Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization (2003)
G.-P. Bonneau, S. Hahmann, C. D. Hansen (Editors)

Detection of constrictions on closed polyhedral surfaces

F. Hétroy[1] and D. Attali[1]

[1] LIS Laboratory, INPG, Grenoble, France

Abstract

We define constrictions on a surface as simple closed geodesic curves, i.e. curves whose length is locally minimal. They can be of great interests in order to cut the surface in smaller parts. In this paper, we present a method to detect constrictions on closed triangulated surfaces. Our algorithm is based on a progressive approach. First, the surface is simplified by repeated edge collapses. The simplification continues until we detect an edge whose collapse would change the topology of the surface. It happens when three edges of the surface form a triangle that does not belong to the surface. The three edges define what we call a seed curve *and are used to initialize the search of a constriction. Secondly, the constriction is progressively constructed by incrementally refining the simplified surface until the initial surface is retrieved. At each step of this refinement process, the constriction is updated. Some experimental results are provided.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

1. Introduction

In computer graphics, data acquisition hardwares provide more and more fine and thus complex *surface meshes*. A surface mesh can be complex from a combinatorial point of view (the number of faces in the mesh), from a topological one (the number of holes, that is to say the genus of the surface), or from a geometrical one (different parts of the surface have very different local feature sizes). In order to better handle such surfaces, it can be very useful to decompose them in smaller parts.

Our goal in this paper is to decompose a polyhedral surface into several components, connected together by narrower parts. The first step is to characterize the parts of the surface where we want to cut the object. We define *constrictions* as locally shortest simple closed curves on a closed polyhedral surface. By definition, constrictions are closed geodesics. The goal of this paper is to detect constrictions on closed polyhedral surfaces. To do so, we start from the idea that when we simplify the mesh, constrictions correspond to areas where the surface is split into several connected components. To compute constrictions, we use a *progressive approach*. We first simplify the mesh by iteratively applying a well-known elementary mesh transform called the *edge collapse*. We stop the simplification process when we detect an edge whose collapse would change the topology of the surface. Equivalently, it means that three edges of the surface form a triangle that does not belong to the surface. This small loop of three edges is called a *seed curve*. Seed curves are used to initialize the construction of constrictions. The second step of the algorithm is to deduce from the coarse constriction detected on the simplified surface a fine constriction on the initial surface. For this, the simplified surface is progressively refined using the *vertex split operator*, which is the inverse process of the edge collapse, until the initial surface is retrieved. After each vertex split, the constriction is updated. The updates remain local. Indeed, one can observe experimentally that a constriction passes through a limited number of vertices on the surface, called *pivot vertices*. The update concerns only a portion of the curve between two successive pivot vertices.

The paper is organised as follows. In Section 2, we recall previous works about surface simplification and geodesics and shortest paths computation on polyhedral surfaces. We define constrictions on triangulated surfaces in Section 3, and give some basic properties about them. Our algorithm is explained in Section 4. Some results are shown and discussed in Section 5, before we conclude in Section 6.

2. Related works

2.1. Surface simplification

Surface simplification has been widely studied for more than ten years, as this field has become more and more important to reduce storage and improve transmission, computation and visualization of surface meshes. A very large number of techniques for surface simplification have been proposed, including re-tiling[20] or wavelet approximation[4]. The most common ones are perhaps the refinement methods and decimation methods. The first are bottom-up: starting from a very coarse approximation of the initial surface, we refine it by inserting new vertices and increasing the number of faces; we stop when the error between the initial surface and the current refined surface has reached a sufficiently small value. Error can be defined different ways: for example, as a geometrical error, or as a number of vertices. Decimation methods are top-down methods: we remove at each step an edge (or a vertex[18], or a face) and its neigbourhood from the mesh, and remesh the removed area. The choice of the removed element depends on heuristics. These methods usually produce fast and good results (i.e., simplified surfaces have far less faces than initial surfaces and the geometrical error between them is small). One of the most commonly used scheme in these methods is the *edge collapse*, also known as *edge contraction* or *edge decimation*. This local operator removes an edge from the mesh and merges its two endpoints (see Figure 1).

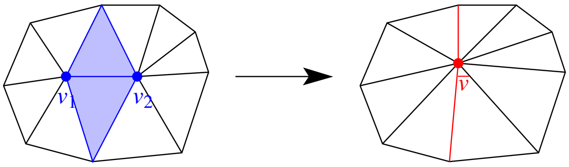


Figure 1: *Edge collapse: the edge v_1v_2 and the two adjacent faces are removed, the vertices v_1 and v_2 are merged in $\overline{v}$.*

We will simplify our surface using edge collapse, since it is a simple, efficient and widely used scheme. Methods using edge collapse are described in e.g. [5, 8, 11, 17]. Also note that several surveys about surface simplification methods exists, see for example [2, 7].

2.2. Geodesics and shortest paths

In order to detect constrictions, we will have to compute geodesics. A geodesic is a locally length-minimizing curve: between any two sufficiently close points on a geodesic, this geodesic follows a shortest path. Moreover, globally shortest paths are geodesics. The problem of computing the shortest path between two points in a polyhedral space has been studied for long in computational geometry [13, 19]. This is because this problem has many pratical applications, such as robot motion planning, shape analysis, or terrain navigation. One of the best known algorithms to compute exact shortest paths is Chen and Han's[3], which is a $O(n^2)$ algorithm. However, it can fail for non-convex polyhedra. Many methods have been proposed to approximate shortest paths on polyhedral surfaces (e.g. [1, 6, 9, 10, 12, 21]), some of them using Chen and Han's algorithm. Pham-Trong proposed in her thesis[15] an iterative method to compute an exact geodesic path between two points on a polyhedral surface by updating the current processed path around the vertices where it is not optimal. This method is particularly well adapted to our problem since our approach is progressive; this is why we have chosen to apply it in our case, as described in Section 4.

