

Article

Repair of Voids in Multi-Labeled Triangular Mesh

Deyun Zhong ^{1,2,3}, Benyu Li ^{1,2}, Tiandong Shi ^{1,2}, Zhaopeng Li ^{1,2}, Liguan Wang ^{1,2,3} and Lin Bi ^{1,2,3,*}

¹ School of Resources and Safety Engineering, Central South University, Changsha 410083, China; deyizhiyun@csu.edu.cn (D.Z.); 195512077@csu.edu.cn (B.L.); 195511033@csu.edu.cn (T.S.); whpeng2020@csu.edu.cn (Z.L.); liguan_wang@csu.edu.cn (L.W.)

² Research Center of Digital Mine, Central South University, Changsha 410083, China

³ Changsha Digital Mine Co., Ltd., Changsha 410221, China

* Correspondence: mr.bilin@csu.edu.cn

Abstract: In this paper, we propose a novel mesh repairing method for repairing voids from several meshes to ensure a desired topological correctness. The input to our method is several closed and manifold meshes without labels. The basic idea of the method is to search for and repair voids based on a multi-labeled mesh data structure and the idea of graph theory. We propose the judgment rules of voids between the input meshes and the method of void repairing based on the specified model priorities. It consists of three steps: (a) converting the input meshes into a multi-labeled mesh; (b) searching for quasi-voids using the breadth-first searching algorithm and determining true voids via the judgment rules of voids; (c) repairing voids by modifying mesh labels. The method can repair the voids accurately and only few invalid triangular facets are removed. In general, the method can repair meshes with one hundred thousand facets in approximately one second on very modest hardware. Moreover, it can be easily extended to process large-scale polygon models with millions of polygons. The experimental results of several data sets show the reliability and performance of the void repairing method based on the multi-labeled triangular mesh.



Citation: Zhong, D.; Li, B.; Shi, T.; Li, Z.; Wang, L.; Bi, L. Repair of Voids in Multi-Labeled Triangular Mesh. *Appl. Sci.* **2021**, *11*, 9275. <https://doi.org/10.3390/app11199275>

Academic Editor: Yosoon Choi

Received: 11 August 2021

Accepted: 3 October 2021

Published: 6 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: mesh repairing; geometry processing; 3D modeling; geological modeling; polygon mesh processing

1. Introduction

Polygonal representations of three-dimensional geometry modeling are widely used in computer graphics and practical applications [1]. However, due to limitations of the modeling methods, the models with multiple meshes often cannot be directly utilized by applications that require manifold meshes (topologically correct, e.g., free of singular edges and intersections) without voids as input. In this paper, based on several polygonal meshes without labels, we consider studying the representation method of the multi-labeled mesh and the model repairing method according to specific rules.

Despite the extensive research on mesh repairing, few works have addressed the topological correctness of the meshing repairing involving several input meshes. To describe clearly, the mesh repairing involving single material is called as single-domain model repairing, the mesh repairing involving multiple materials is called as multi-domain model repairing. Compared with the single-domain model repairing, the multi-domain model repairing not only needs to consider the mesh quality of each sub-model, but also needs to consider the adjacent rules among each sub-model according to the practical applications. In this paper, we mainly focus on the multi-domain model repair method used to deal with the voids between models caused by insufficient data sampling or modeling algorithm defects. For example, when each sub-model representing different materials is modeled separately, there may be voids overlaps and other unexpected defects between sub-models in the modeling results. For a geological body with specific geological interfaces, the voids between geological interfaces, which may be regarded as ‘new’ geological interfaces,

are undesired defects in geological modeling. In implicit geological modeling [2,3], it is easy to determine a point is inside or outside the implicit model by computing the sign of the function value. However, for a void combined by several implicit functions, it is difficult to determine whether an exterior point is in the void. We consider repairing the voids between several combined surface meshes, which is useful for the methods of meshing multiple domains.

