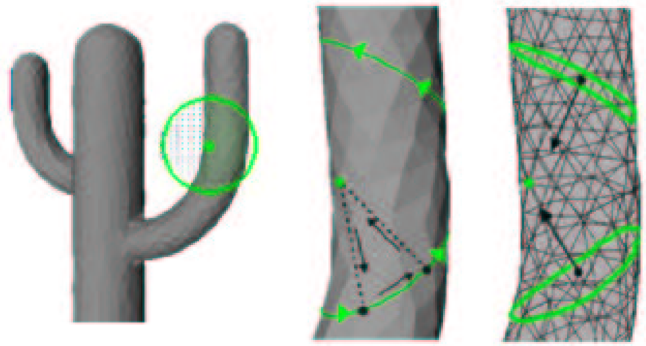


FIGURE 13. Example of a tubular through hole on a triangular mesh.

If the number of connected components of the intersection curve are two as in Figure 14(a), we consider the orientation of each component of $\gamma(p, R_i)$ as naturally induced by the triangle orientation (see Figure 14(b)). It happens that if p lies on a tubular protrusion of the surface, the normal vector of the average plane related to each connected component of $\gamma(p, R_i)$ is directed towards p (see Figure 14(c)) according to the right-hand rule; if p appears on a tubular depression of the object, the vectors have opposite directions. This statement holds for three or more connected components too, thus it is possible to discriminate between a branch on the outer surface or a splitting cavity.



(a) (b) (c)

FIGURE 14. (a), (b) Curve orientation derived by triangle orientation, (c) average normal of intersecting curves.

5. ALGORITHM AND IMPLEMENTATION DETAILS

This section deals with the implementation details of the algorithm used for the sphere-mesh intersection which is the core of the shape characterization. Throughout the discussion we refer to the pseudo-code given at the end of the section.

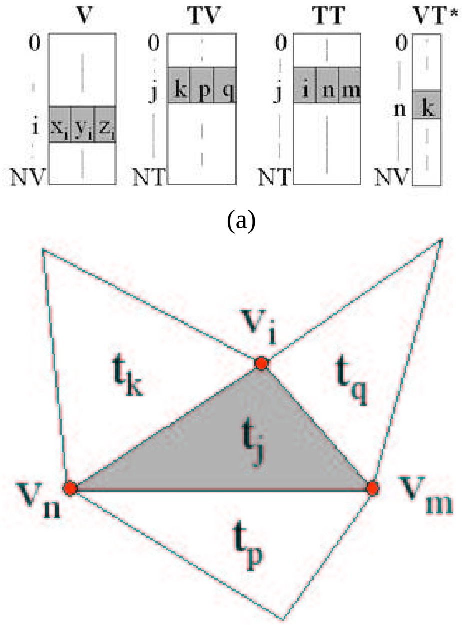
The mesh is encoded by mean of a triangle-based data structure, which stores:

- for each triangle, its three vertices and its three adjacent triangles, which represent the Triangle-Vertex and Triangle-Triangle relations, respectively
- for each vertex, its coordinates and one (arbitrary) of its incident triangles.

Actually the Vertex-Triangle relation (VT) associating to a vertex all its incident triangles is necessary to navigate the mesh, but the memory space required can be strongly optimised by coding just one of those triangles per vertex (partial Vertex-Triangle relation or VT^*). The total VT relation can then be retrieved in linear time by iteratively apply the TT relation starting from the stored triangle. A scheme of the data structure is given in Figure 15. The storage of this data structure requires $3n_V * sizeof(float) + 6 * n_T * sizeof(int) + n_V * sizeof(int)$, since vertices and triangles occur in the TT, TV and VT^* structures as integers.

