Title:
Reconstruction of Volumes from Soup of Faces with a Formal Topological Approach

Authors:
Hakim Belhaouari, hakim.belhaouari@univ-poitiers.fr, University of Poitiers, XLIM UMR CNRS 7252
Sebastien Horna, sebastien.horna@univ-poitiers.fr, University of Poitiers, XLIM UMR CNRS 7252

Keywords:
3D modeling, Topological reconstruction, Volumes and faces intersection

DOI: 10.14733/cadconfP.2018.246-251

Introduction:

Nowadays, digital 3D models are key components in many industrial and scientific sectors, such as architecture, geoscience, simulation and obviously computer graphics. Numerous tools propose to build virtual objects, either by construction from scratch, or by reconstruction from acquisition data (point cloud, photos, and so on). However, each application has its own quality requirements that restrict the class of acceptable and supported models [1, 5].

Although 3D modeling is more and more successful, 3D models often correspond to a geometrical representation: either by a set of faces, or (in the best case) by a set of volumes. No additional topological information is available, except the face neighborhood of volumes (according to the used modeler). Hence, the produced objects may still contain inconsistencies, especially interpenetrations between elements of the same dimension (volume to volume, face to face or edge to edge). Although 3D model libraries are increasingly created/published, these objects cannot be used directly. Indeed, a geometrical correction and a dedicated information reconstruction are relevant in most application contexts.

Based on previous researches [3, 4, 10, 13], a large survey of model repairing methods is proposed in [5]. More precisely, authors distinguish two main categories for repairing models: global volumetric and surface approaches. Volumetric approaches consist in reconstructing a new valid mesh after transformation of existent mesh into an intermediate mesh. These methods are often too time consuming and relies on chosen intermediate structure. On the contrary, it solves highly inconsistent model in a robust way [5, 14]. Surface approaches work directly on the input mesh, and attempt to repair locally (self-)intersections *one by one* [3, 14]. Thus, modifications on original mesh are very limited to inconsistent parts. However, the resolution of intersecting geometry is numerically unstable which may produce artifacts. Finally, above mentioned methods can not respect at same time robustness, accuracy or efficiency properties.

Our purpose is clearly different, because boolean operations produce, from surface objects (i.e. 2D topological objects with immersions in 3D), a unique volume built from them. In addition, processed volumes must be spatially closed (they have an interior and an exterior) in various simulation as visibility problem in rendering [11]. Thus, our approach keeps all topological volumes called co-refinement (similar to Minkowski sum).

In this paper, we present a topological reconstruction method of volumes from soup of polygons in a robust, accurate and efficient way. The main purpose is the direct production of a 3D model with

Proceedings of CAD'18, Paris, France, July 9-11, 2018, 246-251

all topological information in any dimension (i.e. neighborhood of edges, faces and volumes) and all embedding information (color, domain properties and so on) for various simulation algorithms. One of our scientific contribution is this correction method that relies on the topological information in order to improve efficiency and to control geometrical imprecision. More, embedding information (domain specific data) are handled automatically along reparation in order to maintain consistent business model. Finally, the topology provides enough information on corrected objects to be used directly in many application domains (3D printing, simulation, visualization).

Fundamental Background:

Many data structures exist in computer graphics to represent cells subdivisions with their adjacency relations [7], among which generalized-maps (G-maps) are a formalization of all these concrete data structures [12]. G-maps also present a formal and mathematical definition that allows us to address robustness problems. Similar topological models have already been used in topological reconstruction [9, 11] and refinement process [6, 8]. The main advantage of G-maps model is a uniform description through dimension of structure, operations and additional defined properties. Here we use a non-oriented graph [2], where nodes (denoted by the set G) are basic elements called darts and arcs represent the adjacency relationship. Each arc is labeled depending on the dimensional relationship (often called α_i). Thus, a $3-$G-map represents intuitively a 3D object from its decomposition into topological cells.

