

## AUTOMATIC REPAIR OF CITYGML LOD2 BUILDINGS USING SHRINK-WRAPPING

J. Zhao<sup>a, b \*</sup>, H. Ledoux<sup>a</sup>, J. Stoter<sup>a</sup>

<sup>a</sup> Department of GIS Technology, Faculty of Architecture and The Built Environment,  
Delft University of Technology, Delft, the Netherlands  
- (J.Zhao-3, H.Ledoux, J.E.Stoter)[@tudelft.nl](mailto:tudelft.nl)

<sup>b</sup> Center for Spatial Information Science and Sustainable Development, Tongji University, Shanghai, P.R. China

**KEY WORDS:** LoD2, Repair, Shrink-wrapping, Tetrahedralization

### ABSTRACT:

The LoD2 building models defined in CityGML are widely used in 3D city applications. The underlying geometry for such models is a GML solid (without interior shells), whose boundary should be a closed 2-manifold. However, this condition is often violated in practice because of the way LoD2 models are constructed and exchanged. Examples of the resulting errors are holes in the wall surface, intersecting and overlapping building parts etc. Those invalid models often cannot be accepted by downstream analytical applications that demand 2-manifold exterior shells for LoD2 building models. Unlike traditional local mesh repair approaches, this paper presents a global repair method for invalid LoD2 building models. Our method is based on the idea of shrink-wrapping a valid bounding surface to the invalid model. It starts by extracting the convex hull of a given model, all the faces of both the input model and the convex hull are treated as constraints in the subsequent tetrahedralization process. Defects like intersections and overlapping between polygons are also handled in the process. Then, based on a heuristic carving process, the bounding convex hull shrinks by incrementally deleting the insignificant boundary tetrahedra and wrapping the exact geometry of the building, holes and gaps are filled accordingly. The method makes no assumption on the input model, regardless of the type of geometric errors and the forms of the building. The output model is a watertight bounding shell, which is valid and represents the exterior of the building.

### 1. INTRODUCTION

Three-dimensional (3D) LoD2 building models defined in CityGML are widely used in GIS applications, such as city planning, navigation and environmental analysis (Kolbe, 2008; Kolbe et al., 2008; Gröger and Plümer, 2009). The geometric type of such models is solid without interior shells, whose boundary should be a closed 2-manifold (Herring, 2005; Gröger et al., 2012). However, a LoD2 building model is often modelled incorrectly in practice due to mistakes made in data acquisition or modelling processes (Wagner et al., 2012). As a result, geometric errors such as gaps in a surface (Figure 1 a), intersection between surfaces (Figure 1 b), non-manifold situations (Figure 1 c) and degenerated edge and surface primitives (Figure 1 d) are present, which violates the criteria of a valid LoD2 building geometry. These flawed geometric models eventually lead to serious errors or crashing of the downstream applications. Therefore, they should be repaired before being used.

The repair of 3D models, especially of a triangular mesh, is already a popular research area in the field of CAD and computer graphics (McKenney, 1998; Mezentsev and Woehler, 1999). Campen et al. (2012) have provided an elaborate review of the current progress of mesh repair researches, but most of the existing methods are proposed for continuous meshes and

few of them deal with regular shaped 3D building models. In the GIS field, 3D model repair has become an important topic in recent years because of the increasingly available 3D city models worldwide. However, not much work has been done except the validation of datasets (Kazar et al., 2008; Ledoux et al., 2009; Karki et al., 2010) and the 2D repair using triangulation (Ledoux et al., 2012). Recently, a validation and repair pipeline for CityGML models has been proposed (Bogdahn and Coors, 2010; Wagner et al., 2012). Although both the geometric as well as semantic errors are visited, its repair capability is still quite limited.

This paper proposes a repair method for LoD2 building models, which uses an initial approximation to gradually approach the correct shape of the input building model following a shrink-wrapping process. This repair method takes the advantage of constrained tetrahedralization and therefore can accept any arbitrary input model. A heuristic carving process, where tetrahedra are removed sequentially, is employed, which guarantees validity for the output and maintains the shape of the building. Our experiments show that common errors like gaps in the model, holes in the surface, intersections between building parts and unnecessary interior building structure can all be fixed using the proposed method.

---

\* Corresponding author.

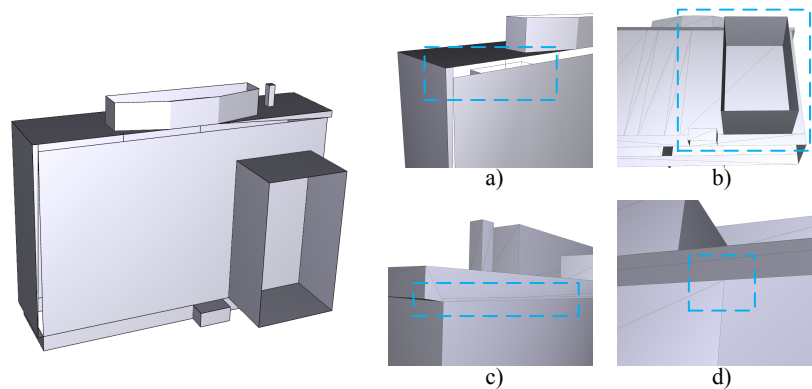


Figure 1. Demonstration of errors in a 3D building model (a) a gap on the façade of the left model; b) intersections between building parts in the left model; c) a non-manifold edge shared by multiple faces; d) a T-junction caused by a degenerated triangle)

## 2. RELATED WORK

### 2.1 Geometric repair of 3D models

The existing model repair methods can be classified into two broad categories: local approach and global approach.