Different from the single-domain model repairing method, which focuses on the quality of the mesh, the multi-domain model repairing method is mainly concerned with the topology adjacent relationship and the corresponding adjacent rules among each sub-domain model. Considering that each sub-mesh model can be repaired by the traditional single-domain model repairing method, we assume that the input meshes of the multi-domain model repairing method satisfy the characteristics of closed manifold, and the triangular topologies of the boundaries of the meshes are completely consistent. Based on the assumption, we can convert the multiple input meshes into the multi-labeled mesh which is convenient for searching the topology adjacent relationship among sub-meshes. It is worth noting that the mesh topology should be adjusted through preprocessing steps when the topology of overlapped triangular facet between multiple input meshes is inconsistent.

In this paper, we seek a robust method for repairing the voids existing in multiple triangular meshes, so that the meshes are topologically consistent (without undesired material, namely the interior voids). Similar to the basic idea of the model repairing method based on multi-labeled voxels, the model repairing method based on multi-labeled mesh ensures the topological consistency between polygonal meshes by the strategy of greedy searching. Different from the multi-domain model repairing method based on multi-labeled voxels [4], the novel method repairs the polygonal mesh directly. The former repairs the model by searching the voids in the multi-domain model represented as labels, and the latter repairs the model by searching the voids in the multi-domain model represented as polygon mesh. Compared with the former, the latter has the advantage that it can repair the triangular mesh obtained by any modeling method.

In this paper, we propose a multi-domain model repair method based on multi-labeled mesh by defining a multi-labeled mesh data structure. The method takes several closed and manifold meshes without labels as input, repairs the voids and generates the output meshes without voids. Based on the traditional polygon mesh data structure, we design a multi-labeled mesh data structure, and provide a method of conversion between the single-label meshes and the multi-labeled mesh. Compared with the traditional polygon mesh data structure, the multi-labeled mesh data structure reduces the data storage because it shares some mesh vertexes and triangular facets. In addition, it is useful to search and process the adjacent relationship between different meshes. On the basis of the designed multi-labeled mesh, we propose the judgment rules of voids between the input meshes and the method of void repairing based on the specified model priorities. The method is guaranteed to produce closed and topologically consistent meshes for any valid input meshes. It can be applied to geological modeling, medical modeling and other related model repairing problems involving multi-labeled triangular mesh.

2. Related Works

Generally speaking, the mesh repairing started around the 1990s, when mesh technology was just taking off. For example, the paper [5] published by William and Lorensen is one of the important foundations of volume reconstruction. The work of Bohn [6] and Barequet [7] in the 1990s is a typical example in the early stage of mesh repairing.

The existing studies on model repair mainly focus on the single-domain model repairing methods, which are roughly divided into two categories: the voxel-based repairing methods and the surface-based repairing methods [8].

The voxel-based repairing methods [9–12] first convert the mesh surfaces into voxels, then repair the meshes in the voxel space, and then use polygon extraction technology

to convert the voxels into meshes. Davis et al. [13] first constructed a signed distance function, then applied a diffusion process to extend the function through the volume to repair holes. Ju [14] used an octree grid to reconstruct a closed surface by contouring. Nooruddin et al. [15] converted polygonal models to a volumetric representation and repaired the model through the conversion between the model and the volume domain. Bischoff [16] built an adaptive octree for the input model and then reconstructed the model through voxel topology simplification. In general, the voxel-based repairing methods have a good performance in efficiency, but some original detailed features may be lost between model conversions, and the repaired model may have more redundancy triangular patches compared with the original model.

The surface-based repairing methods [17–21] generally first identify the location of the holes, and then directly perform mesh repairing on the local areas. These kinds of methods are roughly divided into three categories. The first are the repairing algorithms based on interpolation. The interpolated meshes can be generated using simple polynomial functions [22], triangular B-splines [23] or radial basis functions [24]. This kind of algorithms is suitable for repairing simple holes, but not for holes with complicated situations with grooves and islands. The second are the algorithms based on triangulation. Liepa [25] and Attene [26] first triangulated the hole area, and then continuously adjusted the newly generated meshes through mesh refinement and smoothing to make the new meshes similar to the surrounding mesh density. Li et al. [27], Wang et al. [28] and Ngo [29] first segmented the complex holes into multiple sub-holes along the characteristic curve, and then repaired the complex holes by repairing these relatively smooth small holes. This kind of algorithms can quickly achieve mesh repair, but it is often difficult to obtain the detailed features of the model. The third are the template-based repairing methods [30–32]. The idea of this type of algorithms is to search for a patch similar to the feature of the hole in the template library or the model itself and realize the repair by copying and pasting. This type of algorithms consumes relatively high time, and it is often difficult to repair the meshes in complex situations.