For a general survey about shortest paths, see [14].

3. Constrictions on triangulated surfaces

Let $\mathcal{P}$ be a 2-manifold embedded in $\mathbb{R}^3$. We define constrictions on $\mathcal{P}$ as locally length-minimizing simple, closed curves of $\mathcal{P}$. More formally, let $L(\alpha)$ denote the length of the curve α. $d(p,q)$ denotes the geodesic distance between two points p and q of $\mathcal{P}$, i.e. the infimum of the lengths of the paths from p to q on $\mathcal{P}$. The Hausdorff distance between two sets $\alpha,\beta \subset \mathcal{P}$ is

$$d_H(\alpha,\beta) = \max(\sup_{x\in\alpha}(\inf_{y\in\beta}(d(x,y))), \sup_{y\in\beta}(\inf_{x\in\alpha}(d(x,y))))$$

Let $\mathcal{C}$ be the set of simple, closed curves on $\mathcal{P}$. The *open ball* of radius $\varepsilon > 0$ and centered at γ is defined as $\mathcal{B}(\gamma,\varepsilon) = \{\alpha \in \mathcal{C}, d_H(\alpha,\gamma) < \varepsilon\}$.

Definition 1 γ is a *constriction* on $\mathcal{P}$ if there exists $\varepsilon > 0$ such that $\forall\alpha \in \mathcal{B}(\gamma,\varepsilon), L(\gamma) \leq L(\alpha)$. γ is a *strict constriction* if $\exists\varepsilon > 0, \forall\alpha \in \mathcal{B}(\gamma,\varepsilon), \alpha \neq \gamma, L(\gamma) < L(\alpha)$.

It follows from the above definition that constrictions on $\mathcal{P}$ are closed geodesics on $\mathcal{P}$.

In the sequel, we consider the case of polyhedral surfaces. Since constrictions are closed geodesics, it follows that constrictions on polyhedral surfaces are closed polygonal curves. In order to describe more precisely the shape of constrictions, we recall some definitions which can be found, for example, in [15]. For sake of simplicity, we assume in the sequel that polyhedral surfaces are triangulated.

Definition 2 A *pivot vertex* of a curve γ is a vertex of the triangulated surface through which γ goes.

Definition 3 A *sequence of faces* on a triangulated surface is a list of faces $(F_1, F_2, \ldots, F_n)$, such that $\forall 1 < i \leq n, F_i$ is adjacent to F_{i-1}.

Definition 4 Let F and F' be two faces of a triangulated surface sharing an edge E. The *planar unfolding* of a face F onto F' is defined as the image of F' when rotated about E into the plane of F, with F' lying on the opposite side of F. The planar unfolding of a sequence of faces is defined iteratively by the unfolding of faces onto the first face of the sequence (see e.g. [3, 15, 16, 19] for more explanations).

Definition 5 Let γ be a geodesic path on a triangulated surface. Let $p \in \gamma$ and $T(p)$ be the faces of the triangulated surface containing p. Let r be a strictly positive real number such that:

1. the geodesic circle $C(p,r)$ centered at p with radius r is contained in $T(p)$,
2. γ cuts $C(p,r)$ in two connected components of length l_1 and l_2.

The angle made by γ at point p is defined as the minimum of l_1/r and l_2/r.

We now recall two properties on geodesic paths, the proof of which can be found for example in [15].

Property 6 Let γ be a geodesic path on a triangulated surface. Let $\mathcal{F}$ be a sequence of faces that γ intersects. Assume γ does not go through any vertex of $\mathcal{F}$. Then γ unfolds to a straight line segment onto the planar unfolding of $\mathcal{F}$.

Property 7 If a geodesic path γ of a triangulated surface $\mathcal{P}$ passes through a vertex v of $\mathcal{P}$, the angle of γ at v is greater or equal to π.

Therefore, a constriction unfolds to a straight line segment between any two successive pivot vertices, and the angle of γ at each pivot vertex is greater or equal to π. We now characterize strict constrictions.

Proposition 8 A strict constriction on $\mathcal{P}$ goes through at least one vertex of $\mathcal{P}$.

Proof Let α be a strict constriction on $\mathcal{P}$. Assume for a contradiction that α does not go through any vertex of $\mathcal{P}$. Let E be any edge of $\mathcal{P}$ intersected by α. We call a the intersection point of α and E. Let $\mathcal{F} = (F_1, F_2, \ldots, F_n)$ be the sequence of faces that α intersects and such that $E = F_1 \cap F_n$. Since α is a geodesic path, if we unfold $\mathcal{F}$, then α unfolds to a straight line segment $[a_1a_2]$ and E unfolds to the edges E_1 and E_2 (see Figure 2). Since the angle of α at a is π, $[a_1a_2]$ is perpendicular to E_1 and E_2 and the edges E_1 and E_2 are parallel. One can find $\varepsilon > 0$ such that for every segment $[b_1b_2]$ parallel to $[a_1a_2]$ with $b_1 \in E_1$, $b_2 \in E_2$ and $d(a_1,b_1) \leq \varepsilon$, $[b_1b_2]$ is contained in the unfolding of $\mathcal{F}$. Therefore, $[b_1b_2]$ corresponds to the unfolding of a constriction β such that $L(\alpha) = L(\beta)$. Consequently, amongst all simple closed paths in any neighbourhood of α, one can find a path β whose length is equal to the length of α, which leads to a contradiction. □