For each vertex v the computation of the intersection curves between the mesh and a set of n spheres centered in v with increasing radii $R_1, \ldots, R_n$ is computed as follows:

- one of the triangles incident in v is inserted in the queue Q (this operation takes constant time if we have the $VT^\star$ data structure) and it is marked as visited;
- a triangle t is extracted from Q and the main loop is repeated until Q is empty;
- the algorithm checks if the spheres intersect t: for each radius R_i, if at least two vertices p, q of t, distinct from v, satisfy the conditions $\|p - v\|_2 \leq R_i$, $\|q - v\|_2 \geq R_i$, t is intersected by the sphere $S(v, R_i)$. This operation takes constant time. In this case, t is considered as a seed triangle for tracing the whole line of intersection whose continuation is searched in the triangles adjacent to t. The function *intersection* is thus invoked to complete the intersection curve starting from t and moving on its neighbours intersected by $S(v, R_i)$. The curve $t \cap S(v, R_i)$ is calculated considering the intersection points between the sphere and each edge of t. More precisely, given an edge $[a, b]$ of t, we can parameterize



(b)

FIGURE 15. Data structure organization; the information for the vertex v_i and the triangle t_j is highlighted as shown in (b).

it as $u(s) := sa + (1-s)b$, $0 \leq s \leq 1$; the intersection points, if any, are located as $u(s_0)$, $u(s_1)$ with s_0, s_1 solutions in $[0,1]$ of the equation $\|u(s) - v\|_2^2 = R^2$ of degree two in the unknown s. As shown in Figure 16, the length of $\gamma(v, R_i)$, is given by the sum of the composing arc lengths, each one belonging to an intersecting triangle, and given by $R_i\theta$ where θ is the angle $\widehat{apb}$, a, b being the intersection points. The call of this function increases the number of connected components of the intersection line for a given radius. Moreover, the neighbours of the intersection triangles traversed but not marked, that is, those which lay outside $S(v, R_i)$ but inside $S(v, R_n)$, are inserted in Q;

- If t is not an intersection triangle and it lies inside the sphere of maximum radius $S(v, R_n)$, its neighbor triangles (if not marked) are inserted in Q; otherwise, it is simply discarded.

The construction of a connected component of the intersection line may take as many constant operations as the number of intersection triangles, i.e. $O(n_T)$ in the worst case. However, in this implementation each triangle is visited only once: marking triangles when they are inserted in Q avoids to consider them more than once, and the intersection triangles traversed during the execution of *intersection* are not stored in Q. Therefore, the main loop takes $O(n_T)$, that is $O(n_V)$; doing this operation on the whole mesh takes $O(n_V^2)$ time. Note that if the step between the radii is low with respect to the average edge length, a triangle can be easily intersected by more than one sphere, and the function *intersection* could be invoked on the same triangle as many times as the number of radii. Anyway, once the number n of radii is chosen this is a constant value, so that the loop (L) does not increase the complexity.

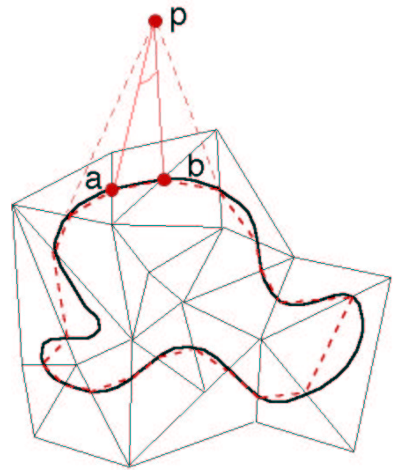


FIGURE 16. Approximated length of the intersection paths.

main(){
$Q = \emptyset$;
(num. connected comp. for R_i)=0;
for all $v \in V$ {
    $Q \longleftarrow TV^{\star}(v)$;
    // main loop
    while($Q \neq \emptyset$) {
        t = first element removed from Q;
        (L) **for** ($R_i = R_1, \ldots, R_n$)
            if ($\exists\, v_l, v_m \in TV(t)\ :\ \|v - v_l\|_2 \leq R_i$
                & $\|v - v_m\|_2 \geq R_i$){
                intersection(t, R_i,v);
                ($\#$ connec. comp. for R_i) ++ ;}
    }