The above studies are all single domain mesh repairing algorithms. Scholars in the field of multi-domain mesh repairing also put forward some ideas. Gao et al. [33] presented a scene reconstruction method. The method used RGB-D data to reconstruct a 3D scene mesh model, then segmented different types of surfaces and objects in the scene, and then deployed an adaptive hole-filling algorithm to recover the original meshes. Ramos et al. [34] proposed a method to repair non-manifold boundaries of segmented simplicial meshes, and developed algorithms about relabeling, point insertion and simulated annealing to reconstruct models.

3. Overview of the Method

The previous model repairing methods mainly focus on the mesh quality of a single mesh, and there are few studies on the topological adjacent relationship between models composed of different materials. In this paper, we define a multi-labeled mesh data structure to represent the topological adjacent relationship between the input meshes and repair the unexpected voids by modifying the labels of triangular facets in the meshes.

As an example, Figure 1 shows the repairing process and modeling effect of interior voids between two simple meshes. Among them, Figure 1a,b are two original input meshes that meet the characteristics of closed and manifold, and the black box in Figure 1c indicates an interior void between the two meshes. By defining the judgment rules of voids among multiple meshes, we can find the set of connected triangular facets that make up the void, and use the method based on model priorities to repair the void. When the priority of the red mesh is higher than the yellow mesh, the void will be filled by the yellow mesh with lower priority. In addition, the corresponding model repairing effect is shown in Figure 1d. When the priority of the yellow mesh is higher than the red mesh, the void will be filled by the red mesh with lower priority. In addition, the corresponding model repair effect is shown in Figure 1e.

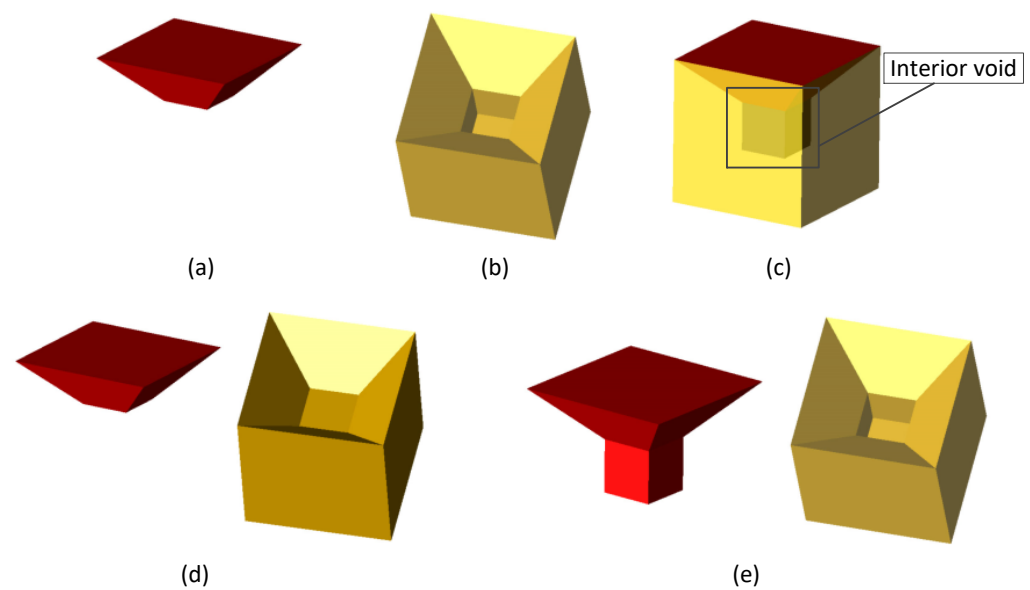


Figure 1. Repairing examples of the multi-labeled triangular mesh: (a,b) the input meshes; (c) an interior void combined by the input meshes; (d,e) the repaired results.