G-map may represent oriented and non-oriented objects. Here, we want to handle real-world objects as a set of disjoint volumes. We must define a set of properties for identifying an acceptable object. That is why we use a 3D space subdivision that helps us to define formally closed and oriented 3D partition. The space partition criteria relies on a mix of geometry and topology, where we define C_i as the geometrical space of a $i-$cell (e.g. C_3 defines the interior space of a volume, C_2 defines the area of a face, C_1 is the length of an edge, C_0 is the vertex itself). Formally, a 3D object is well formed if and only if a space partition P satisfies the Equation 2.1. In other words, two cells i must not occupy the same space and/or must be connected along cells of lower dimensions.

$$\forall i \in \{0,1,2,3\}, \forall C_i^1, C_i^2 \in P, C_i^1 \cap C_i^2 = \emptyset \vee \exists j \in [0; i[, C_i^1 \cap C_i^2 = C_j \text{ where } C_j \in P \quad (2.1)$$

The object closure property is well known in the Computer Graphics community but with $3-$G-map this property differs and becomes a topological closure defined in the Equation 2.2. Less formally, it means there is no self-loop in the graph thus opened faces are prohibited. This property relates to the dangling face, a specific case where $d.\alpha_2 = d.\alpha_3$ that corresponds to a geometric hole. In other word, after topological reconstruction holes are detected immediately.

$$\forall d \in G, \forall i \in [0; 3], d.\alpha_i \neq d \quad (2.2)$$

The last property is the orientation: G-maps may represent oriented (or not) objects. For representing orientation, we include a special boolean embedding (often named *orient*). Formally, the property behind is defined in the Equation 2.3 where orient accesses to the special boolean and it forces that two adjacent darts have an opposed value for any dimensions.

$$\forall d \in G, \forall i \in [0; 3], d.orient \neq d.\alpha_i.orient \quad (2.3)$$

Topological Reconstruction and Correction Process:

Our contribution lines up to repair/clear soup of faces to obtain volumes. Initially, the load process reconstruct basic faces (correct them if necessary) and affect initial embedding information depending on the model (for instance orientation, color or nature of element as material).

The first step attempts to reconstruct a first topology by linking faces that have a common geometric edge. The angle arrangement makes it and produces volumes, that satisfy partially consistent space partition. In practice, the angle arrangement connects around geometrical edges, incident topological faces in a specific order. The topological reconstruction process guarantees the respect of closure and orientation properties. Whereas the partition property is partially satisfied.

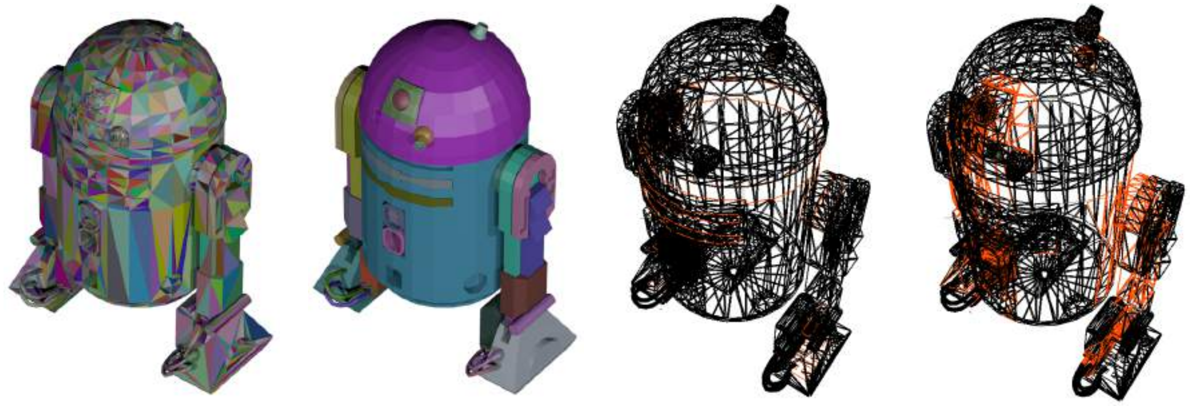

Fig. 1: Topological reconstruction result: (a) initial set of geometric faces; (b) after the topological reconstruction, several volumes are built (one per color); (c) immediate detection of dangling face; (d) visualization of face intersections (in red).

