Topological Quadrangulations of Closed Triangulated Surfaces Using the Reeb Graph

Franck Hétroy and Dominique Attali

LIS Laboratory,
Domaine Universitaire, BP 46,
38402 Saint Martin d'Hères cedex, France
{Franck.Hetroy,Dominique.Attali}@lis.inpg.fr

Abstract. Although surfaces are more and more often represented by dense triangulations, it can be useful to convert them to B-spline surface patches, lying on quadrangles. This paper presents a method to construct coarse topological quadrangulations of closed triangulated surfaces, based on theoretical results about topological classification of surfaces and Morse theory. In order to compute a canonical set of generators, a Reeb graph is constructed on the surface using Dijkstra's algorithm. Some results are shown on different surfaces.

1 Introduction

Surfaces of arbitrary shape and topology are often represented in computer graphics by large fine triangle meshes, obtained with complex data acquisition hardware. Such discrete surfaces are not easy to store and handle because of the huge amount of data. Many works trying to reduce the number of triangles have been carried, see e.g. [9] for a survey. Other approaches convert these dense triangle meshes representation to other suitable models, such as parametric surfaces [5,12].

Our goal is to create a new discrete representation of the surface by large non planar quadrangles, in order to later set B-spline surface patches on them (one patch per quadrangle). Constructing a quadrangulation can be a very different task depending on whether we try to optimize the shape of the quadrangles (i.e. we try to maximize the minimum angle among quadrangles), the number of quadrangles, or the number of extraordinary vertices. Minimizing the number of extraordinary vertices can be an important point since problems generally occur at their vicinity, when analyzing continuity between B-spline surface patches or the limit surface of a subdivision scheme for example.

On one hand, most of existing techniques try to construct good shaped quadrangles with few extraordinary vertices (see e.g. [15] for a survey), but, as a result, the constructed quadrangulations are often fine. On the other hand, even if some theoretical results exist concerning the minimal number of quadrangles in a quadrangulation of a surface [8], there exists no algorithm able to construct such a minimal quadrangulation for a surface of arbitrary topology.

A. Braquelaire, J.-O. Lachaud, and A. Vialard (Eds.): DGCI 2002, LNCS 2301, pp. 57–68, 2002.

Unlike most of the previous works, we focus on the number of quadrangles, in order to reduce the amount of data used to describe the surface. This paper presents an algorithm to construct a coarse topological quadrangulation of a closed triangulated surface with strictly positive genus. First, we recall some theoretical definitions and give a theoretical topological quadrangulation of the surface based on its canonical polygonal schema (Section 2). Then we describe an algorithm to construct first a Reeb graph, then a canonical set of generators and finally a coarse topological quadrangulation of a closed triangulated surface (Section 3). Some results are shown in Section 4. Finally we conclude and discuss future work in Section 5.

2 Theoretical Background

2.1 Quadrangulations

We call *surface* a connected compact 2-manifold of $\mathbb{R}^3$. A *quadrangle* designates any subset of $\mathbb{R}^3$ homeomorphic to a closed square of $\mathbb{R}^2$. In this paper, we consider special decompositions of surfaces into vertices, edges and quadrangles that we call *pre-quadrangulations*.

Definition 1 (Pre-quadrangulation). *A* pre-quadrangulation *Q of a surface M is a decomposition of M into a finite number of vertices, edges and quadrangles such that:*

1. *Q is a cell complex;*
2. *any pair of vertices is joined by at most two edges;*
3. *any two quadrangles of Q are either disjoint or meet in a common vertex or intersect along one common edge or intersect along two non-consecutive common edges;*

Our definition includes quadrangulations as defined in [8]. But, unlike usual quadrangulations, two vertices in a pre-quadrangulation can be joined by two edges and two quadrangles can intersect along two edges. Figure 1 shows a pre-quadrangulation of the torus: we have four distinct quadrangles, all having the same four vertices A, B, C and D.