The method proposed in this paper is mainly composed of the following steps. Figure 2 shows the overall process of the method.

1. Convert the input unlabeled meshes into a multi-labeled mesh that is convenient to determine the adjacent relationship between meshes. Based on the idea of graph theory, the multi-labeled mesh is regarded as an undirected graph with different label attributes and the topology adjacent relationship can be obtained by traversing the triangular facets directly.
2. The original multi-labeled mesh can be separated into connected sub-graphs according to the connectivity of undirected graphs, and each connected sub-graph corresponds to a connected multi-labeled sub-mesh. Traversing each separated sub-graph, search for the quasi-voids using the breadth-first searching (BFS) algorithm and record the labels associated with corresponding quasi-voids.
3. Distinguish the true voids of the input meshes according to the judgment rules of voids and fill the voids by modifying the labels of triangular facets according to the specified model priorities. Remove the invalid triangular facets and obtain the repaired meshes.
4. Merge the repaired multi-labeled sub-meshes into a single multi-labeled mesh and convert the multi-labeled mesh into the unlabeled meshes.

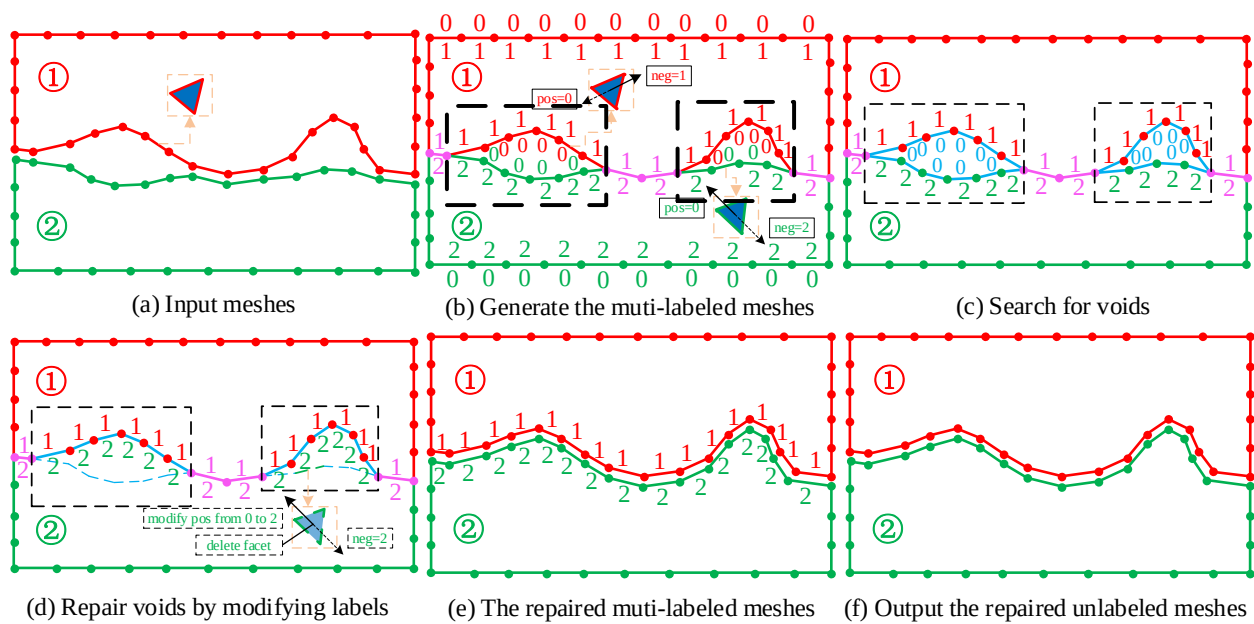


Figure 2. Overall flow chart of the method: (a) input meshes; (b) generate the multi-labeled meshes; (c) search for voids; (d) repair voids by modifying labels; (e) the repaired multi-labeled meshes; (f) output the repaired unlabeled meshes. The black boxes in (b) point to the voids between the input meshes, and the purple parts are overlapping regions.