Since strict constrictions always go through at least one pivot vertex, a strict constriction α can be decomposed into a circular sequence of polygonal lines $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ with $n \geq 1$ such that each polygonal line α_i passes through exactly two pivot vertices which are its begin- and endpoint and such that each α_i unfolds to a straight-line segment. Furtheremore, the angle made by α at its pivot vertices is greater or equal to π. The following proposition states that the converse is also true, which gives us a way to characterize constrictions.

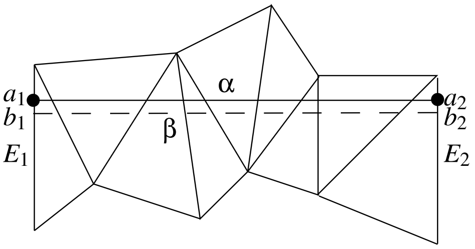


Figure 2: *Planar unfolding of a sequence of faces. The closed geodesic path α unfolds to the line segment $[a_1a_2]$, but is not a strict constriction since β which unfolds to $[b_1b_2]$ has the same length.*

Proposition 9 (Characterization of constrictions) Let α be a simple, closed curve on $\mathcal{P}$. Let $p_1, \ldots, p_n$ be the vertices of $\mathcal{P}$ through which α goes. If α unfolds to a straight line segment on $\mathcal{P}$ between any two successive vertices p_i and p_{i+1} and the angle of α at each p_i is greater or equal to π, then α is a constriction.

Proof Let α be a simple, closed curve on $\mathcal{P}$ passing through the vertices $p_1, p_2, \ldots, p_n \in \mathcal{P}$. In this proof, the indices are considered modulo n. We denote by α_i the part of α that connects p_i to p_{i+1}. Let us assume that each α_i unfolds to a straight line segment and that the angle of α at p_i is greater or equal to π. Let:

$$\alpha^{+\varepsilon} = \{x \in \mathcal{P}, d(x,\alpha) = \inf_{y \in \alpha} d(x,y) \leq \varepsilon\}$$

Let us choose ε such that the two following properties hold: First, the vertices of $\mathcal{P}$ contained in $\alpha^{+\varepsilon}$ are $p_1, p_2, \ldots, p_n$. Secondly, $\forall i \in \{1, \ldots, n\}$, $\varepsilon \leq L(\alpha_i)/4$. We define V_i as:

$$V_i = \{x \in \alpha^{+\varepsilon}, d(x,\alpha_i) \leq d(x,\alpha_j), 1 \leq j \leq n\}$$

and $M_i = V_i \cap V_{i+1}$. Let β be a simple closed curve in $\alpha^{+\varepsilon}$. Let $\beta_i = \beta \cap V_i$. In order to establish the result, we have to prove that $L(\beta) \geq L(\alpha)$. For this, we prove that $\forall i \in \{1, \ldots, n\}$, $L(\beta_i) \geq L(\alpha_i)$. We consider two cases:
Case 1: β_i intersects M_i and M_{i+1}. Let b_i and b_{i+1} be two points respectively in $\beta_i \cap M_i$ and $\beta_i \cap M_{i+1}$. Let p'_i, p'_{i+1}, b'_i and b'_{i+1} be the points corresponding to the points p_i, p_{i+1}, b_i and b_{i+1} after the unfolding of V_i. Since the angles of α at p_i and p_{i+1} are greater or equal to π, $\angle b'_i p'_i p'_{i+1} \geq \pi/2$ and $\angle p'_i p'_{i+1} b'_{i+1} \geq \pi/2$. Therefore, $L(\beta_i) \geq L(\alpha_i)$.
Case 2: β_i does not intersect one of the two sets M_i or M_{i+1}. β_i must intersect the two geodesic disks centered respectively at p_i and p_{i+1} with radius ε. Since β_i is a simple closed curve, we have $L(\beta_i) \geq 2L(\alpha_i) - 4\varepsilon$. Since $\varepsilon \leq L(\alpha_i)/4$, we conclude that $L(\beta_i) \geq L(\alpha_i)$.
□

4. Algorithm

Our algorithm searches one constriction on a closed triangulated surface in two steps:

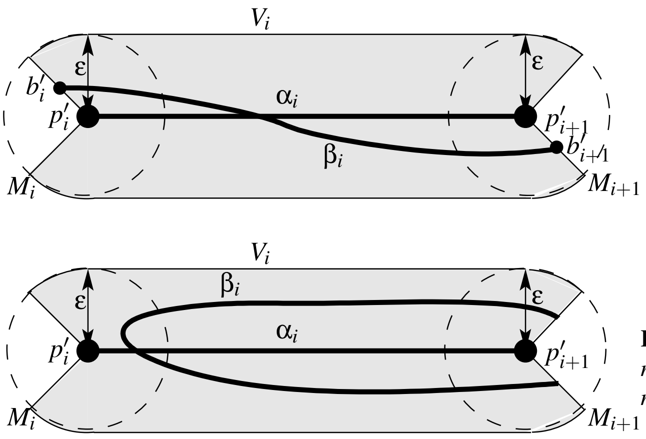


Figure 3: *Unfolding of V_i used in the proof of Proposition 9.*

1. first, the surface is simplified until we find a *seed curve*. This term will be properly defined in Section 4.1. The seed curve is used to initialize the search of a constriction;
2. second, we go back to the initial surface and compute a simple closed curve on each simplified surface. The last curve is the output of our algorithm.