A vertex in a pre-quadrangulation is said to be *ordinary* if its valence (or degree) is 4. Vertices with a valence different from 4 are said to be *extraordinary*. We have the following remarkable relation:

Property 2. Let Q be a pre-quadrangulation of M, V_e the number of extraordinary vertices in Q, and $v_1, \ldots, v_{V_e}$ the valences of these extraordinary vertices. We have: $\sum_{i=1}^{V_e}(v_i - 4) = 8(g-1)$, where g is the genus of M.

Property 2 shows that the sum of the orders (that is to say, the valences minus four) of all extraordinary vertices is a surface invariant. This also shows that, except for surfaces with genus equal to 1, any pre-quadrangulation of a surface contains extraordinary vertices.

Since a pre-quadrangulation is a cell complex, we have the Euler relation $\chi_M = V - E + F = 2 - 2g$.

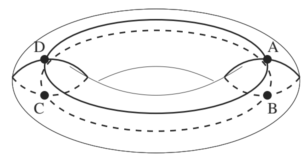

Fig. 1. Pre-quadrangulation of a torus.

Proposition 3. *Let Q be a pre-quadrangulation of a surface M. Let V be the number of vertices in Q, E the number of edges, F the number of faces (quadrangles) and χ_M the Euler characteristic of M: $\chi_M = 2-2g = V-E+F$ where g is the genus of M.*
A lower bound for the number of quadrangles needed to construct Q is $\lceil \frac{1}{2}(\sqrt{9-8\chi_M}+3)\rceil - \chi_M = \lceil \frac{1}{2}(\sqrt{16g-7}-1)\rceil + 2g$.

Example 4. For a torus ($g=1$), we have $F \geq 3$. For a 2-torus ($g=2$), we have $F \geq 6$.

Proof. According to rule nr. 2 of the definition, E is lower or equal to twice the number of pairs of vertices in Q, which is $\frac{V(V-1)}{2}$: $E \leq V(V-1)$.
Since each edge belongs to two faces, and each face has four distinct edges, $2E = 4F$. We thus have $F = V - \chi_M$ and $F \leq \frac{V(V-1)}{2}$. Consequently, $2V - 2\chi_M \leq V(V-1)$. This quadratic inequality has the solution: $V \geq \lceil \frac{1}{2}(\sqrt{9-8\chi_M}+3)\rceil$. Finally, $F = V - \chi_M \geq \lceil \frac{1}{2}(\sqrt{9-8\chi_M}+3)\rceil - \chi_M$. □

We do not know if a general formula gives the minimum number of quadrangles in a pre-quadrangulation of a surface. We now construct a pre-quadrangulation of a surface with only $4g$ quadrangles, based on the canonical polygonal schema of the surface.

2.2 Canonical Polygonal Schema of a 2-Manifold

We assume here that the reader knows some basic notions of combinatorial topology. We recall here the classification theorem for surfaces [6].

Theorem 5. *An orientable 2-manifold M with genus $g > 0$ can be represented canonically using a $4g$-gon G, where all $4g$ vertices represent the same vertex on M and edges are oriented. The labels of the edges around the polygon are of the form: $a_1 b_1 a_1^{-1} b_1^{-1} ... a_g b_g a_g^{-1} b_g^{-1}$. Curves formed by partnered edges (e.g. a_1 and a_1^{-1}) are called canonical generators of M, and G is called the canonical polygonal schema (or canonical fundamental polygon) of M.*

A canonical set of generators is composed of two kinds of generators: g generators surround the g holes (we will call them *longitudinal generators*) and g other turn around each handle associated with a hole (we will call them *latitudinal generators*). See Fig. 2 for an example and [4] for more details.