4. Method

4.1. Generate Multi-Labeled Mesh

To deal with the voids between multiple meshes, the input meshes without labels are converted into multi-labeled meshes that can represent the topology relationship between meshes of different materials. Based on the original polygon mesh representation, we add a pos label and a neg label to represent the materials of the interior and exterior regions of the polygon mesh, respectively, as shown in Figure 3.

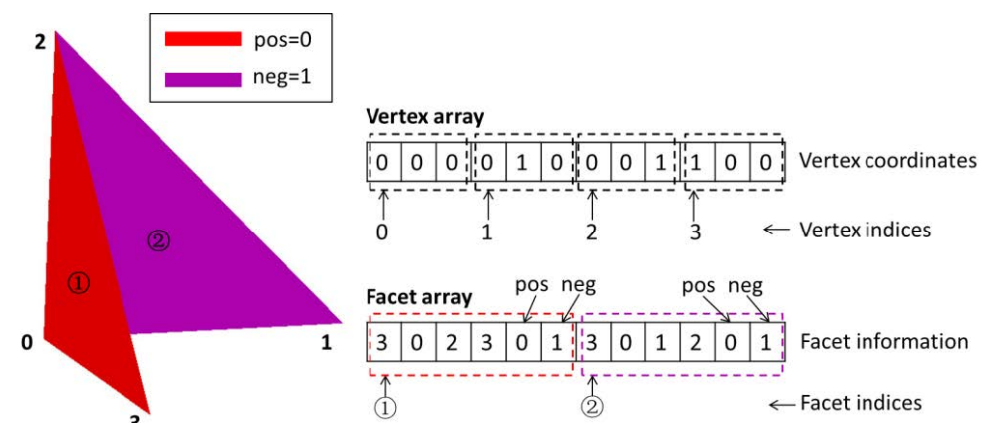


Figure 3. An example of a simple multi-labeled mesh represented by 2 arrays (vertex array and facet array). The vertex array stores the vertex indices and vertex coordinates. The facet array stores the facet information and the corresponding labels. Each facet is composed of the number of vertexes and the indices of vertexes. For example, facet ① is composed of 3 vertexes (the vertex indices are 0, 2 and 3); Facet ② is composed of 3 vertexes (the vertex indices are 0, 1 and 2).

The input mesh M_i without labels is added the pos and neg labels according to the interior and exterior directions of the model to form a single-labeled mesh SM_i . The input l meshes are assigned a label i ($i = 1, 2, \dots, l$), respectively, corresponding to the material in turn. The zero label represents exterior boundary of the mesh model. In addition, to

distinguish the materials corresponding to interior and exterior regions of the model, the interior and exterior sides of the triangular facet t are set to different labels. The interior region corresponding to the negative side of the triangular facet t (the region pointed in the opposite direction to the normal direction of the triangular facet) is represented as a neg label, and the label value is set to the label i of the corresponding material (namely $t_{neg} = i$). The exterior region corresponding to the positive side of the triangular facet t (the region pointed in the normal direction of the triangular facet) is represented as a pos label, and the label value is set to zero (namely $t_{pos} = 0$).

The multi-labeled mesh LM can be obtained by merging the single-labeled mesh $SM_i (i = 1, 2, \dots, l)$. As a simple example, a multi-labeled mesh generated by merging two simple single-labeled meshes is shown in Figure 4. In the process of merging, it is necessary to merge the overlapping mesh vertexes and the overlapping mesh facets, and the vertex indices of each triangle facet should be corrected according to the new point array in the multi-labeled mesh. If a point is close enough to another point within a given geometry accuracy, the two points are considered to overlap. To search for the overlapping mesh vertexes efficiently, the spatial searching methods, such as kd-tree, can be used to speed up the searching process. It is worth noting the specified geometry accuracy should be smaller than the distance of any two mesh vertices and bigger than the machine precision of vertex coordinates. In addition, the pos label or the neg label of the triangular facet t in the multi-labeled mesh should be modified according to the internal and external positions of the model, which ensures that the region corresponding to the interior and exterior of the triangular facet is set to the label i of the corresponding material ($t_{pos} = i$ or $t_{neg} = i$). It is worth noting that the topologies between the triangles in the intersecting and adjacent parts of the single-labeled meshes must be consistent, otherwise the merging process will cause errors. The pseudo-code of the algorithm flow for generating the multi-labeled mesh is shown in Algorithm 1 (Generate multi-labeled mesh).