Local approaches deal with each of the defects locally, for example the filling of holes (Liepa, 2003), the splitting of non-manifold edges (Guéziec et al., 2001) and the removal of intersections (Campen and Kobbelt, 2010). However, this kind of approach usually solves one or a few of errors and may introduce new errors, as stated by Campen et al. (2012). Moreover, the local approaches are sensitive to the error types contained in the input model and may not function properly when unexpected errors are present, such as intersection of surfaces.

Global approaches take the advantages of a volumetric representation, i.e. voxels, tetrahedra. These methods try to fix the volumetric representation of the object and then reconstruct the surface, thus are robust. In (Nooruddin and Turk, 2003), voxels are used to represent the spaces divided by the input mesh. Holes and other geometric errors are healed by using morphologic operators defined for voxels. The final mesh is extracted using iso-surface extraction. In the repair step, each voxel has to be classified properly by using an expensive multi ray-stabbing method, and unwanted discretizing artefacts are introduced in the result, even with a smoothing and a mesh optimization process applied afterwards. This is especially problematic for regular and planar surfaces like these of 3D buildings. In (Bischoff and Kobbelt, 2005), a shape preserving method is proposed to extract the sharp features from an Octree-based representation of the input, but this method cannot preserve the input tessellation.

### 2.2 Shrink-wrapping algorithm

The concept of shrink-wrapping is not new in geometric processing. It simulates the process where a membrane is shrunk and finally wrapped an object. Kobbelt et al. (1999) introduce the idea to remeshing a polygonal surface model into a result with the property of subdivision connectivity. This method uses an initial mesh with the desired property as a membrane, and then a force that applied to each of the vertices moves the vertices accordingly. However, this method is not proposed for flawed models with holes or intersections. Koo et al. (2005) use a similar method to shrink-wrap an unstructured point cloud to reconstruct the mesh model. Voxels are used to

structure the points and thus smoothing has to be applied in the end. This approach cannot deal with input surfaces. In order to simplify a polygonal mesh, Hagbi and El-Sana (2010) propose a tetrahedron carving approach, which can generate several topologically simplified models. This method uses the similar shrink-wrapping concept and incrementally carves the tetrahedra according to certain criteria. Although the heuristic carving paradigm seems similar to our approach, this method only accepts valid input, thus it cannot deal with geometric errors. Furthermore, their carving operation may produce non-manifold results, which have to be fixed by using stitching and cutting method (Guéziec et al., 2001). Another similar approach is the approximation of the input polygon soup based on implicit surfaces (Shen et al., 2004). However, the interpolation with an implicit surface cannot preserve the tessellation of the input geometry and introduce a smooth effect, which is not wanted for 3D buildings.

## 3. REPAIR BY SHRINK-WRAPPING

With regard to the solid-based geometric form of a LoD2 building model, the repair method from top-down using a shrink-wrapping approach is desirable, because most of the geometric errors can be solved in this uniform process and the validity of the result can be ensured. To implement this idea, first of all, a “membrane” has to be constructed for the input model, which should be valid and easy to build. Then, a step-by-step shrinking process should be applied. There are two options for this task. One is to directly move all the primitives of the membrane, i.e. vertices or faces, towards the proper positions of the original input model. However, the exact correspondences between primitives of the membrane and the input model are difficult to model, especially for the input with non-zero genus or even with holes or intersections (Kobbelt et al., 1999). Another option is to develop the process implicitly based on an intermediary structure, such as voxels or tetrahedra. As discussed before, the voxelization of an input model could not preserve either the exact shape of the model or its tessellation. The latter defect is particularly unacceptable for the repair of an input model with properties attached to its primitives, such as a CityGML model. Therefore, the constrained tetrahedralization is a better choice to structure the input model and provide elements for shrinking process.

The purpose of shrinking is to gradually eliminate the superfluous elements and only keep the desired ones, and finally wrap the correct exterior shell of the model. By using tetrahedra, this process can be implemented by a carving operation similar to (Hagbi and El-Sana, 2010). However,

proper heuristics and constraints should be introduced to guide and confine the operations, so that to fix the errors.

According to the above analysis, the proposed repair work flow using shrink-wrapping includes four consecutive steps: 1) the initial approximation; 2) the constrained tetrahedralization; 3) the heuristic carving and 4) the extraction of the exterior shell.

### 3.1 Initial approximation

The initial approximation of the shrink-wrapping process which acts as a membrane can be any valid shell that bounds the input model. It is straight forward to use the convex hull of the input model, which is easy to compute and satisfies the validity criterion for the approximation. Comparing to the alternative bounding box or sphere, a convex hull reuses a subset of the input vertices thus eases the sequent step of tetrahedralization. In addition, it offers a tight bounding shell which is more efficient for carving.

### 3.2 Constrained tetrahedralization

The constrained tetrahedralization (CT) decomposes the geometric set into 3-simplices (tetrahedra) that are non-overlapping and every input constraint, i.e. triangle surfaces in both the input model as well as the convex hull, are represented in the tetrahedralization result. Therefore, the input geometry can be preserved during repair. However, CT cannot be directly conducted due to the possible errors of intersections and degeneracies. These errors should first be handled by tessellation and decomposition. The tessellation step triangulates all the input polygons and results in triangles that are not degenerated as illustrated in Figure 2 b). Then, the decomposition step detects all sorts of intersection between triangles and decomposes the intersected ones (Figure 2 c)). The result of this step is a complex (can be further improved by coplanar merging as shown by Figure 2 d)), which is the demanded input for CT. During the process of CT, the Delaunay property is not mandatory because more Steiner points will not help improve the repair result but rather introduce extra expenses for carving.