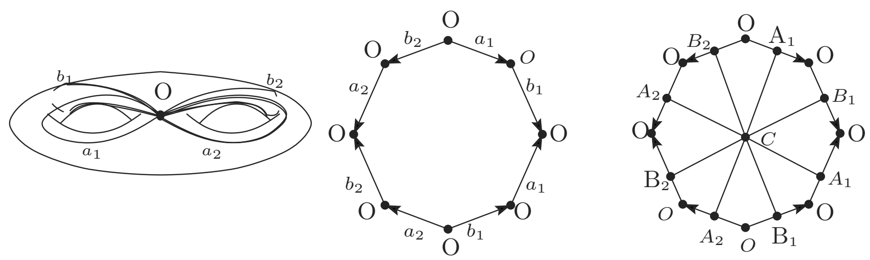

Fig. 2. Canonical set of generators, canonical polygonal schema, and pre-quadrangulation of a 2-torus. a_1 and a_2 are the longitudinal generators, and b_1 and b_2 are the latitudinal ones.

2.3 Quadrangulation of the Canonical Polygonal Schema

We present here a simple pre-quadrangulation of a surface with genus $g > 0$, with $4g$ quadrangles and $2g + 2$ vertices. This pre-quadrangulation is based on the canonical polygonal schema of the surface: half the edges of the pre-quadrangulation follow the canonical generators.

First, we choose an arbitrary vertex on the surface, O, to be the base-point of the canonical polygonal schema. We then choose $2g$ other vertices A_1, B_1, ..., A_g, B_g, and join twice O and each A_i (resp. each B_i) together, along a longitudinal (resp. latitudinal) canonical generator. We eventually choose a final vertex C (which is not on an edge) and join twice C and each A_i (resp. each B_i) together. We end up with $2g + 2$ vertices, $8g$ edges and $4g$ quadrangles. An example is shown on Fig. 2 (right).

The number of quadrangles, $4g$, is close to the lower bound of Prop. 3. The only extraordinary vertices of the pre-quadrangulation we propose are O and C, which valences are equal to $4g$ (if $g = 1$, our pre-quadrangulation does not contain any extraordinary vertex). We have chosen to minimize the number of extraordinary vertices (with respect to the number of vertices), but it implies the valence of these extraordinary vertices to be high, according to Prop. 2.

We will now call this pre-quadrangulation of the surface based on a quadrangulation of its canonical polygonal schema a *topological quadrangulation*.

2.4 The Reeb Graph

To construct a canonical set of surface generators on a triangulated surface, we will use a topological structure for compact manifolds named the Reeb graph. The Reeb graph, named after Reeb [16], and Morse theory are powerful tools to describe topological features on surfaces. Both are increasingly used in computer graphics, mainly for terrain analysis and shape modeling [17,11,18,1]. See [6] for an introduction about Morse theory and the Reeb graph.

A Reeb graph is defined as follows:

Definition 6. *Let $f : M \to \mathbb{R}$ be a real-valued continuous function on a compact manifold M. The Reeb graph of f is the quotient space of M by the equivalence relation $\sim$ defined by:*
$x_1 \sim x_2 \Leftrightarrow f(x_1) = f(x_2)$ *and x_1 and x_2 are in the same connected component of $f^{-1}(f(x_1))$.*

Figure 3 shows a Reeb graph of a height function on a torus (figure inspired by [6]).

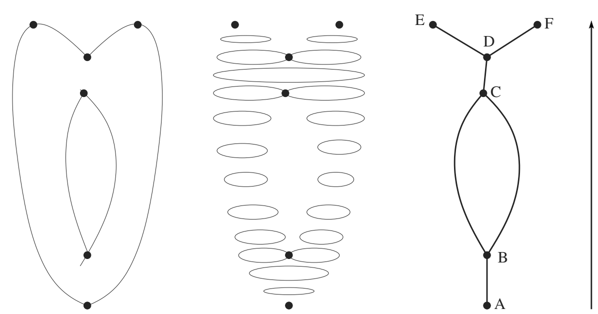

Fig. 3. A torus and the critical points of its height function f (left), connected components of some level sets for f (middle), and the Reeb graph of f on this torus.