for all $t_i \in TT(t)$
    if(t_i is not marked & ($\exists v_l \in TV(t_i)$: $\|v - v_l\|_2 \leq R_n$)
        $Q \longleftarrow t_i$;
}
intersection(t,R_i,v) {
L_i= *intersectionLength*(t,R_i,v);
$t_{next} = t$;
do {
    $t_{next} = t_i \in TT[t_{next}] : t_i \cap S(v, R_i) = \emptyset$
    L_i += *intersectionLength*(t_{next},R_i);
    if(t_i is not marked & ($\exists v_l \in TV(j,i) : \|v - v_l\| \leq R_{max}$))
        $Q \longleftarrow t_i$;
        } **while**($t_{next} \neq t$);
}

TABLE 1. Morphological feature characterization.

Label	Feature	Color	#∩	Geometric	Status
T	TIP	red	1	$L/R_i \leq T_s$	convex
P	PIT	blue	1	$L/Ri \leq T_s$	concave
M	MOUNT	orange	1	$T_s < L/Ri \leq T_b$	convex
D	DIP	cyan	1	$T_s < L/R_i \leq T_b$	concave
B	BLEND	pink	1	$L/R_i > T_b$	–
L	LIMB	yellow	2	$L_2/L_1 \geq T_c$	full
W	WELL	violet	2	$L_2/L_1 \geq T_c$	empty
J	JOINT	brown	2	$L_2/L_1 < T_c$	full
F	FUNNEL	gray	2	$L_2/L_1 < T_c$	empty
S	SPLIT	green	≥ 3	–	full
H	HOLLOW-Y	black	≥ 3	–	empty

6. MESH DECOMPOSITION

The focus of this section is the integration of the different characterizations, described in Section 4 to achieve a unique segmentation of the input mesh into morphological features represented by closed regions with uniform properties. In Table 1, a summary of the labels assigned to vertices, for a given scale R_i, is showed. According to this classification, each vertex is assigned a label; then, with a region-growing procedure the input mesh is decomposed into patches which correspond to shape features relevant at scale R_i.

The morphological classification associates a vector of feature labels to each vertex, and each label describes the vertex at the corresponding scale. Selecting the scale of interest, the surface can be rendered using a color-coding of the feature labels. The achieved decomposition is an affine-invariant segmentation into disjoint, non-empty subsets which code the geometry and shape evolution through scale changes. In Figure 17 an example is shown; the different views display the mesh decomposition at different scales, and the colors are those related to Table 1. For each scale, a reference sphere is also drawn.

The tools defined above permit the analysis of a shape at different scales, and to derive information about the persistence of a shape feature across the scale range. It is also possible to define a basic query language which allows to extract features defined by relations between morphological labels at different scales. The combination of these relations using logical operators enables the construction of a high-level language for shape interrogation guaranteeing a multi-task model. In fact, the user is able to extract a single shape element using a single query, to combine them and, in future improvements, to locally modify the geometry, by using other surface patches, or modifying the topology by changing the structure of the adjacency graph.

Currently, a coarse feature-based query language is available, which allows the user to submit a query like "which are the vertices whose feature type is TIP at scale R_3 and MOUNT at scale R_6?". To this purpose, a query vector with wild card v_q, where $v_q[i]$ specifies the requested feature type at scale R_i, and the AND/OR Boolean operators are used. For instance, suppose we used a set of ten radii; the vector [*,*,T,*,*,M,*,*,*,*] with the AND operator specify the previous query. Here, feature labels are those in Table 1, and the symbol '*' means that, at that scale, each feature type is allowed.