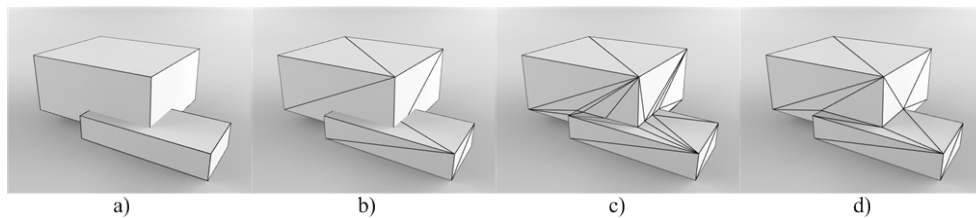


Figure 2. Tessellation and decomposition of intersecting objects (a) the input model; b) tessellated result; c) decomposition using triangle-triangle intersection; d) refined result by merging coplanar triangles)

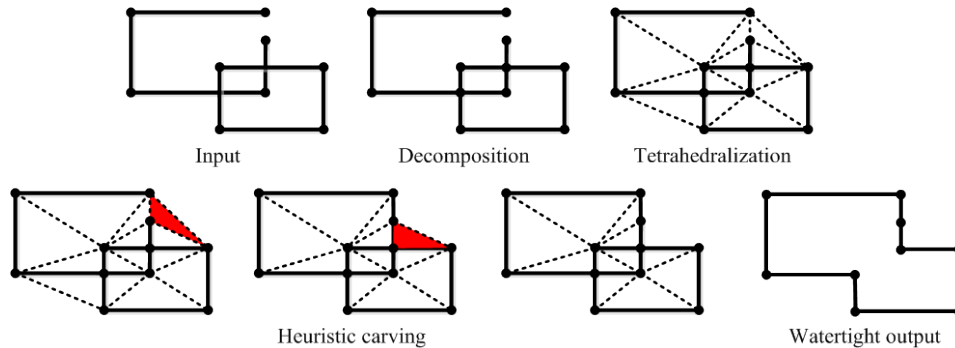


Figure 3. The 2D demonstration of the approach (in heuristic carving, a hole is filled because carving inwards the hole breaks one of the constraints)

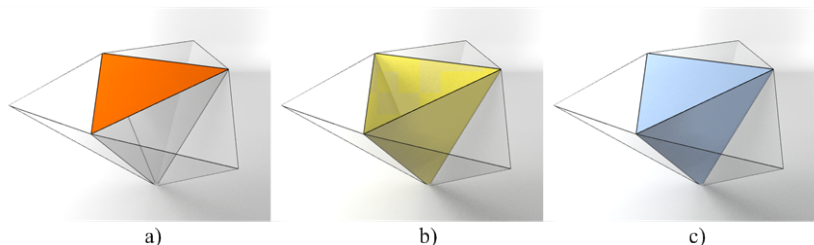


Figure 4. Demonstration of the carving operation (a) the candidate tetrahedron with a triangular facet colored in red<sup>1</sup>; b) after carving, the rest three triangular facets are classified as candidate triangles colored in yellow; c) if the tetrahedron is preserved, all its triangular facets are classified as fixed, colored in blue)

<sup>1</sup> This color configuration will be used throughout this paper (current candidate tetrahedron: red; candidate triangles: yellow; fixed triangles: blue).

### 3.3 Heuristic carving

The carving operation is only executed on the candidate tetrahedra. These tetrahedra are extracted based on the candidate triangles, which are on the boundary shell but not part of input model. After a carving operation, a candidate tetrahedron is removed and its adjacent tetrahedra that were not on the boundary are then converted to candidates, as shown in Figure 4. Then the boundary shell is shrunk. To make sure the carving does not produce invalid geometry (non-2-manifold) in the final output, validity constraints are required. A candidate tetrahedron that breaks the constraints should be removed from the candidates and be preserved in the result. Carving should also take the shape of a building into consideration, especially the planar features. As a result, a shape constraint is also developed to preserve candidate tetrahedra with such a feature. Additionally, because the carving operation only eliminates one tetrahedron each time, a heuristic paradigm should be used to guide the process. Otherwise, all the tetrahedra have to be examined before a carving operation, which is expensive. The above process is the key of this repair method and will be discussed in detail in the next section.

### 3.4 Extraction of the exterior shell

Finally, the exterior shell is extracted from the boundary tetrahedra remained after the heuristic carving process. An illustration of the whole process in 2D is shown in Figure 3.

## 4. HEURISTIC CARVING

This section explains the heuristics and constraints used in detail. Hagbi and El-Sana (2010) also discuss and use different carving styles for simplification, however these styles cannot heal errors of the input model and the validity of the output model is not considered.

### 4.1 Triangle classification

As mentioned before, the candidate tetrahedra are extracted based on the type of triangles. For this purpose, all the triangles that are in the input model and the convex hull are classified into two categories, namely the fixed triangle and the candidate triangle. The fixed triangles are those that will contribute to the final output. Because we assume that all the input surfaces should be preserved during repair, these fixed triangles thus are initialized by all the triangles of the input model after tessellation and decomposition. The candidate triangles are boundary triangles newly generated from the shell approximation. If a tetrahedron is carved, its unclassified triangular facets are converted to candidate triangles on the fly (Figure 4 b)). Otherwise, all its four triangular facets are tagged as fixed (Figure 4 c)). Based on this classification, the candidate tetrahedra are then defined as those with at least one of their four triangular facets tagged as candidate triangle.