There are two kinds of nodes in a Reeb graph:

- *Extremal nodes* are nodes incident to only one edge (e.g. nodes A, E and F on Fig. 3). They represent local extrema of the Morse function.
- *Internal nodes* are nodes incident to at least three edges (e.g. nodes B, C and D on Fig. 3). They represent saddle points of the Morse function.

To find a canonical polygonal schema of a surface, we just need to find a cycle basis in the Reeb graph; then each cycle will give us two canonical generators of the surface (one longitudinal generator and the associate latitudinal one).

3 Constructing Quadrangulations of a Triangulated Surface

3.1 Construction of a Reeb Graph on a Triangulated Surface

To define our Reeb graph, we must choose a function f. As said before, the Reeb graph is mainly used in terrain analysis. In this case f is often a height function, because we have a given orientation where the height is meaningful. In our case (surfaces) we do not have such a particular orientation, so we choose, as Lazarus and Verroust did before us [13], the *shortest path distance to a given source point on the surface.*

Very recently Hilaga et al. [10] have chosen to integrate this function over the surface, in order to avoid the choice of the source point. Since we are only interested in detecting a cycle basis on the Reeb graph, that is to say on the topology of the surface and not the exact location of the critical points of f, we do not need to choose a very subtle (but computationally more expensive) function f.
We approach f in our discrete case by the *shortest edge path to the source point on the triangulated surface*. The source point S is chosen as the furthest vertex from an arbitrary vertex: in practice S will be at the end of some long branch of the surface; consequently the direction induced by f is geometrically meaningful.

If f is a Morse function and if all its singularities are simple, we can prove that the Reeb graph we construct using this function determines the genus of the surface [20]. If singularities are not simple, we can use Carr et al. algorithm [2] to break them into multiple simple singularities. This method turns f into a Morse function.

Proposed Algorithm. To avoid confusion between the nodes (resp. edges) of the Reeb graph and the vertices (resp. edges) of the mesh, the first will be called Reeb nodes (resp. Reeb edges) and the last just vertices (resp. edges).
To construct the Reeb graph of f on the surface, we must first compute the value of f at all the vertices of the mesh, then detect the critical points of f and finally link them to create the Reeb edges. Actually, we will do the three steps at the same time, using Dijkstra's algorithm.
With each vertex X of the triangulation we associate a *value* $v(X)$ (at the end of the algorithm $v(X)$ will be equal to the value of f at X), and a list of *attributes*. Attributes will be used to identify Reeb edges: two vertices X and Y will have the same attribute if and only if the segment $[f(X), f(Y)]$ contains non critical value of f and X and Y belong to the same connected component of $f^{-1}([f(X), f(Y)])$. At the beginning, each vertex value is $+\infty$ and vertices have no attribute, except the source point which value is 0 and which has one attribute.
At each step the current processed vertex is the vertex X with minimal value: we know that $f(X) = v(X)$. Values of all neighbouring vertices of X are updated using Dijkstra's algorithm.
We then compute the number of sign change $n(X)$ for $v - v(X)$ when we cover all neighbouring vertices Y. Since we use Dijkstra's algorithm, if $v(Y) < v(X)$ then $v(Y) = f(Y)$ and thus $f(Y) < v(X)$, otherwise $v(X) < f(Y)$. Thus $n(X)$ is also the number of sign change for $f - f(X)$. Furthermore, note that $n(X)$ is always even (our triangulated surface is closed, so the neighbourhood of X is a ring).
If $n(X) = 0$, X is a local extremum for f, that is to say an extremal Reeb node. His only attribute will be equal to the (first) attribute of its predecessor (for Dijkstra's algorithm). If $n(X) = 2$, X is not a Reeb node. His only attribute will also be equal to the (first) attribute of its predecessor. Finally if $n(X) \geq 4$,

X is an internal Reeb node. We compute its local level set and its connected components (see below). Each vertex of each upper connected component will be given a new attribute (the same for all vertices in the same connected component). X will be given all attributes of its upper and lower connected components.