Figure 1 presents result of the topological reconstruction process. From a set of polygon, our topological reconstruction process produces a set of connected volumes (see Fig. 1(b)). Thanks to topological information, unclosed volumes which contains dangling face are directly detected (see Fig. 1(c)).

The second step consists in cleaning/repairing because up until this point, the reconstruction process does not completely satisfy the partition criteria. The original model may include several other inconsistencies (face overlapping, self-intersection, ...) as shown in Figure 1(d). The 3D co-refinement splits objects correctly until satisfaction of the partitioning constraints. Volume intersections (and self intersection) are solved by 3D co-refinement of lower dimension cells. We therefore use two types of co-refinement according to the dimension of the treatment for our object embedded in 3D. One of our notable contribution consists in solving the problem by using the topological information to reduce computations.

Edges intersection process ($1-$co-refinement) relies heavily on computing intersections between topological edges. As a topological edge regroups multiple geometrical edges (one per incident face), the number of tested elements is reduced. Furthermore, the insertion of vertex impacts automatically all incident faces without additional computation.

The $2-$co-refinement handles face intersection in 3D. This feature takes into consideration the support plane besides each face polygon. In 3D, The intersection of two planes is a line, if they are not parallel. When the intersection exists, it is possible to split both faces along the *splitting line* with some constraints. Concretely the split decision becomes a $1D$ problem depending on the intersection between the splitting line and each edge of faces. The study of cross interval and origin of the stored information indicates if the split is required or not. In a real complex model, due to the great number of polygons, a face may be intersected by several other faces. To achieve that, one of our contribution is to buffer all splitting lines in an ad-hoc structure without effective split. A splitting line stores all sorted intersection vertices along its director vector from all edges of each intersected face. An analysis of this line with all vertices delimit for each face a segment with a set of additional vertices.Then, faces are finally cut along this resulting segment. Thus, if several faces share a same split line, then all intersections vertices may be reported

Proceedings of CAD'18, Paris, France, July 9-11, 2018, 246-251

on each concerned face. A more complex case treats intersection between splitting lines. The process is similar to the previous case but intersection vertex of splitting line is added to the basic process. A last treatment is dedicated for overlapped faces, which may handle the business logic.

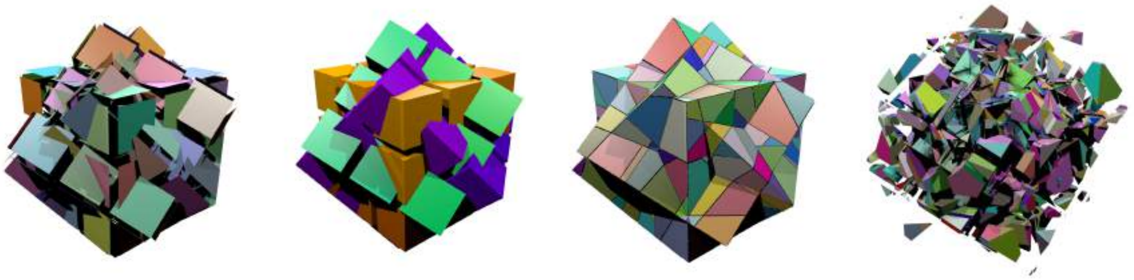

Fig. 2: Global process: (a) 3 Rubik's cubes overlapped (total: 324 faces); (b) result after first topological reconstruction, 81 volumes are built; (c) result after 3D co-refinement process, all faces and edges are split along intersections, the model is composed of 3998 faces and 618 volumes; (d) exploded view of all obtained volumes.

To conclude this section, Figure 2 illustrates the final full reconstruction process. Initially, a first reconstruction is performed after importation of raw data (see Fig. 2(a-b)). Then, we exploit the topology in order to improve accelerator structure and we apply the cleaning/repairing stage for ensuring all the properties of our model (see Fig. 2(c)). Finally, the process applies a last reconstruction in order to consider new edges and faces for satisfying the partition criteria (see Fig. 2(d)).