4.1. Progressive surface simplification

In order to simplify the surface $\mathcal{T} = \mathcal{T}^0$, we use a classical simplification method using the *edge collapse* operator (e.g., Garland and Heckbert's algorithm[5]). We iteratively apply this operator to simplify $\mathcal{T}^0$. Let us denote $\mathcal{T}^k$ the simplified surface after k edge collapses. If $\mathcal{T}^k$ is a 2-manifold, $\mathcal{T}^{k+1}$ will have one vertex and two faces less than $\mathcal{T}^k$. The choice of the two merged vertices v_1 and v_2 and the position of the new vertex $\bar{v}$ depends on the simplification algorithm. In this paper, we assume that the new surface $\mathcal{T}^{k+1}$ produced by the chosen simplification algorithm is geometrically close to the previous surface $\mathcal{T}^k$. We stop the simplification when we find three neighbouring vertices a, b and c on a surface $\mathcal{T}^n$ such that abc is not a face of $\mathcal{T}^n$. The curve made by the three edges ab, bc and ca will be called a *seed curve* and will be denoted β_n.

Note that $\mathcal{T}^n$ has the same topology as $\mathcal{T}^0$, but if we simplify one more step, merging either a and b or a and c or b and c, $\mathcal{T}^{n+1}$ will not be a manifold anymore (see Figure 4). Therefore, if we want to detect other constrictions elsewhere on the surface, we reject the collapse of the three edges ab, ac and bc since their collapse would change the topology of the surface. We simply continue the simplification process until another seed configuration is detected somewhere else on the surface at a step $m \geq n$.

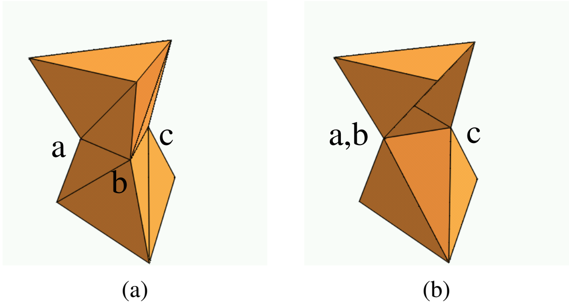


Figure 4: *(a) Seed curve (abc) on a surface. (b) After one more simplification step, the surface is not a manifold anymore.*

4.2. Constriction reconstruction

Let us assume a seed curve β_n has been detected on the simplified surface $\mathcal{T}^n$. In order to derive a constriction β_0 on the initial surface $\mathcal{T}^0$, we build a sequence of closed curves $\beta_{n-1}, \beta_{n-2}, \ldots, \beta_0$ on the sequence of simplified surfaces $\mathcal{T}^{n-1}, \mathcal{T}^{n-2}, \ldots, \mathcal{T}^0$.

β_k is computed thanks to β_{k+1}.

Let us assume the surface $\mathcal{T}^{k+1}$ has been obtained from the surface $\mathcal{T}^k$ by collapsing the edge $v_1 v_2$ to the vertex $\bar{v}$. Let $\mathcal{F}^k_{v_1,v_2}$ be the set of all the faces of $\mathcal{T}^k$ incident to v_1 or v_2, $\mathcal{F}^{k+1}_{\bar{v}}$ the set of all the faces of $\mathcal{T}^{k+1}$ incident to $\bar{v}$.

We denote by d_{k+1} the set of polygonal lines connecting two successive pivot vertices of β_{k+1} and having no intersection with the interior of $\mathcal{F}^{k+1}_{\bar{v}}$. Since d_{k+1} has no intersection with $\mathcal{F}^{k+1}_{\bar{v}}$, it is not affected by the split of the vertex $\bar{v}$ and does not need to be updated. Let $c_{k+1} = \beta_{k+1} \setminus d_{k+1}$. We have $c_{k+1} \cap \mathcal{F}^{k+1}_{\bar{v}} \neq \emptyset$. Therefore, the curve c_{k+1} is affected by the split of the vertex $\bar{v}$ and must be updated. The computation of the curve c_k which results from the update of the curve c_{k+1} is described in Section 4.3. The curve β_k is defined as:

$$\beta_k = d_{k+1} \cup c_k$$

Observe that if $c_{k+1} = \emptyset$, it means that $d_{k+1} = \beta_{k+1} = \beta_k$ and $c_k = \emptyset$. Afterwards, we assume $c_{k+1} \neq \emptyset$.

4.3. Computation of c_k

Let us now describe how $c_k \in \mathcal{T}^k$ is computed from $c_{k+1} \in \mathcal{T}^{k+1}$. We define c_k as a geodesic path between two vertices p_1 and p_2 of $\mathcal{T}^k$. In order to define p_1 and p_2, we consider two cases.