A Reeb edge between vertices A and B with $f(A) < f(B)$ will be computed as a path made from B to its successive predecessors through Dijkstra's algorithm until A or a vertex with the corresponding attribute is found, and the path made from this vertex to A following a connected component of A (if A is not the source point, A is an internal node of the graph, so its local level set has been computed).

Local Level Sets. The *local level set* $LS(X)$ *of* X is a set of pairs of vertices (Y, Z) where $f(Y) > f(X)$ and $f(Z) < f(X)$ (see Fig. 4). An *upper* (resp. *lower*) *connected component* of X will be a connected component of the set of all Y (resp. Z) in $LS(X)$ such that $f(Y) > f(X)$ (resp. $f(Z) < f(X)$). Note that a connected component of X will be a set of vertices, and not a set of pairs of vertices.

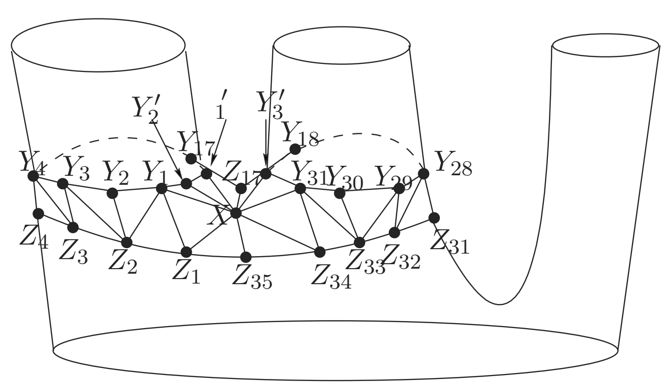

Fig. 4. Local level set of an internal node X and its connected components. Here we have two upper connected components ($Y_1...Y_{17}Y_1'Y_2'$ and $Y_{18}...Y_{31}Y_3'$) and one lower connected component ($Z_1...Z_{17}...Z_{35}$).

To compute $LS(X)$ and the connected components of X, we use a very simple algorithm:

```
LS:=0
Repeat
  Choose two adjacent vertices Y and Z in the neighbourhood of X
  such as f(Y)>f(X) and f(Z)<f(X)
```

```
  Add (Y,Z) to LS
  avoid_vertex:=X
  new_vertex:=Y
  While new_vertex is different from X do:
    new_vertex:=the vertex adjacent to Y and Z which is not avoid_vertex
    If f(new_vertex)<f(X) then
      Add (Y,new_vertex) to LS
      avoid_vertex:=Z
      Z:=new_vertex
    Elsif f(new_vertex)>f(X) then
      Add (new_vertex,Z) to LS
      avoid_vertex:=Y
      Y:=new_vertex
    End if.
  End while.
  We now have two new connected component of X (not complete)
Until it is no more possible to choose such two adjacent vertices.
```

To complete our connected components and possibly merge them, we compute the vertex NV adjacent to X and Y (where Y is the last vertex of a connected component) which is not the previous vertex of the connected component. We add it to the connected component, replace Y by this new vertex and do the same until NV is in $LS(X)$. If the last vertex NV (which is in $LS(X)$) is also in the connected component, this connected component is complete. If not, we must merge it with the connected component where is NV.

3.2 Construction of a Canonical Set of Generators

Only a few algorithms have been proposed to construct a canonical set of generators [19,3,14]. All three only use mesh properties and not the Reeb graph. Providing that the genus of the surface is strictly positive, finding a cycle basis $\mathcal{B}$ on our constructed Reeb graph will give us information about the location of surface holes, and thus about surface generators.
To find $\mathcal{B}$, we use a traditional graph theory algorithm: we find a maximal forest[1] on the Reeb graph, then for each edge which is not in this forest we find a minimal cycle containing this edge. The set of all minimal cycles found is $\mathcal{B}$. Such algorithms are detailed in graph theory books, e.g. [7].