### 4.2 Carving constraints

The criteria for the decision of a carving operation are specified as constraints. We consider two basic constraints essential for the repair of building models, i.e. the validity constraint and the shape constraint. If the carving of a candidate tetrahedron breaks one of these constraints, (that means the operation will either introduces invalid geometry to the result or damages the

desired shape of the input model), the tetrahedron should therefore be converted to fixed.

**4.2.1 Validity constraints:** Validity is the underlying requirement for all the repair methods. When carving, it is therefore uppermost to test whether a possible invalid situation will be introduced. Because the output should be a 2-manifold, the validity constraints detect possible non-manifold situations after a carving operation. However, it should be noted that when a non-manifold situation occurs after carving, it does not necessarily mean that the final result will also be invalid, because the temporary invalid situations might be resolved by the succeeding carving operations. Therefore, we check the validity constraints based on the fixed triangles in a candidate tetrahedron and its neighbours.

As showed by Botsch et al. (2007), the non-manifold situations can happen at triangles, edges or vertices. Thus the validity of all these primitives of a candidate tetrahedron should be ensured. For a fixed triangle, the invalid case is that the triangle becomes a dangling triangle after carving, as shown in Figure 5. Topologically speaking, each of the fixed triangles of a candidate tetrahedron to be carved should have exactly two neighbouring tetrahedra. This is defined as Constraint I in this repair method.

For an edge, it should be ensured that carving will not turn the edge into a complex edge, i.e. an edge shared by more than two fixed triangles. Figure 6 a) and b) illustrate the star tetrahedra of an edge and its projection in 2D. It is obvious that only when the edge is already on the boundary of the approximated shell (as shown by the red edge in Figure 6 a)), its validity should then be checked due to the possible non-manifold situation generated after carving. To prevent such invalid case, the carving around the edge should result in a connected component. Therefore, all the neighbouring tetrahedra of the edge should be carved continuously. As shown in Figure 6 b), because the carving operation could not go cross the fixed triangles, which should be preserved in the result, it will eventually produce two disconnected components. However, in Figure 6 c) and d), carving from the left side of adjacent tetrahedra will finally reach the existing boundary shell, therefore valid results can be obtained. As a result, the Constraint II for edges is defined as: if an edge of a candidate tetrahedron to be carved is on the boundary shell, a path made up by non-fixed triangles in the star tetrahedra of the edge must exist. It should connect the triangular facets of the tetrahedron to be carved and the surface of the existing boundary shell.

For a vertex, its validity constraint is similar to that of an edge: if it is already on the boundary of the shell, its validity should be checked while carving one of its incident candidate tetrahedra. The non-manifold case of a vertex is that the vertex is singular, i.e. shared by more than one volumes. This can be illustrated by using the sphere map concept (as illustrated in Figure 7 c) and f)) where the sphere represents the local neighbourhood of a vertex in which the black half-spaces indicate the incident "occupied" volumes and the white half-spaces indicate the incident "void" volume (Granados et al., 2003). It is obvious that if the sphere map of a vertex is segmented into more than two disconnected "occupied" half-spaces in the result, the vertex becomes invalid (Figure 7 f)). Therefore, if a path made up by the adjacent non-fixed tetrahedra (face-to-face adjacency) can be found which connected the triangles of the candidate tetrahedron to be carved and the current boundary shell (Figure 7 a)), this means the carving will not introduce one more half-space to the

neighbourhood of the vertex thus is valid (Figure 7 b)). Otherwise, if such a path does not exist (Figure 7 d)), the carving operation will result in invalid output as shown in Figure 7 e). As a result, the Constraint III for vertices is defined as: if a vertex of a candidate tetrahedron to be carved is on the

boundary shell, the tetrahedron should connect to the boundary shell from a path of connected non-fixed tetrahedra.

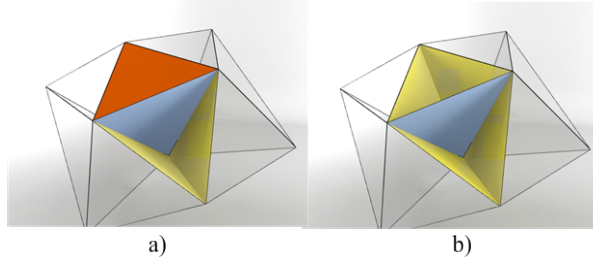


Figure 5. Illustration of Constraint I (a) a candidate tetrahedron with a fixed triangular facets; b) the carving results in a dangling triangle which breaks the constraint I)

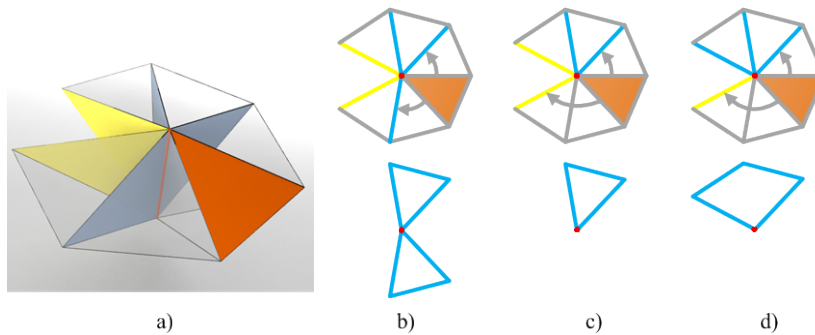


Figure 6. Illustration of Constraint II (a) the star tetrahedra of an edge (red) in the candidate tetrahedron; b) the 2D projection of the left case (top) and the carving will lead to an invalid result (bottom); c) d) two valid cases for carving)