- If $c_{k+1} \neq \beta_{k+1}$, p_1 and p_2 are the two endpoints of c_{k+1}.
- If $c_{k+1} = \beta_{k+1}$, it means that $\bar{v}$ is the only pivot vertex of β_{k+1}. Let F_1^{k+1} and F_2^{k+1} be the two faces of $\mathcal{F}^{k+1}_{\bar{v}}$ intersected by β_{k+1} (since β_{k+1} is a geodesic, there are only two such faces). Let F_1^k and F_2^k be the corresponding faces

in $\mathcal{F}^k_{v_1,v_2}$. If F_1^k and F_2^k are incident to the same vertex v_1 or v_2, let $p_1 = p_2$ be the commun vertex, otherwise let $p_1 = v_1$ and $p_2 = v_2$ (see Figure 5).

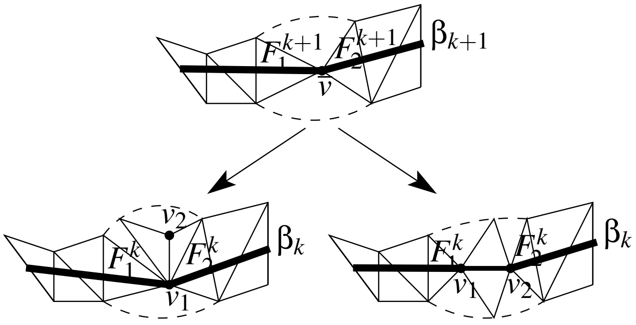


Figure 5: *Choice of the pivot vertices for β_k when $\bar{v}$ is the only pivot vertex of β_{k+1}.*

In the two cases, we have to connect p_1 and p_2 by a geodesic path in $\mathcal{T}^k$. To compute a geodesic between two endpoints using Pham-Trong's algorithm[15, 16], we need an initial sequence of faces $\mathcal{F}^k \in \mathcal{T}^k$ between these two points. We compute $\mathcal{F}^k$ using the faces $\mathcal{F}^{k+1} \in \mathcal{T}^{k+1}$ intersected by c_{k+1}. If a geodesic follows an edge of the surface between its two endpoints, we choose arbitrarily one of the two adjacent faces as the unique face of the sequence.

We consider two cases.

- If $c_{k+1} \neq \beta_{k+1}$, p_1 and p_2 are the two endpoints of c_{k+1} and c_k. In order to give more flexibility to our algorithm, we initiate the search of a geodesic path between p_1 and p_2 using two different sequences of faces between p_1 and p_2. The result depends strongly on the chosen sequence. Nevertheless, we compute the two geodesics resulting from the two different initializations and keep the shortest resulting path.
  Let F_1^{k+1} and F_2^{k+1} be the two faces of $\mathcal{F}^{k+1}_{\bar{v}}$ intersected by c_{k+1}. Let F_1^k and F_2^k be the corresponding faces in $\mathcal{F}^k_{v_1,v_2}$. Let us consider a sequence of faces in $\mathcal{F}^k_{v_1,v_2}$ such that the first face of the sequence is F_1^k, the last face of the sequence is F_2^k and the edge v_1v_2 is not a commun edge of two successive faces of the sequence. There are exactly two possible sequences of faces $\mathcal{F}_1^k$ and $\mathcal{F}_2^k$ satisfying those properties as illustrated in Figure 6. We deduce from $\mathcal{F}_1^k$ and $\mathcal{F}_2^k$ two different sequences of faces connecting p_1 and p_2 by replacing in $\mathcal{F}^{k+1}$ the sequence of faces from F_1^{k+1} to F_2^{k+1} either by $\mathcal{F}_1^k$ or $\mathcal{F}_2^k$.
- If $c_{k+1} = \beta_{k+1}$, we consider only one sequence of faces $\mathcal{F}^k$ in order to initiate the search of a geodesic path between p_1 and p_2. This sequence is obtained by replacing in $\mathcal{F}^{k+1}$ the two faces F_1^{k+1} and F_2^{k+1} by F_1^k and F_2^k.

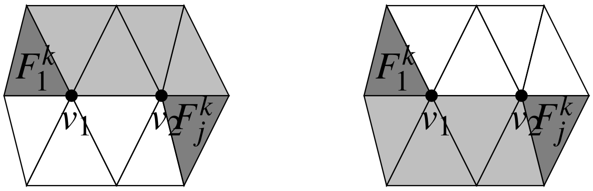


Figure 6: *There exist two possible sequences of faces between F_1^k and F_j^k.*

4.4. Detecting several constrictions

This algorithm can be extended in order to detect several constrictions on a surface.

During the simplification step, each time we find a seed curve we simply forbid the collapse of its three edges, in order to avoid a topological change of the surface. We also mark this simplification stage and this seed curve. Then we can continue to simplify the surface, until we cannot go further.

The reconstruction step starts back from the last seed curve we found. Each time we reach a marked surface $\mathcal{T}^k$, we start to construct a new constriction, initialized with the corresponding seed curve, using our algorithm.

5. Results and discussion

All following results were obtained using Garland and Heckbert's simplification algorithm[5].

5.1. Results

Figure 8 shows the two simplification steps before a seed curve is detected (left column) on a very simple "butterfly" model (10 vertices, 16 faces). At each step, the merged vertices v_1 and v_2 are pointed out, and the resulting vertex $\bar{v}$ is shown on the next surface. On the right column, the seed curve on the last surface and the reconstructed curves β_k are shown with their pivot vertices p_i.

On the simple twisted cylinder model (Figure 9, 14 vertices, 24 faces), we must simplify four times before a seed curve is found.