Once we have found $\mathcal{B}$ we can construct a canonical set of surface generators: for each cycle $\mathcal{C}$ in $\mathcal{B}$ there exists one longitudinal generator and one latitudinal generator. The longitudinal generator is simply $\mathcal{C}$. To construct the latitudinal generator we find the cycle vertex V with the minimal value, and then follow an upper connected component of V (we have computed the local level set of V when we have constructed the Reeb graph).

[1] A forest is a subgraph with no cycle.

3.3 Topological Quadrangulations of a Triangulated Surface

To construct our pre-quadrangulation on the triangulated surface, we must first choose which vertices will be O, the A_k, the B_k and C (cf. Section 2.3 for the notations).

We take O equals to the source point and C equals to the vertex for which f is maximum. Then for each k we define B_k as the internal node of the Reeb graph with the lowest value and belonging to the k^{th} latitudinal cycle of the basis, and A_k as the internal node with the greatest value and belonging to the k^{th} longitudinal cycle of the basis.

To construct the edges OA_k and OB_k of the quadrangulation, we continuously distort each generator of the constructed canonical set of generators so that O belongs to it (this can be done in time $O(g.n)$, n being the number of vertices in the initial triangulation, since Dijkstra's algorithm computes for each vertex a path to the source point). Then we follow the distorted generators: the longitudinal ones for the edges OA_k and the latitudinal ones for the edges OB_k. We can do the same to construct the edges CA_k and CB_k, since O and C are dual points on the pre-quadrangulation (see Fig. 5 for an example): edges CA_k correspond to the latitudinal generators and edges CB_k correspond to the longitudinal generators.

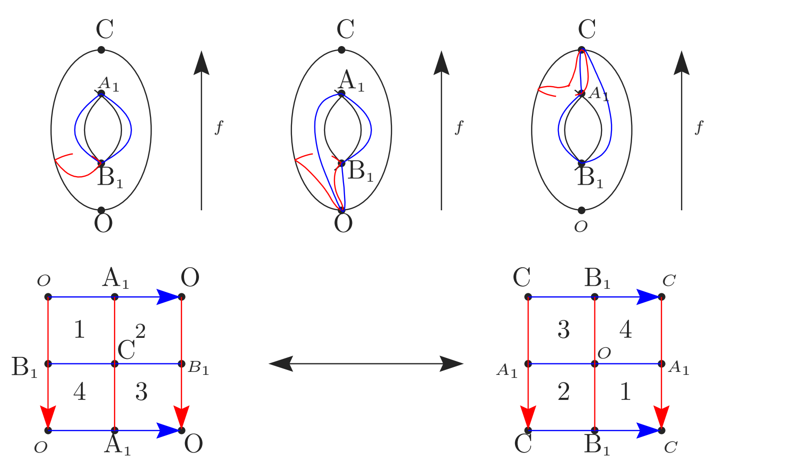

Fig. 5. Generators of a torus (top left), edges OA_i and OB_i of the topological quadrangulation (top middle), edges CA_i and CB_i of the topological quadrangulation (top right). The bottom figure shows the quadrangulation "centered" either at C or at O (quadrangles are numbered from 1 to 4).