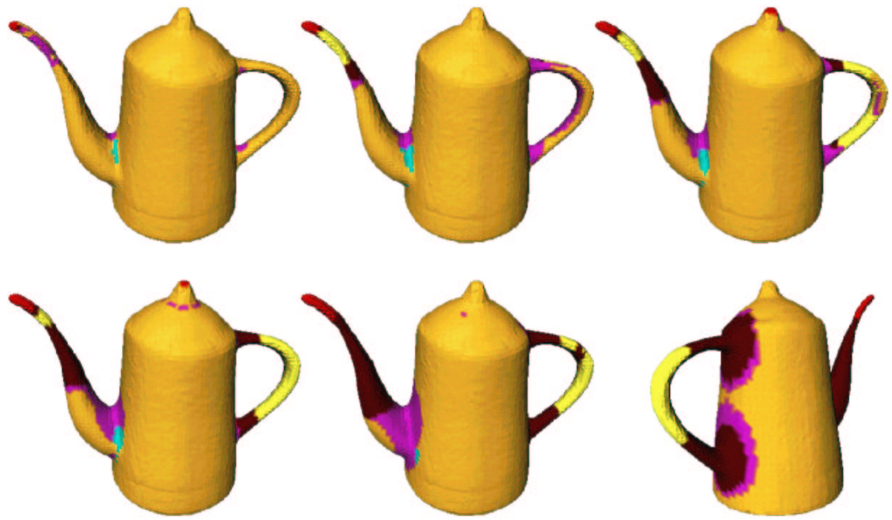


FIGURE 17. Shape segmentation on the pot at different scales.

The results obtained (see Figure 18) suggest that further improvements of the query language will allow the extraction of higher-level features, like handles or main body of a given object. Tubular components can be extracted choosing LIMB OR JOINT vertices; among them, handles correspond to cycles in the region adjacency graph, and protrusions are adjacent to TIP or MOUNT zones. Selecting points which assume TIP OR MOUNT OR DIP OR BLEND features at most levels of detail identify the main body of the object, and so on. Using the described language, a mesh can be analyzed in a rather flexible way.

Instead, if we are willing to extract a global shape classification which takes into account the whole range of scales into a single decomposition of the mesh, the following voting algorithm for persistence can be used. First, the points are classified according to the intersection connectivity, that is, according to the number of single, double or multiple components in the intersection boundary, considering the whole range of scales. The mesh is therefore segmented in parts which are characterized by having either almost always one intersection, or two or more. This step provides a first insight on features which are persistently protrusion-like, handle-like or branch-like, without distinguishing if they are convex/concave or full/empty. In Figure 19(a), the result of this segmentation is shown, where the blue parts are composed by vertices having only one intersection for more than the 75% of scales, the red are those having two intersections for the same threshold. Finally the grey areas are those corresponding to shape transitions where both one and two intersections occur approximatively in the same percentage.

Within each of the resulting parts, a further classification can be done considering the related geometric characterization. For example, in Figure 19(b), the shape vertices are colored with different blue saturations depending on the curvature; a darker blue corresponds to a higher curvature. An analogous criterion is applied to vertices with two intersections using their relative length where a lighter red is related to cylindrical-like features. A threshold bigger than 75% can be selected for the persistency analysis in order to achieve

FIGURE 18. Queries with matched points depicted in red. The use of AND, OR operators among the scales is specified. The round parenthesis work as OR between feature at the same scale.

a stronger identification of patches with one or two intersections; this choice generally makes transition areas grow.

(a) (b)

FIGURE 19. (a) Persistence analysis based on the number of intersection curves, (b) the previous classification is refined by geometric information.

7. METHOD INSIGHTS

In this section several considerations on the algorithm are discussed comparing the multi-resolution description with that achieved with the persistency algorithm.

With the proposed strategy, the curvature is analyzed in a neighborhood whose size depends on R_i: for small values of R_i, such as the average length of the edges incident in p, the curvature approximation resembles the discrete curvature estimation proposed in [2] and suffers for the same problems while, for increasing values, it becomes more stable to noise. In Figure 20, the feature decompositions obtained on the rabbit and on the same data set with added noise are compared. The main features, like tips and limbs, are preserved: the influence of the thresholds involved is lessened by the persistence analysis. The multi-scale decomposition depends on the chosen set od radii, and if a too small radius leads to noise problems on the other hand, a too large radius can give a meaningless result. Small radii can be used to determine detail features while bigger ones are able to capture global characteristics of the surface. From these considerations, it follows that the choice of R_i is related to the scale of the features which have to be extracted, and the use of a set of increasing radii is suitable for performing a multi-scale analysis of the shape. The previous considerations have been tested by evaluating the multi-resolution curvature on a torus with two radii: R_i equal to the minimum edge of the input mesh and $2R_i$. Curvature values achieved with R_i corresponds to the theoretical point classification into parabolic, hyperbolic and elliptic regions on the torus (see Figure 21(a)). Increasing the radius, i.e. choosing $2R_i$, results in a similar classification where parabolic points, identified as the boundary between hyperbolic and elliptic ones, are shifted with respect to their theoretical position represented by the green line (see Figure 21(b), (c)) .