Figure 10 shows some constrictions detected with our method on surfaces having from 500 to 2000 faces.

On the cactus and the screwdriver models several constrictions are detected by our algorithm, but for example on the cactus model not all constrictions are detected : there exists at least one constriction around the trunk and between the two branches. This is due to the simplification process which will simplify the surface to a single face without detecting any seed curve around the trunk. But since our goal

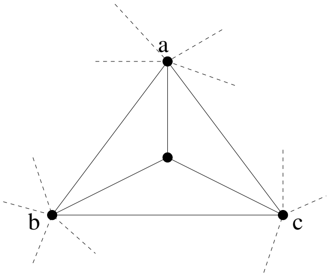


Figure 7: *This seed curve (abc) will lead to a degenerate constriction.*

is to decompose the surface into homogeneous components, we do not need to find all constrictions on a surface. The problem is the same when we have two crossing constrictions on a surface : to decompose the surface, we only need to detect one of them.

Finally, on the last exemple (the torus) one can observe that our algorithm does not take into account the object delimitated by the surface but only the surface. In this exemple, the computed constriction is more appropriate to segment the complement of the object rather than the object itself.

5.2. Discussion

Our final curves are, by construction, closed piecewise geodesics. But, even if the results look visually satisfying on the previous examples, this curves might not be constrictions. Indeed, our algorithm does not garanty that the angle made by the curve at the pivot vertices is greater or equal to π.

In order to be sure to construct constructions, we must either find a suitable condition about the seed curves, or "loosen" the reconstructed curves. This will be done in a future work.

Another problem is that some seed curves may not lead to a simple closed curve, but to a single point (which can be seen as a degenerate constriction). This is in particular the case when we have the configuration shown on Figure 7.

That is why we need to choose a suitable condition our seed curve must fulfill in order to detect only nondegenerate constriction. We noticed that selecting only seed curves whose three vertices are saddle points of the surface $\mathcal{T}^n$, will indeed lead to adequate curves, but no curve will be detected on some surfaces.

If we choose not to take into account some seed curves, we must also make sure these seed curves will not change the topology of the surface when simplifying further. At the moment, this is done by preventing edge collapse around these seed curves, as said in Section 4.4. Other strategies could be studied.

6. Conclusion and future work

In this paper we have explained a method to detect constrictions on closed triangulated surfaces, based on progressive surface simplification and geodesics computation. Our first results are promising, but the algorithm needs to be improved to be practically useful.

First of all, we must be sure to get constrictions and not only closed piecewise geodesics. As said in the previous section, a part of our future work will be to "loosen" the final reconstructed curve. Another possibility can be to loosen the curve at each reconstruction step, to make sure we have a closed geodesic. This will also be done in a future work, and compared to the first solution in order to see what is the most effective algorithm.

We must also study conditions about our seed curves and the simplification algorithm to be sure to detect all constrictions, or at least all constrictions we need to decompose the surface into homogeneous components.

A more precise definition of a constriction, or a clever condition about seed curves, is also necessary to distinguish between constrictions which cut the object delimitated by the surface in separate parts or which cut the complement of the object in separate parts.

At last, the algorithm only locally modify the curves during the reconstruction step, but we need a lot of simplification steps to detect seed curves. We thus need to precisely study the time and memory costs of our algorithm.

Acknowledgement

The authors thank Valérie Pham-Trong for her help and her interest in this work, and the reviewers for their interesting comments.

References

1. P. Agarwal, S. Har-Peled, M. Karia. Computing Approximate Shortest Paths on Convex Polytopes. *Proc. of ACM* 16^{th} *Symposium on Computational Geometry*, pp. 270–279, 2000.
2. C. Andújar. Geometry Simplification. Research report Nr LSI-99-2-R, Universitat Politècnica de Catalunya, Barcelona, 1999.
3. J. Chen, Y. Han. Shortest Paths on a Polyhedron. *Proc. of ACM* 6^{th} *Symposium on Computational Geometry*, pp. 360–369, 1990.
4. M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, W. Stuetzle. Multiresolution Analysis of Arbitrary Meshes. *ACM Computer Graphics (Proc. of SIGGRAPH '95)*, **29**:173–182, 1995.
5. M. Garland, P. Heckbert. Surface Simplification Using Quadric Error Metrics. *ACM Computer Graphics (Proc. of SIGGRAPH '97)*, **31**:209–216, 1997.

6. S. Har-Peled. Approximate Shortest-Path and Geodesic Diameter on Convex Polytopes in Three Dimensions. *Proc. of ACM 13th Symposium on Computational Geometry*, pp. 359–366, 1997.

7. P. Heckbert, M. Garland. Survey of Polygonal Surface Simplification Algorithms. *ACM Computer Graphics (Proc. of SIGGRAPH '97), Course Notes 25: Multiresolution Surface Modeling*, 1997.

8. H. Hoppe. Progressive Meshes. *ACM Computer Graphics (Proc. of SIGGRAPH '96)*, **30**:99–108, 1996.

9. T. Kanai, H. Suzuki. Approximate Shortest Path on a Polyhedral Surface Based on Selective Refinement of the Discrete Graph and Its Applications. *Proc. of IEEE Geometric Modeling and Processing 2000*, pp.241–250, 2000.

10. M. Lanthier, A. Maheshwari, J.-R. Sack. Approximating Weighted Shortest Paths on Polyhedral Surfaces. *Proc. of ACM 13th Symposium on Computational Geometry*, pp. 274–283, 1997.