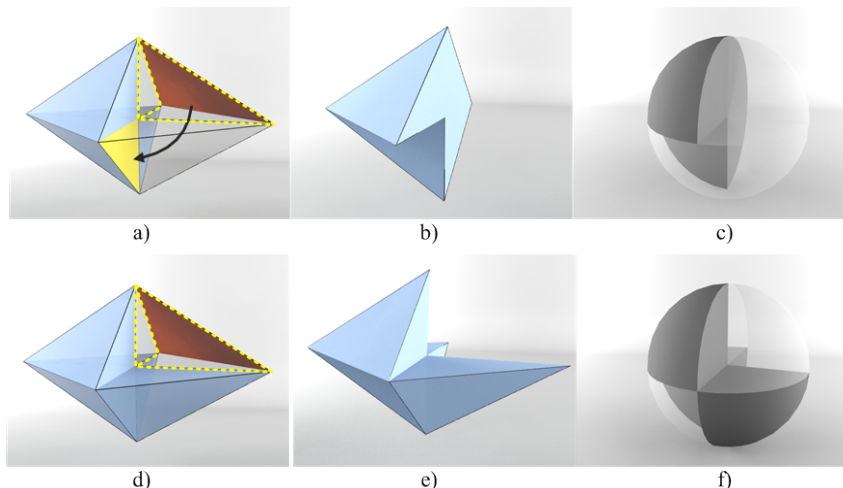


Figure 7. Illustration of Constraint III (a) the star tetrahedra of a vertex of a candidate tetrahedron, in which a path (black arrow) can be found from the candidate tetrahedron to the boundary shell (yellow); b) the possible result of carving of a); c) the sphere map of the center vertex of b); d) another case of the star tetrahedra of the vertex, in which all paths from the candidate tetrahedron to the boundary shell are blocked by the fixed tetrahedra; e) the possible result of carving of d); f) the sphere map of the center vertex of e))

**4.2.2 Shape constraints:** Based on the validity constraints, the result of the shrinking-wrapping will be a closed 2-manifold surface. However, if holes are present in a model (Figure 8 a)), the shrink process may carve these areas and results in an undesired concave shape on the surface as shown in Figure 8 b). This is especially undesired for building models because such models are often formed by planar features. As a result, a constraint, concerning the planar shape characteristics of a building, is introduced (Constraint IV). This constraint detects all the connected coplanar candidate triangles of a candidate triangle. If all these triangles are bounded by fixed triangles (as shown by triangles 2 in Figure 9 b)), it implies that these triangles can be a missed planar surface of the input model. Therefore carving should be prohibited inside this area. Instead, if these triangles are not bounded by fixed triangles (as shown by triangles 1 in Figure 9 b)), these triangles must be the auxiliary geometry generated from approximation and can then be carved. Based on the experiments, we found this simple shape constraint to be sufficient to fix many missing geometry of a building model. However, more shape constraints should be proposed for specific feature types, such as non-planar holes.

### 4.3 Carving priority

Besides the introduced constraints, the carving operations should be executed in a proper order, so that the knowledge of the input model is considered as much as possible in each step of carving. A pragmatic choice is the best-first strategy which

first carves the most promising candidate tetrahedron. Therefore, two indicators which describe the “promisingness” of a tetrahedron are proposed. One is the “degree of freedom” (DoF) and the other is the “carving profit” (Cp).

The DoF is defined as the number of known triangles of a candidate tetrahedron, e.g. a tetrahedron with  $i$  candidate triangles and  $j$  fixed triangles (where  $1 \leq i + j \leq 4$ ) has a DoF of  $(4 - i - j)$ . The smaller the DoF, the stronger the consequence of the carving is bounded and vice versa. The Cp is defined as the difference of the number of candidate triangles of a candidate tetrahedron with the number of newly produced candidate triangles after carving. For instance, if a tetrahedron with  $i$  (where  $1 \leq i \leq 3$ ) candidate triangles is carved, which introduces  $k$  (where  $1 \leq k \leq 3$ ) new candidate triangles, the Cp of this tetrahedron is  $(i - k)$ . Carving that starts with a larger Cp decreases the overall DoF of all the candidate tetrahedra in the sequent steps. As a result, a candidate tetrahedron with smaller DoF and higher Cp should be carved first.

The proposed heuristic carving work flow is shown in Figure 10. In practice, the volume of the candidate tetrahedron is also used in the heuristics. We found that carving of a tetrahedron with a larger volume first will generally improve the overall performance of the method.