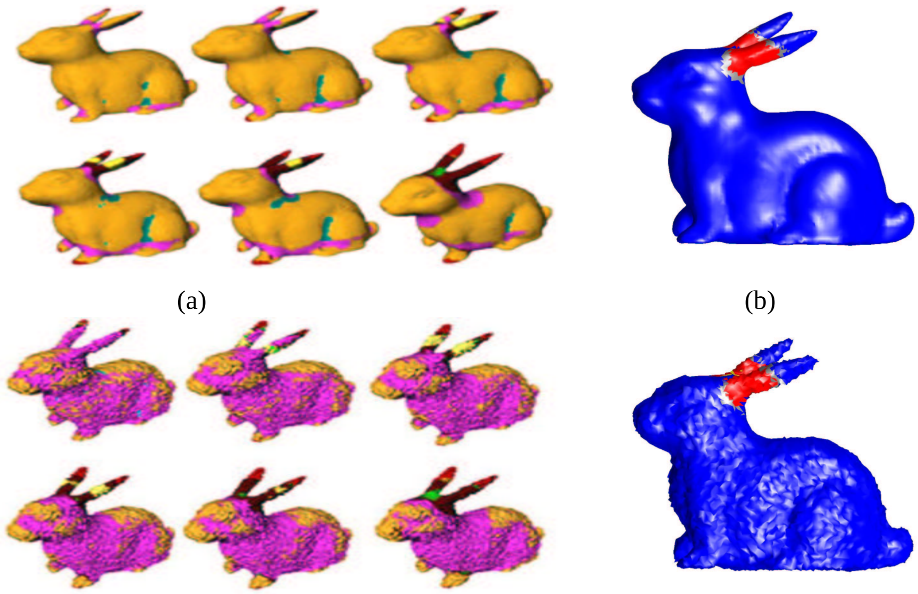
The multi-resolution curvature evaluation can be used for segmenting the input surface grouping those points which share a common value with respect to a given threshold ϵ and a radius R_i, i.e.

$$p, q \text{ belong to the same patch} \Longleftrightarrow |L_\gamma(p, R_i) - L_\gamma(q, R_i)| \leq \epsilon.$$

This approach is commonly applied by segmentation methods based on curvature. On the other hand, the refined segmentation into morphological features is meaningful in all those cases where the object admits a decomposition into these building patches. As underlined in Figure 22, detailed features are better recognized using simply the multi-resolution curvature evaluation, because in such case a decomposition into features, such as limbs and tips, is not meaningful.

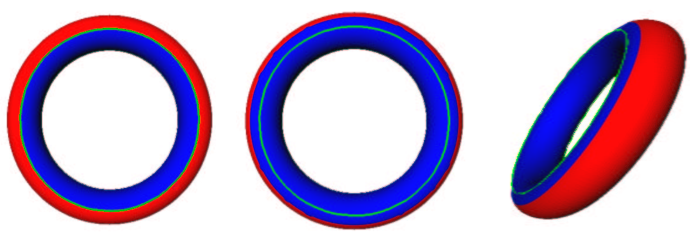
8. CONCLUSIONS AND FUTURE WORK

The evolution of intersection curves produced by blowing bubbles at mesh vertices has been proved to be a good approach to characterize a shape using meaningful shape features. Increasing the radius of the bubbles produces an easy and efficient multi-scale analysis of the shape, which can be effectively used to produce a set of specific and flexible tools for shape analysis. The resulting description is an affine-invariant segmentation of disjoint, non-empty subsets and equally distributed in all directions, which codes the geometry and shape evolution through scale changes. Finally, the decomposition is algorithmically affordable and coherent with respect to previous work on these topics. The extracted information has a more general usefulness which is suitable not only for segmentation but also for the definition of a shape abstraction tool and of an editing model for triangular meshes. For example, it can be used to find the seed points for the construction of an affine-invariant skeleton [13] which represents the input for a wide class of applications such as matching