11. P. Lindstrom, G. Turk. Fast and Memory Efficient Polygonal Simplification. *Proc. of IEEE Visualization '98*, pp. 279–286, 544, 1998.

12. C. Mata, J. Mitchell. A New Algorithm for Computing Shortest Paths in Weighted Planar Subdivisions. *Proc. of ACM 13th Symposium on Computational Geometry*, pp. 265–273, 1997.

13. J. Mitchell, D. Mount, C. Papadimitriou. The Discrete Geodesic Problem. *SIAM Journal of Computing*, **16**(4):647–668, 1987.

14. J. Mitchell. Geometric Shortest Paths and Network Optimization. In *Handbook of Computational Geometry*, J.-R. Sack and J. Urrutia, ed., Elsevier, pp. 633–701, 1998.

15. V. Pham-Trong. Détermination Géométrique de Chemins Géodésiques sur des Surfaces de Subdivision. Ph.D. thesis, Université Joseph Fourier, Grenoble, 1995.

16. V. Pham-Trong, L. Biard, N. Szafran. Pseudo-Geodesics on Three-Dimensional Surfaces and Pseudo-Geodesic Meshes. *Numerical Algorithms*, **26**(4):305–315, April 2001.

17. R. Ronfard, J. Rossignac. Full-range Approximation of Triangulated Polyhedra. *Computer Graphics Forum (Proc. of Eurographics '96)*, **15**(3):67–76, 1996.

18. W. Schroeder, J. Zarge, W. Lorensen. Decimation of Triangle Meshes. *ACM Computer Graphics (Proc. of SIGGRAPH '92)*, **26**:65–70, 1992.

19. M. Sharir, A. Schorr. On Shortest Paths in Polyhedral Space. *SIAM Journal of Computing*, **15**(1):193–215, 1986.

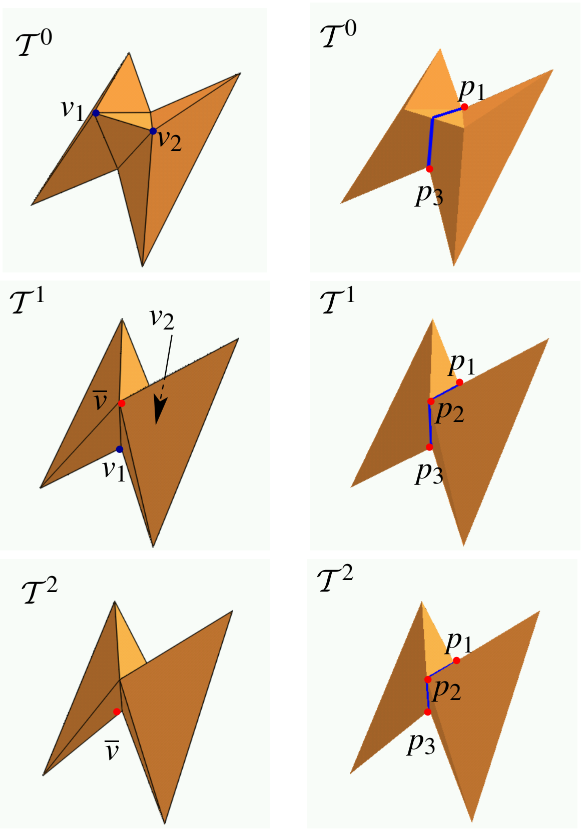
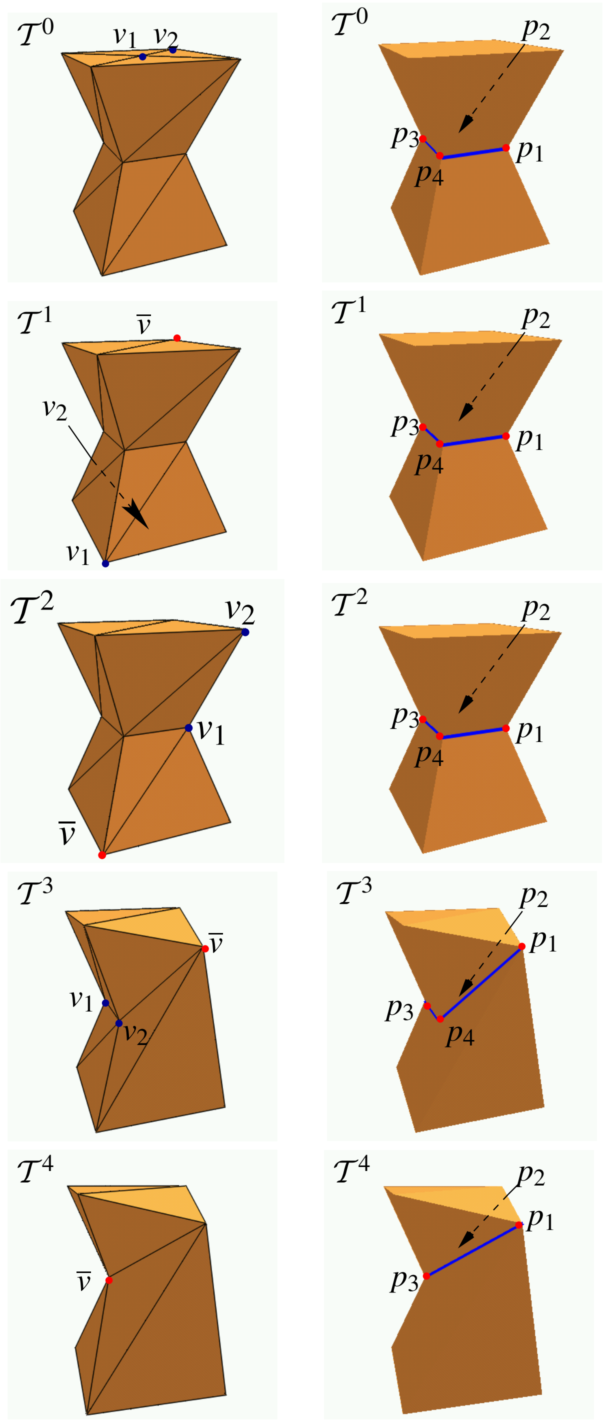