Results and Discussion:

This section presents results of our method and to prove generality of our approach we take different meshes from various domains (lighting, geology, artistic). Object examples are extracted of community meshes and a real geological data. The chosen objects are illustrated in Figure 3. Each model presents particularities and highlights specific part of our method. Among all examples, we count small and big objects, and more or less correct objects. For instance, Figure 1(a) is our synthetic model because it contains few faces but a lot of intersections. Suzanne and Stanford Bunny (see Figs. 3(a)-3(d)) are well formed. Whereas, Figures 3(d) is artistic creations with many unintentional errors. Figure 3(c) is a real geological mesh created from noised data that induce multiple interpenetration errors.

Table 1 presents execution time of our process on meshes of the Figure 3. This study has been performed on an Intel i7-2600 with 16Go RAM under Linux OS. Regular grids are used for optimizing computation time. The purpose of this structure is to avoid unnecessary tests for remote edges/faces, since their geometrical location in the scene do not allow possible intersection.

Generally, the topological reconstruction is fast and results depends heavily on the number of incident faces to a common geometric edge. The 3D co-refinement does not depend directly of the mesh size and depends drastically on number of intersection in objects.

Conclusions:

In this paper, we present a topological reconstruction method of volumes from a soup of polygons (from various input data source: obj, off, others geometrical format and G-map format). This framework relies on the topological structure G-map and divides reconstruction into two major steps: topological 3D angle

Proceedings of CAD'18, Paris, France, July 9-11, 2018, 246-251

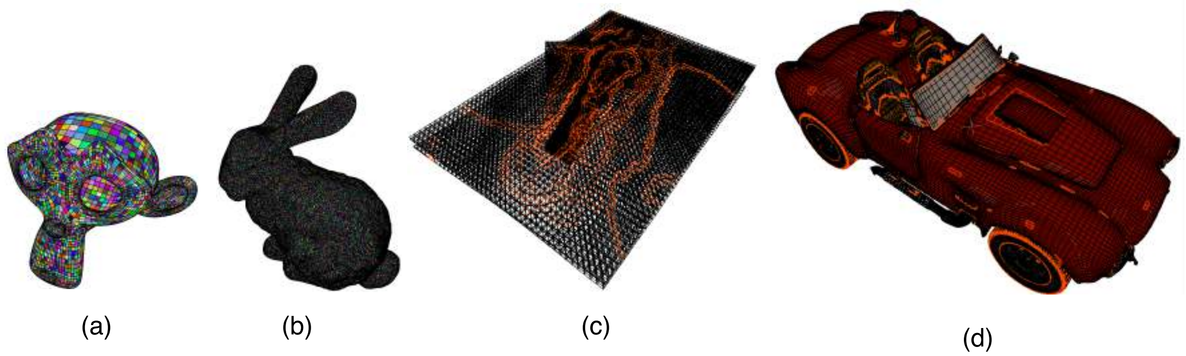

Fig. 3: Used meshes for experimenting our process, (self-)intersections are underlined in red: (a) 3 intersected Rubik's cubes, many faces are intersected several times; (b) CAD gears; (c) droid; (d) Stanford Bunny; (e) geological model; (f) artistic toy; (g) fan cobra car.

Mesh	#Faces	Topo Reconstruction	#Volumes Built	Co-refinement	#Intersections Processed
1(a)	6,872	0.4s	66	0.2s	2,165
2(a)	324	0.02s	81	0.03s	2,674
3(a)	6,912	0.05s	1	0.04s	326
3(b)	69,451	3.2s	1	0.4s	0
3(c)	61,720	3.5s	10	1.5s	537
3(d)	342,000	26s	1,740	23m12s	59,966

Table 1: Execution time for the full 3D-reconstruction and 3D-co-refinement process.

arrangement, then cleaning/repairing operations (called co-refinement). Furthermore, the proposed angle arrangement allows to reduce number of effective test and to avoid multiple reapplication of the whole process. The co-refinement gathers all split operations in order to decrease the number of artefact and to ensure best computation on initial mesh directly. All these operations lead to satisfy a specific topological model with strong criteria that ensure high quality of the produced object. Finally, the obtained objects are formed from disjoint volumes where all topological and embedding information are consistent. And they may be directly used for various simulations algorithms.