(a) (b) (c) (d)

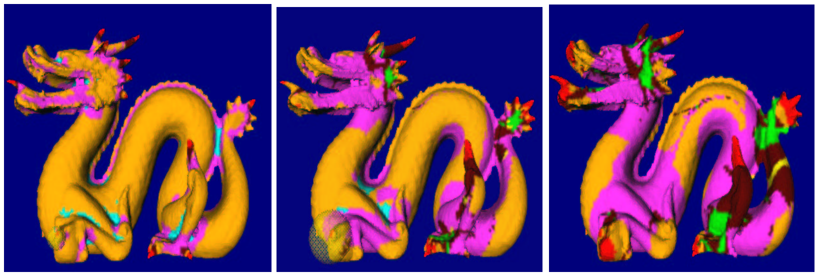
FIGURE 20. (a), (b) Shape segmentation and percistence analysis on the original rabbit, and (c), (d) on the model with added noise. Achieved segmentations based on persistence analysis are nearly identical.



(a) (b) (c)

FIGURE 21. (a) The point classification corresponding to R_i chosen as the minimum edge: red and blue vertices locate elliptic and hyperbolic points while the green line visualizes the theoretical parabolic line, (b), (c) the point classification with radius $2R_i$.

and topological characterization of 3D-shapes. Figure 23 shows all the steps of the framework. Future developments will mainly focus on the definition of a feature adjacency graph and on the study of its evolution within the scale range required.



(a) (b) (c)

FIGURE 22. Feature decomposition on the dragon at three different scales.

REFERENCES

[1] S. Biasotti, B. Falcidieno and M. Spagnuolo, "Extended Reeb Graphs for Surface Understanding and Description". *Proceedings of the* 9^{th} *Discrete Geometry for Computer Imagery Conference, LNCS*, Springer Verlag, Uppsala, 2000.

[2] M. Desbrun, M. Meyer, P. Schroeder, A.H. Barr, "Discrete Differential Operators for Triangulated 2-manifolds", in *VisMath 2002, Proceeedings*, "Visualization and Mathematics III", Springer Verlag, Berlin, 22-25 May 2002.

[3] Dill, A.R., Levine, M.D., Noble, P.B.: Multiple Resolution Skeletons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, no. 4, 495-503,1987.

[4] DoCarmo, M.: *Differential geometry of curves and surfaces*. New Jersey: Prentice Hall 1976.

[5] Falcidieno, B., Spagnuolo, M.: Geometric Reasoning for the Extraction of Surface Shape Properties. *Communicating with the Virtual World*, N.Magnenat Thalmann and D. Thalmann (Eds.), Springer-Verlag, (1993)

[6] Falcidieno, B., Spagnuolo, M.: A Shape-abstraction paradigm for modelling geometry and semantics. *Proceedings of Computer Graphics International 1998*, IEEE Computer Society Press (1998)

[7] Gonzales, R.C., Woods, R.E.: *Digital Image Processing*. Reading, MASS.: Addison-Weshley, 1992.

[8] V. Guillemin, and A. Pollack, *Differential Topology*, Englewood Cliffs, NJ: Prentice-Hall, 1974.

[9] B. Hamman: Curvature Approximation for Triangulated Surfaces, *Computing Suppl.* 8, 1993, pp. 139-153.

[10] M. Hilaga, Y. Shinagawa, T. Kohmura, T.L. Kunii, "Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes", in *Computer&Graphics, Proceeding of Siggraph 2001*, Los Angeles, 2001.

[11] Lipschutz, M.M.: *Theory and Problems of Differential Geometry*. Schaum's Outline Series.