4 Results

Some results are shown on Fig. 6. Top left figure shows the Reeb graph computed on a topological sphere (no hole, 1071 vertices, 2138 triangles). Note the local (non global) extremum on the middle of the helix. The two other top figures show the Reeb graph and a canonical set of generators computed on a rocker arm with one hole (10044 vertices, 20088 triangles). This model is courtesy of Cyberware (`www.cyberware.com`). The two bottom left figures shows canonical sets of generators of a knot-shaped torus (4245 vertices, 8490 triangles; the latitudinal generator is bottom left of the figure) and a topological 3-torus (1572 vertices, 3152 triangles). Finally bottom right figure shows the coarse topological quadrangulation of a simple torus (400 vertices, 800 triangles) constructed with our algorithm.
The computation time of our algorithm is $O(n^2)$. Time used by Dijkstra's algorithm to construct the Reeb nodes is $O(n.\log(n))$, but then to construct the Reeb edges we need to compute the local level sets of Reeb nodes. This takes $O(n^2)$ time.

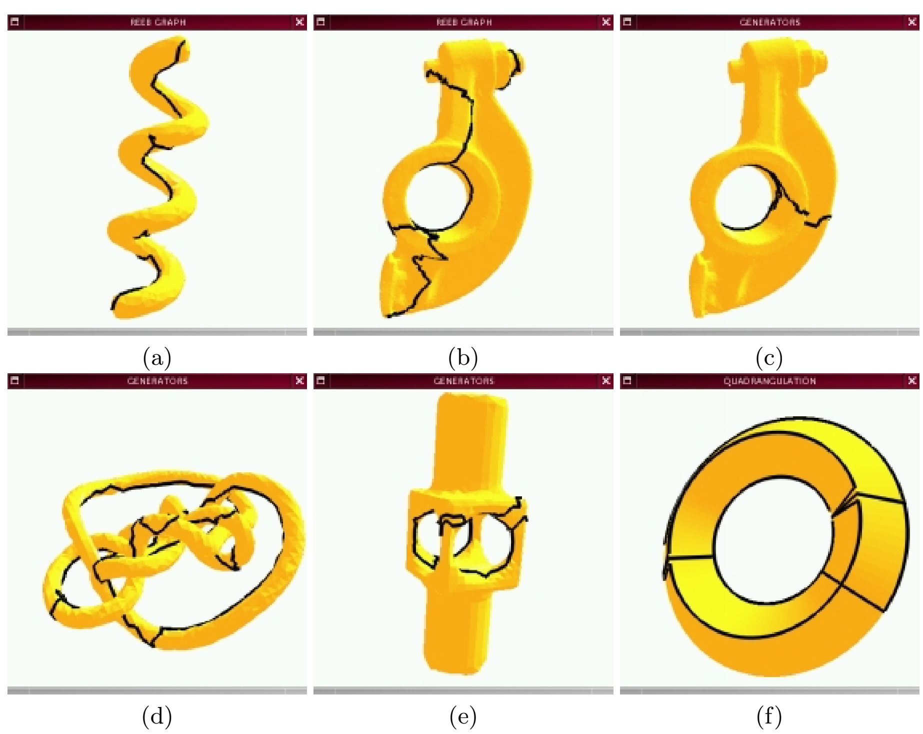

Fig. 6. Reeb graph (a,b), canonical set of generators (c,d,e), and topological quadrangulation (f) of some closed surfaces.

5 Conclusion and Future Work

We have presented theoretical results and a new algorithm to construct a new discrete representation of a closed triangulated surface with strictly positive genus by a coarse topological quadrangulation. Our method first construct a Reeb graph on the surface using Dijkstra's algorithm and then the generators of this surface.
Further work includes quadrangulation refinement to take surface geometrical singularities into account, and optimization (in terms of angles in quadrangles, extraordinary vertices, ...). The choice of the function used to define the Reeb graph can also be improved to fulfill these goals.

Acknowledgment. The authors would like to thank Francis Lazarus for fruitful discussions and helpful comments, especially concerning the number of cycles in the Reeb graph.