Algorithm 1 Generate multi-labeled mesh.

Input: Meshes without labels $M_i (i = 1, 2, \dots, l)$
Output: Multi-labeled mesh LM
 Initialize an empty multi-labeled mesh LM .
foreach $M_i (i = 1, 2, \dots, l)$ **do**
 foreach $t \in M_i$ **do**
 Push unique vertexes of t to LM using a kd-tree data structure.
 if $t \in LM$ **then**
 $t_{pos} = i$
 else
 $t_{pos} = 0$
 $t_{neg} = i$
 Modify the vertex indexes of t via the new vertex array in LM .
 Push t to LM .
 Record t using an unordered set to store unique triangular facets.
 end if
 end for
end for

4.2. Judgment Rules of Voids

To distinguish different searching results, we call the set of connected triangular facets with zero labels as quasi-void. If the directions of the triangular facets are consistent, then pos and neg labels should be considered. In this case, the quasi-void is a set of connected triangular facets whose pos labels are zero.

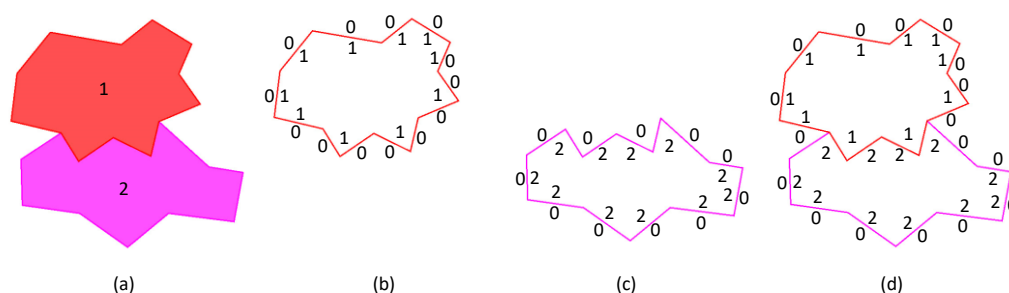


Figure 4. An example of a multi-labeled mesh combined by general meshes without labels: (a) two simple input meshes; (b) labels of mesh 1; (c) labels of mesh 2; (d) labels of the multi-labeled mesh.

According to the internal and external position relationship of the model between different materials, quasi-voids can be divided into three types: exterior boundary, interior boundary and void, as shown in Figure 5. Among them, the exterior boundary is a quasi-void adjacent to the external region of the multi-labeled mesh. The interior boundary and void are quasi-voids located inside the exterior boundary of the multi-labeled mesh. The difference is that the triangular facet set of the interior boundary contains only one type of non-zero label, while the triangular facet set of the void contains two or more non-zero labels, as shown in Figure 5. In the searching process, we can find all the quasi-voids. Then the judgment rules of voids are used to distinguish the types of quasi-voids, so that the true voids can be filled by the repairing method.

To distinguish the types of searched quasi-voids, we use the methods of comparing corresponding bounding boxes and comparing corresponding volumes to determine the true voids. For a connected multi-labeled mesh, the volume comparison method is based on the natural fact that the volume of the closed mesh that composes the interior boundary or the void is smaller than the volume of the mesh that composes the exterior boundary. To accelerate the speed of judgment, the bounding box comparison method is used to distinguish the quasi-voids. For a connected multi-labeled mesh, the bounding box comparison method is based on the similar fact that the bounding box (such as the rectangular bounding box) of the void or the interior boundary is contained in the bounding box of the exterior boundary. Therefore, when the bounding box of the searched triangular facets S_T is smaller than the bounding box of the exterior boundary, the triangular facets S_T is a void or an inner boundary. Then, whether S_T is a void can be directly determined by the number of non-zero labels in S_T . In special cases, when the bounding box of the searched triangular facets set S_T is exactly equal to the bounding box of the exterior boundary, it is necessary to use the volume comparison method to further determine whether S_T is a true void.