Figure 8: *Left: Simplification steps on a simple butterfly model. Right: corresponding reconstructed curves. The seed curve β_2 is made of three edges, between the vertices p_1, p_2 and p_3 of $\mathcal{T}^2$. Since $p_3 = \bar{v}$, β_1 is initialized with only two pivot vertices, p_1 and p_2, on $\mathcal{T}^1$. The computation of a geodesic path between p_2 and p_1 creates a new pivot vertex p_3. For the same reason, β_0 is initialized with only two pivot vertices, p_1 and p_3, on $\mathcal{T}^0$. Geodesic paths between them cross two faces each.*

20. G. Turk. Re-tiling Polygonal Surfaces. *ACM Computer Graphics (Proc. of SIGGRAPH '92)*, **26**:55–64, 1992.

21. K. Varadarajan, P. Agarwal. Approximating Shortest Paths on a Nonconvex Polyhedron. *Proc. of IEEE 38th Symposium on Foundations of Computer Science*, pp. 182–191, 1997.



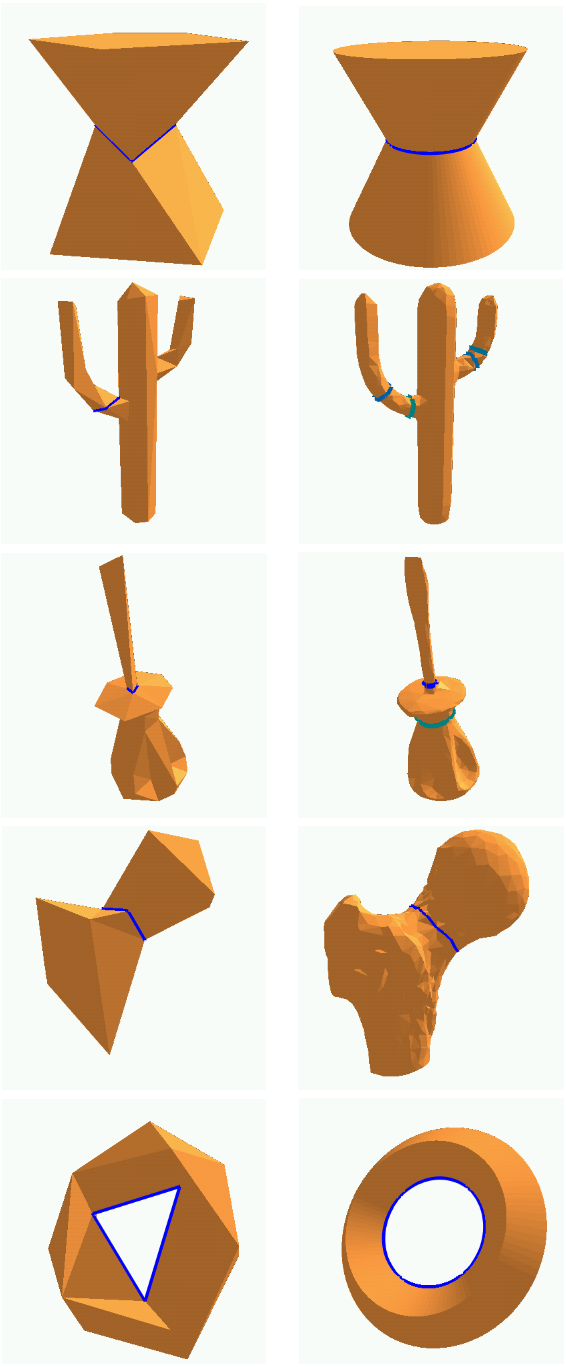


Figure 9: *Left: Simplification steps on a simple twisted cylinder. Right: corresponding reconstructed curves. The seed curve β_4 is made of three edges, between the vertices p_1, p_2 and p_3 of $\mathcal{T}^4$. Since $p_3 = \overline{v}$, β_3 is initialized with only two pivot vertices, p_1 and p_2, on $\mathcal{T}^3$. The computation of a geodesic path between p_2 and p_1 creates two new pivot vertices p_3 and p_4. For the same reason, β_2 is initialized with three pivot vertices, p_2, p_3 and p_4 on $\mathcal{T}^2$. The computation of a geodesic path between p_4 and p_2 creates a ne pivot vertex p_1. Notice that β_2 only follows edges of the sur face $\mathcal{T}^2$. The two last reconstruction steps do not modify curve: $\beta_0 = \beta_1 = \beta_2$.*

Figure 10: *Left: some simplified surfaces and their first found seed curves. Right: constrictions constructed on the initial surfaces.*