Acknowledgement:

This research is a part of collaboration between Total S.A., the company Geosiris and XLim Laboratory. We would like to thank Total S.A. and Geosiris for providing us gladly real geological data from the Alwyn site located in the Northwest Europe.

References:

[1] Attene, M.; Campen, M.; Kobbelt, L.: Polygon mesh repairing. ACM Computing Surveys, 45(2), 15:1–15:33, 2013. http://doi.org/10.1145/2431211.2431214.

[2] Belhaouari, H.; Arnould, A.; Le Gall, P.; Bellet, T.: JERBOA: A graph transformation library for topology-based geometric modeling. In 7th International Conference on Graph Transformation (ICGT 2014), vol. 8571 of LNCS. Springer, York, UK, 2014. http://doi.org/10.1007/978-3-319-09108-2_18.

[3] Bischoff, S.; Kobbelt, L.: Structure preserving cad model repair. Computer Graphics Forum, 24(3), 527–536, 2005. http://doi.org/10.1111/j.1467-8659.2005.00878.x.

[4] Bischoff, S.; Pavic, D.; Kobbelt, L.: Automatic restoration of polygon models. ACM Transactions on Graphics, 24(4), 1332–1352, 2005. http://doi.org/10.1145/1095878.1095883.

[5] Botsch, M.; Kobbelt, L.; Pauly, M.; Alliez, P.; Levy, B.: Polygon Mesh Processing. AK Peters, 2010. ISBN 978-1-56881-426-1.

[6] Brandel, S.; Schneider, S.; Perrin, M.; Guiard, N.; Rainaud, J.F.; Lienhard, P.; Bertrand, Y.: Automatic building of structured geological models. Journal of Computing and Information Science in Engineering, 5(2), 138–148, 2005. http://doi.org/10.1115/1.1884145.

[7] Brisson, E.: Representing geometric structures in d dimensions: Topology and order. Discrete and Computational Geometry, 9(4), 387–426, 1993. http://doi.org/10.1007/BF02189330.

[8] Cazier, D.; Dufourd, J.F.: A formal specification of geometric refinements. The Visual Computer, 15(6), 279–301, 1999. http://doi.org/10.1007/s003710050178.

[9] Diakité, A.A.; Damiand, G.; Van Maercke, D.: Topological Reconstruction of Complex 3D Buildings and Automatic Extraction of Levels of Detail. In Eurographics Workshop on Urban Data Modelling and Visualisation. Eurographics Association, Strasbourg, France, 2014.

[10] Gueziec, A.; Taubin, G.; Lazarus, F.; Horn, W.: Converting sets of polygons to manifold surfaces by cutting and stitching, 383–390. Visualization 1998. IEEE Proceedings, 1998. http://doi.org/10.1109/VISUAL.1998.745327.

[11] Horna, S.; Damiand, G.; Meneveaux, D.; Bertrand, Y.: Building 3D indoor scenes topology from 2D architectural plans. In International Conference on Computer Graphics Theory and Applications (GRAPP). Spain, 2007.

[12] Lienhardt, P.: N-dimensional generalized combinatorial maps and cellular quasi-manifolds. International Journal of Computational Geometry And Applications, 4(3), 275–324, 1994. http://doi.org/10.1142/S0218195994000173.

[13] Murali, T.M.; Funkhouser, T.A.: Consistent solid and boundary representations from arbitrary polygonal data. In Proceedings of the 1997 Symposium on Interactive 3D Graphics, 155–163, 1997. http://doi.org/10.1145/253284.253326.

[14] Skorkovská, V.; Kolingerová, I.; Benes, B.: A simple and robust approach to computation of meshes intersection. In Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 175–182. INSTICC, SciTePress, 2018. ISBN 978-989-758-287-5. http://doi.org/10.5220/0006538401750182.