It is worth noting that the judgment rules of voids based on the bounding box comparison method and the volume comparison method are applicable to a connected multi-labeled mesh. For an unconnected multi-labeled mesh, the judgment rules of voids are no longer applicable. Therefore, to apply the judgment rules of voids, we should first separate the unconnected multi-labeled mesh into several connected sub-graphs based on the idea of graph theory, which ensures that the bounding box of the exterior boundary is the largest one for each connected sub-graph.

4.3. Search for Voids

The void searching stage aims to search for all the connected triangle facet sets that satisfy the void characteristics from the generated multi-labeled mesh LM . From the point of view of graph theory, each set of connected triangular facets can be regarded as a maximum connected sub-graph organized by the topological relationship of mesh vertexes and triangular facets.

Based on the idea of graph theory, a virtual graph $G(V, B)$ is defined by taking the mesh vertexes as the nodes V of the graph and taking the triangular facets as the edges B of the graph. To quickly search for quasi-voids, the topological adjacency relationship

between nodes and edges in the virtual graph should be constructed in advance. For the virtual graph $G(V, B)$, one node corresponds to multiple edges, and one edge corresponds to three nodes. Since the quasi-voids is a set of connected triangular facets in the multi-labeled mesh, the topological relationship of the triangular facets associated with each mesh vertex can be recorded directly by traversing the facet array.

As mentioned above, to accurately search for the voids, the original multi-labeled mesh LM should be separated into connected multi-labeled sub-meshes $LM_j (j = 1, 2, \dots, m)$, which is similar to the process of separating an unconnected graph into multiple connected sub-graphs. Based on the virtual graph $G(V, B)$, the BFS method of graph theory can be directly used to obtain the connected sub-graphs $G_j(V_j, B_j) (j = 1, 2, \dots, m)$. It is worth noting that as the vertex array of LM_j is different from the vertex array of LM , the new vertex indices should be recorded in the process of separating the sub-graphs to correct the vertex indices of the triangular facets in the multi-labeled sub-meshes.

As mentioned above, since the quasi-voids is a set of connected triangular facets with zero labels, the quasi-voids can be regarded as the connected sub-graphs in which the connectivity is judged based on the labels rather than the spatial positions in the virtual graph. Therefore, for each connected sub-graph $G_j(V_j, B_j)$, the improved BFS method can also be used to search for the quasi-voids S_T . Three mesh vertexes of a triangle facet are used as the initial nodes of BFS, and the vertexes of the mesh are breadth-first traversed according to the topological relationship of the virtual graph and the labels of the triangular facets. Finally, all sets of connected triangular facets S_T containing zero labels are obtained. The pseudo-code of the algorithm flow for searching for voids is shown in Algorithm 2 (Search for voids).

Algorithm 2 Search for voids.

Input: Multi-labeled mesh LM

Output: Voids $VO_{j,k} (j = 1, 2, \dots, m, k = 1, 2, \dots, n)$

Construct virtual graph $G(V, B)$ and topological relationship for LM .

Divide LM into connected multi-labeled sub-meshes $LM_j (j = 1, 2, \dots, m)$ using the BFS method.

foreach $LM_j (j = 1, 2, \dots, m)$ **do**

Construct virtual graph $G_j(V_j, B_j)$ and topological relationship for LM_j .

Initialize an array $Mark_t$ to record the visited state of t in LM_j .

foreach $t \in LM_j$ **do**

$Mark_t = 0$

end for

Initialize an array VO_j to store voids in $LM_j; k = 0$.

foreach $t \in LM_j$ **do**

if $Mark_t = 0$ **then**

if $t_{pos} = 0$ **then**

$k \leftarrow k + 1$

Initialize an empty void $VO_{j,k}$.

Initialize an empty triangular facet set S_T .

Push t to S_T .

Search for connected triangular facets T with zero label corresponding to t using the BFS method.

foreach $t' \in T$ **do**

Push t' to S_T .