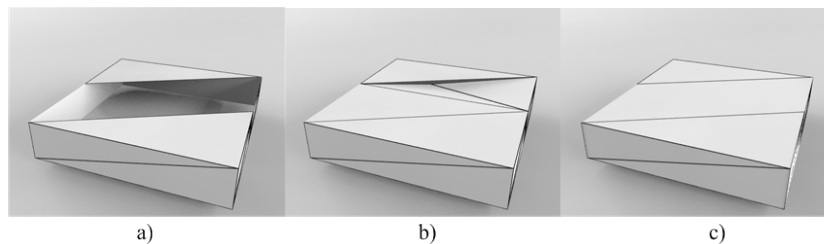


Figure 8. Demonstration of the effect of Constraint IV (a) a cube with a hole; b) the result of repair without the constraint IV; c) the result of repair with the constraint IV)

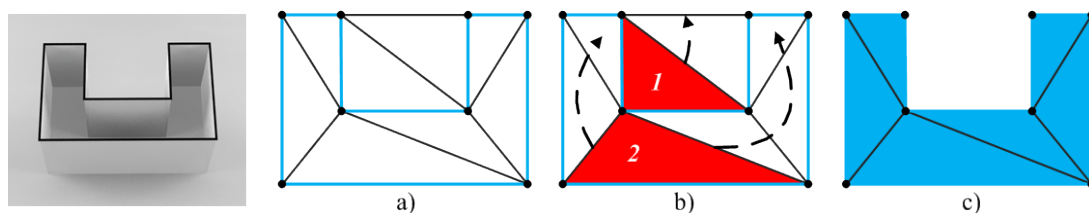


Figure 9. Illustration of Constraint IV (a) the 2D projection of the tetrahedralized result of the left case: a tessellated coplanar surface where boundary edges of fixed triangles are colored in blue; b) the connected neighbours of two candidate; c) carving result when applying the constraint IV)

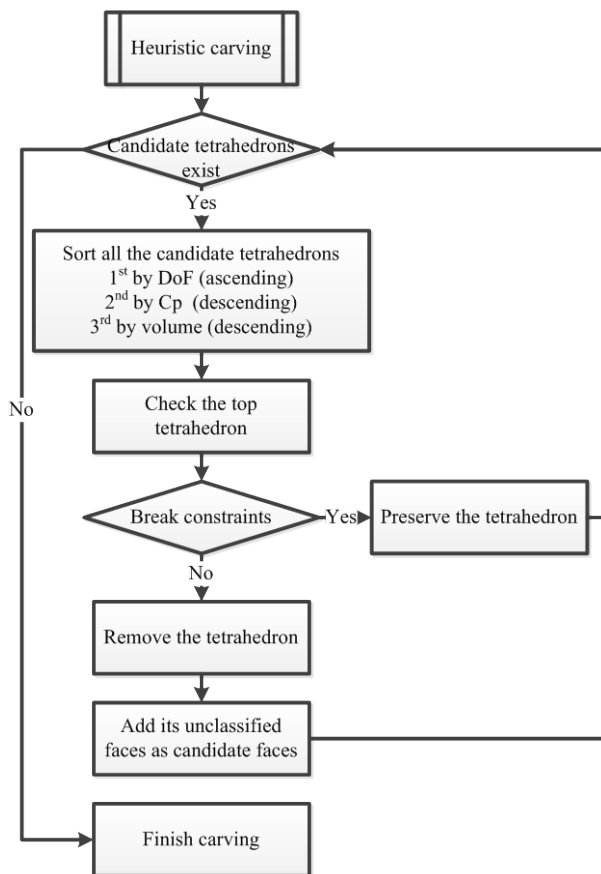


Figure 10. The flow chart of the heuristic carving process

## 5. EXPERIMENTAL RESULTS

### 5.1 Implementation details

In the implementation, the constrained tetrahedralization is one of the essential parts. We employ the third-party package Tetgen to realize the process (Si, 2006). Tetgen allows the preservation of the convex hull of an input and therefore can accept input models with holes. The latest version of Tetgen also allows the minimising of the Steiner points, which results in a smaller set of tetrahedra. However, Tetgen is not designed to solve the intersections and degeneracies of the input. Thus, all these cases should be detected first and the input model should be decomposed into a valid complex, i.e. surfaces only touch at shared boundary edges. Due to the simplicity it brings to the following processes, triangle tessellation of polygons is first conducted for the input models. Then a fast triangle-triangle intersection test is executed for each pair of possible

intersected triangles (Möller, 1997). When intersecting, both triangles are decomposed accordingly into smaller triangles that are free of intersection. If the semantics, i.e. properties of surfaces, need to be preserved, this information should be kept or propagated to the tessellation of input polygons, which is intuitive. During the carving process, our algorithm does not violate the original tessellation thus preserves the semantics. Of course, the semantics of the newly generated surfaces should be extracted or deduced from the input model according to certain rules. Moreover, if the coplanar triangles share the same semantics, they can be merged which further simplifies the surface.

### 5.2 Repair of real building datasets

We take two flawed building models as the test data. The building model shown in Figure 11 a) contains two intersecting building parts with the bottom surface and part of the roof surfaces missing (indicated by the blue arrows). Figure 11 b) shows the decomposition result. In Figure 11 c), the convex hull of the input model is constructed as the initial approximation. All the surfaces of the convex hull and of the input model are used as constraints in the tetrahedralization. After carving, the missing surfaces are filled as shown in Figure 11 d). Then, the exterior shell of the tetrahedra is extracted which is a watertight shell that correctly represents the shape of the input building model.