[12] Loncaric, S.: A survey of shape analysis techniques. *Pattern Recognition*, 31(8), 983-1001, 1998.

[13] Mortara, M., Patané, G.: Affine-invariant skeleton of 3D-shapes. In *IEEE Proceeding of International Conference on Shape Modelling and Applications 2002*, Alberta, Canada, 245-252.

[14] Mortenson, M.E.: *Geometric Modeling*, John Wiley, 1985.

[15] Ogniewicz, R.L.: Skeleton-space: a multiscale shape description combining region and boundary information. *Proceedings of Computer Vision and Pattern Recognition*, 746-751,1994

[16] Pizer, S.M., Oliver, W.E., Bloomberg, S.H.: Hierarchical Shape Descripion Via the Multiresolution Symmetric Axis Transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-9, no. 4, 505-511,1987.

[17] Polthier, K., Schmies, M.: Straighhtest Geodesic polhedral Surfaces. In H.C. Hege and K. Polthier, editors, *Mathematical Visualization*. Springer Verlag, 1998.

[18] Y. Shinagawa, T.L. Kunii, and Y.L. Kergosien, "Surface Coding Based on Morse Theory", *IEEE Computer Graphics & Applications*, 1991, pp. 66-78.

[19] Taubin, G., Estimating the Tensor Curvature of a Surface from a Polyhedral Approximation, *Fifth International Conference on Computer Vision (ICCV'95)*.

[20] Trucco, E., Verri, A.: *Introductory to techniques for 3-D computer vision*. Prentice Hall, 1998.

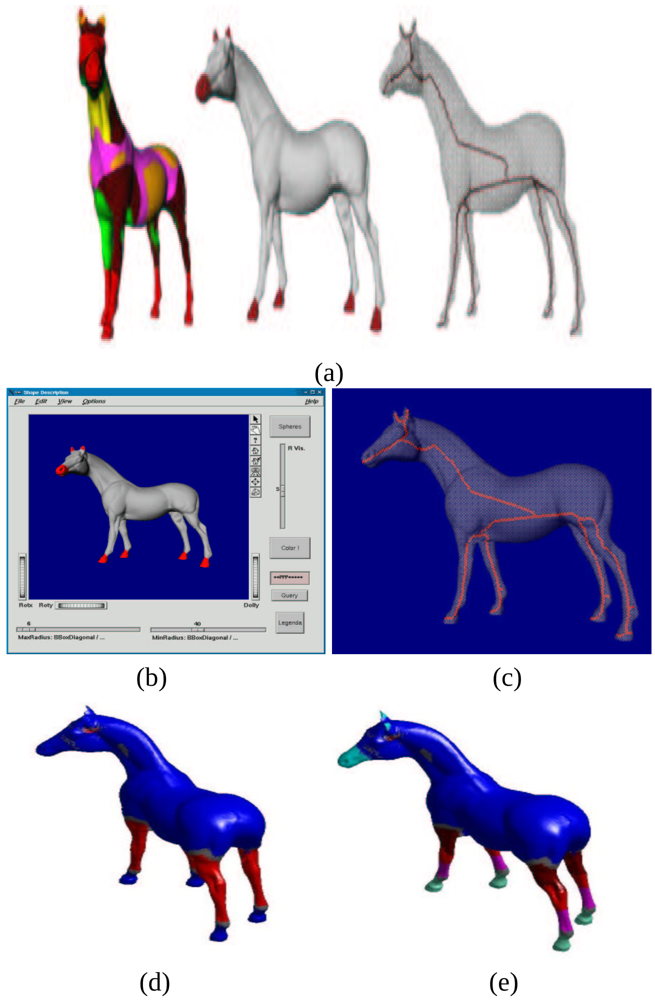


FIGURE 23. Global framework: (a) curvature estimation on the horse with different radii, (b) peak regions are extracted with a query, (c) regions selected in (b) are used as seed points for extracting the skeleton, (d), coarse persistence analysis, (e) refined persistence analysis.