References

1. S. Biasotti, B. Falcidieno, M. Spagnuolo. Extended Reeb Graphs for Surface Understanding and Description. Proceedings of DGCI'00, Lecture Notes in Computer Science, Vol. 1953, pp.185-197, 2000.
2. H. Carr, J. Snoeyink, U. Axen. Computing Contour Trees in All Dimensions. Proceedings of ACM 11^{th} Symposium on Discrete Algorithms, pp. 918-926. San Francisco, California, USA, Jan. 2000.
3. T. Dey, H. Schipper. A new Technique to Compute Polygonal Schema for 2-Manifolds with Application to Null-Homotopy Detection. Discrete and Computational Geometry,14:93-110, 1995.
4. T. Dey, S. Guha. Computing Homology Groups of Simplicial Complexes in $\mathbb{R}^3$. Journal of ACM, Vol. 45, No. 2, pp. 266-287, 1998.
5. M. Eck, H. Hoppe. Automatic Reconstruction of B-Spline Surfaces Of Arbitrary Topological Type. Proceedings of SIGGRAPH'96, pp. 325-334, August 1996.
6. A.T. Fomenko, T.L. Kunii. Topological Modeling for Visualization. Springer, 1997.
7. M. Gondran, M. Minoux. Graphs and Algorithms. Wiley, 1995.
8. N. Hartsfield, G. Ringel. Minimal Quadrangulations of Orientable Surfaces. Journal of Combinatorial Theory, Series B, Vol. 46, No. 1, pp. 84-95, 1989.
9. P.S. Heckbert, M. Garland. Survey of Polygonal Surface Simplification Algorithms. Multiresolution Surface Modeling Course, SIGGRAPH'97, 1997.
10. M. Hilaga, Y. Shinigawa, T. Kohmura, T.L. Kunii. Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes. Proceedings of SIGGRAPH'01, August 2001.
11. M. van Kreveld, R. van Ostrum, C. Bajaj, V. Pascucci, D. Schikore. Contour Trees and Small Seed Sets for Isosurface Traversal. Proceedings of ACM 13^{th} Symposium on Computational Geometry, pp. 212-220. Nice, France, June 1997.
12. V. Krishnamurthy, M. Levoy. Fitting Smooth Surfaces to Dense Polygon Meshes. Proceedings of SIGGRAPH'96, pp. 313-324, August 1996.
13. F. Lazarus, A. Verroust. Level Set Diagrams of Polyhedral Objects. Proceedings of ACM 5^{th} Symposium on Solid Modeling and Applications, pp. 130-140. Ann Arbor, Michigan, USA, June 1999.

14. F. Lazarus, M. Pocchiola, G. Vegter, A. Verroust. Computing a Canonical Polygonal Schema of an Orientable Triangulated Surface. Proceedings of ACM 17^{th} Symposium on Computational Geometry, pp. 80-89. Tufts University, Medford, USA, June 2001.
15. S. Owen. A Survey of Unstructured Mesh Generation Technology. Proceedings of the 7^{th} International Meshing Roundtable, Sandia National labs, pp. 239-267. Dearborn, Michigan, U.S.A., October 1998.
16. G. Reeb. Sur les Points Singuliers d'une Forme de Pfaff Complètement Intégrable ou d'une Fonction Numérique. Comptes Rendus Acad. Sciences, Paris, France, 222:847-849, 1946.
17. Y. Shinagawa, T.L. Kunii. Surface Coding based on Morse Theory. IEEE Computer Graphics and Applications, pp. 66-78, September 1991.
18. S. Takahashi, Y. Shinagawa, T.L. Kunii. A Feature-based Approach for Smooth Surfaces. Proceedings of ACM 4^{th} Symposium on Solid Modeling and Applications, pp. 97-110. Atlanta, Georgia, USA, May 1997.
19. G. Vegter, C.K. Yap. Computational Complexity of Combinatorial Surfaces. Proceedings of ACM 6^{th} Symposium on Computational Geometry, pp. 102-111. Berkeley, California, USA, June 1990.
20. Z. Wood. Semi-Regular Mesh Extraction from Volumes. Master's thesis, Caltech, Pasadena, California, USA, 2000.