Figure 12 a) shows the tessellation result of the previously shown model in Figure 1. This model is a typical imperfect building model generated in a manual process. It contains a gap on one of its facade, several missing surfaces (holes) at the bottom and on the roof and the facades, and many intersections between building parts. In Figure 12 b), all the intersecting parts with the main body, such as the attached building installations on the facades and the roof constructions are all decomposed and triangulated. The non-manifold and degenerate situations are also eliminated. The result of this step is a complex, free of intersection. In order to ease the repair processes, coplanar merging is employed to simplify the result tessellation. We then conduct tetrahedralization and use all the triangles of the decomposed model and its convex hull as constraints as shown in Figure 12 c). The heuristic carving process gradually shrinks the convex hull surface and wraps the correct shape of the building (Figure 12 d)). We then extracted the final result as shown in Figure 12 e). This experiment shows that a model with serious errors can be automatically repaired with our proposed method.

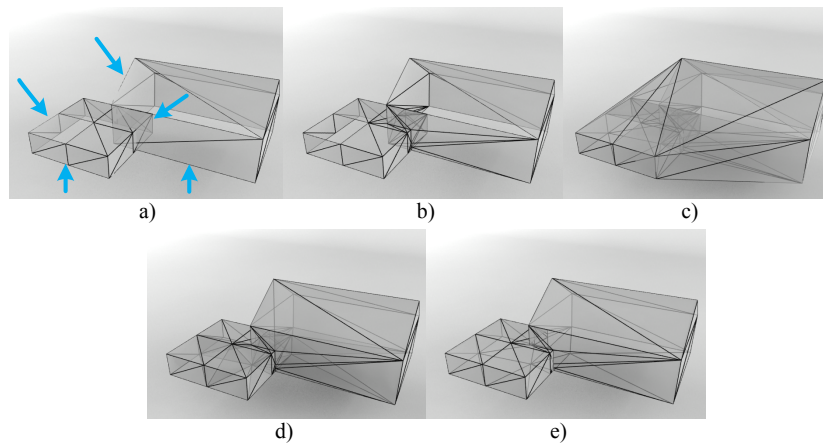


Figure 11. The experimental result of a flawed building model (a) the tessellated input model with intersecting building parts and several missing surfaces; b) the decomposition result of the model; c) the convex hull of the input model and itself are sent to the tetrahedralization; d) the carving result; e) the extracted exterior surface, which is a watertight valid model )

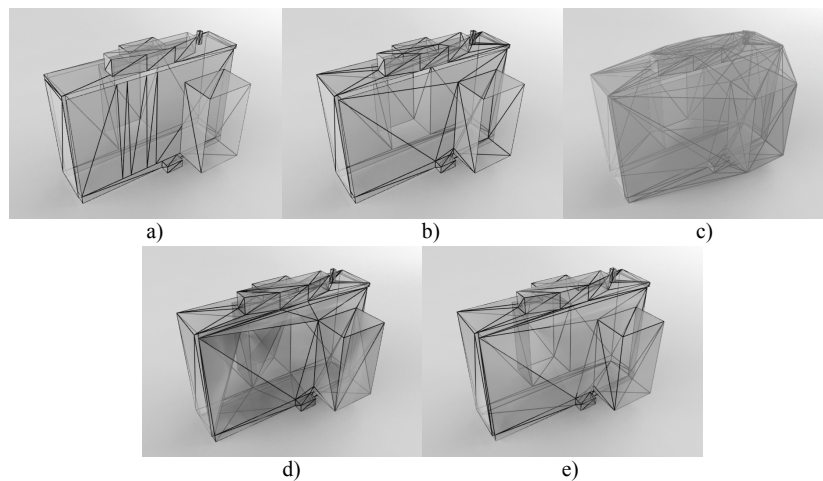


Figure 12. The experimental results of the models shown in Figure 1 (a) the tessellation of the input model; b) the result of decomposition and coplanar triangle merge; c) the initial approximation; d) the result after heuristic carving, with boundary tetrahedra visible; e) the final exterior surface extracted from d))

## 6. DISCUSSION AND FUTURE WORK

Two experiments prove that the proposed repair method is well capable of solving common occurring errors of a 3D building model, e.g. holes, non-manifold primitive, intersections etc. The validity constraints introduced well protect the manifoldness of the approximation during the shrinking process and the shape constraint preserves the missing planar area of a building model. In addition, the valid input model parts are well preserved after repair.

By using a global top down shrink-wrapping process, the method is robust and tolerates arbitrary input model. Experiments on different kinds of models show that the proposed heuristics based on the DoF, the Cp and the volume of a tetrahedron is effective. However, in some extreme cases, such as an unstructured polygon soup with probable nonzero genus, we found the manifoldness of the result might be broken because of the order of carving in a local area. This should be prevented by enhancing the carving operator. Furthermore, the third heuristic parameter, volume, can be replaced by other factors such as the area of the candidate triangle or the shape of a tetrahedron which will also be studied in the future.

Rounding-error is one of the obstacles we encountered during the current implementation. We did observe some unexpected decomposition results when a triangle is decomposed too many times. Therefore, a threshold is employed to determine the duplicated primitives and truncate the coordinates in each carving step. However, we plan to use the exact predicates to make the method more robust (Richard Shewchuk, 1997).

Because our method preserves the input tessellation, the semantics attached with geometry can be held and be propagated to the output model. This will also be implemented in the future. If the input building model is of a higher LoD, e.g. a LoD 3 or LoD 4 model, our method can also be used as a generalization mean to extract the LoD2 model automatically.

## ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (41201379), the Dutch Technology Foundation STW, which is part of the Netherlands Organization for Scientific Research (NWO) and partly funded by the Ministry of Economic Affairs, Agriculture and Innovation. (Project code: 11300)



## REFERENCES

- Bischoff, S. and Kobbelt, L., 2005. Structure preserving cad model repair. *Computer Graphics Forum*, 24(3): 527-536.
- Bogdahn, J. and Coors, V., 2010. Towards an automated healing of 3D urban models. In: *Int. Archives of Photogrammetry, Remote Sensing and Spatial Information Science, Vol XXXVIII-4/W15*, Shaker Verlag, pp. 13-17.
- Botsch, M. et al., 2007. Geometric modeling based on polygonal meshes. In: *ACM SIGGRAPH 2007 courses*, ACM New York, NY, USA.
- Campen, M. and Kobbelt, L., 2010. Exact and robust (self-) intersections for polygonal meshes. *Computer Graphics Forum*, 29: 397-406.
- Campen, M., Attene, M. and Kobbelt, L., 2012. A practical guide to polygon mesh repairing, The Eurographics Association, pp. 14-undefined.
- Granados, M. et al., 2003. Boolean operations on 3D selective nef complexes: data structure, algorithms, and implementation. In: *Lecture Notes in Computer Science*, Springer, 2832, pp. 174 - 186.
- Gröger, G. and Plümer, L., 2009. How to achieve consistency for 3D city models. *Geoinformatica*, (15): 137-165.
- Gröger, G., Kolbe, T.H., Nagel, C. and Häfele, K., 2012. OpenGIS city geography markup language (CityGML) encoding standard. Version 2.0.0. O. Open Geospatial Consortium, OGC 12-019.
- Guéziec, A., Taubin, G., Lazarus, F. and Hom, B., 2001. Cutting and stitching: converting sets of polygons to manifold surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 7(2): 136-151.
- Hagbi, N. and El-Sana, J., 2010. Carving for topology simplification of polygonal meshes. *Computer-Aided Design*, 42(1): 67-75.
- Herring, J.R., 2005. ISO 19107:2005: geographic information-spatial schema. O.G. Consortium, 19107.
- Karki, S., Thompson, R. and McDougall, K., 2010. Data validation in 3D cadastre. In: T. Neutens and P. Maeyer (eds), *Developments in 3D Geo-Information Sciences. Lecture Notes in Geoinformation and Cartography*. Springer Berlin Heidelberg, pp. 92-122.
- Kazar, B.M., Kothuri, R., Oosterom, P. and Ravada, S., 2008. On valid and invalid three-dimensional geometries, *Advances in 3D Geoinformation Systems. Lecture Notes in Geoinformation and Cartography*. Springer Berlin Heidelberg, pp. 19-46.
- Kobbelt, L.P., Vorsatz, J. and Labsik, U., 1999. A shrink wrapping approach to remeshing polygonal surfaces. *Computer Graphics Forum*, 18(3): 119-130.
- Kolbe, T.H., 2008. Representing and exchanging 3D city models with CityGML. In: S. Zlatanova and J. Lee (eds), *3D Geo-Information Sciences. Lecture Notes in Geoinformation and Cartography*. Springer Berlin Heidelberg, pp. 15-31.
- Kolbe, T.H., Groger, G. and Plümer, L., 2008. Citygml-3D city models and their potential for emergency response. In: *Geospatial Information Technology for Emergency Response*, Taylor & Francis, London, pp. 257 - 274.
- Koo, B.K., Choi, Y.K., Chu, C.W., Kim, J.C. and Choi, B.T., 2005. Shrink-wrapped boundary face algorithm for mesh reconstruction from unorganized points. *ETRI journal*, 27(2): 235-238.
- Ledoux, H., Ohori, K.A. and Meijers, M., 2012. Automatically repairing invalid polygons with a constrained triangulation. *Proceedings of Agile 2012*, Avignon, France.
- Ledoux, H., Verbree, E. and Si, H., 2009. Geometric validation of GML solids with the constrained delaunay tetrahedralization. *Proceedings of the 4th International Workshop on 3D Geo-Information*, Ghent, Belgium.
- Liepa, P., 2003. Filling holes in meshes. *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, Aachen, Germany, pp. 200-205.
- McKenney, D., 1998. Model quality: the key to CAD/CAM/CAE interoperability, International TechneGroup Incorporated, Milford, OH.
- Mezentsev, A.A. and Woehler, T., 1999. Methods and algorithms of automated cad repair for incremental surface meshing. *Proceedings 8th International Meshing Roundtable*, pp. 299-309.
- Möller, T., 1997. A fast triangle-triangle intersection test. *Journal of graphics tools*, 2(2): 25-30.
- Nooruddin, F.S. and Turk, G., 2003. Simplification and repair of polygonal models using volumetric techniques. *IEEE Transactions on Visualization and Computer Graphics*, 9(2): 191-205.
- Richard Shewchuk, J., 1997. Adaptive precision floating-point arithmetic and fast robust geometric predicates. *Discrete & Computational Geometry*, 18(3): 305-363.
- Shen, C., O'Brien, J.F. and Shewchuk, J.R., 2004. Interpolating and approximating implicit surfaces from polygon soup. *ACM Transactions on Graphics (TOG)*, 23(3): 896-904.
- Si, H., 2006. Tetgen: a quality tetrahedral mesh generator and three-dimensional delaunay triangulator. *Weierstrass Institute for Applied Analysis and Stochastic*, Berlin, Germany.
- Wagner, D. et al., 2012. Geometric-semantical consistency validation of CityGML models. *3D GeoInfo Conference 2012*, Quebec City, Canada, May 16-17.