$Mark_{t'} = 1$

end for

if S_T is a true void **then**

Push S_T to $VO_{j,k}$.

end if

end if

$Mark_t = 1$

end if

end for

end for

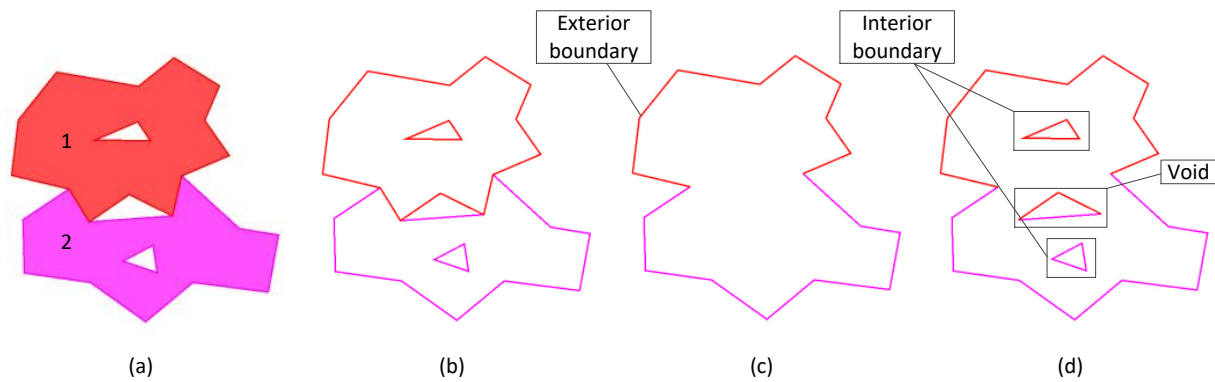


Figure 5. Three types of quasi-voids: interior boundary, exterior boundary and void: (a) two simple input meshes; (b) the boundaries of mesh 1 and mesh 2; (c) exterior boundary of the multi-labeled mesh; (d) interior boundary and void of the multi-labeled mesh.

4.4. Repair Voids

The void repairing stage aims to fill the sets of the searched connected triangular facets according to certain repair rules, and the purpose is to form a repaired multi-labeled mesh with consistent topological relationship and no voids.

To fill the triangular facets of voids, it is necessary to define the rules for repairing the voids. Similar to the model repairing method based on multi-labeled voxels, a simple way is to directly fill the voids by defining the model priorities that characterize the reliability of the model. Without loss of generality, the corresponding label is used to represent model priority. In this case, the model with a smaller label is more reliable. To obtain the repaired multi-labeled mesh, the zero labels of all the triangular facets in the voids are replaced with the corresponding max label in the triangular facets of the voids.

If the searched connected triangular facet set $S_{T,k}$ is a true void, then traverse the triangular facets and modify the zero labels as the max non-zero label in $S_{T,k}$. After filling the voids with the material corresponding to the highest priority of the model, the repaired multi-labeled sub-meshes $LM'_j (j = 1, 2, \dots, m)$ are obtained. It is worth noting that after the voids are filled, there may be some invalid triangular facets with the same pos label and neg label in the repaired meshes, as mentioned earlier. Therefore, a process should be performed to remove the invalid triangular facets.

To obtain a polygonal mesh model that is convenient for traditional 3D software rendering, the repaired multi-labeled mesh $LM'_j (j = 1, 2, \dots, m)$ should be converted into the triangular meshes M'_i corresponding to the original meshes M_i . The repaired multi-label sub-meshes $LM'_j (j = 1, 2, \dots, m)$ are merged to form the repaired multi-label mesh LM' . There are no overlapping triangular facets and mesh vertexes because each LM'_j is not connected. In addition, $LM'_j (j = 1, 2, \dots, m)$ can be merged directly by traversing the triangular facets. As mentioned earlier, since the vertex array of LM' is different from the vertex array of LM'_j , the new vertex indices of the triangular facets in LM' should also be corrected. After determining the labels $i (i = 1, 2, \dots, l)$ in the multi-labeled mesh, all triangular facets t with label satisfying $t_{pos} = i$ or $t_{neg} = i$ are extracted, respectively, to form single-labeled meshes $SM'_i (i = 1, 2, \dots, l)$. When extracting a single-labeled mesh from the multi-labeled mesh, it is also necessary to modify the vertex indices of the triangular facets according to the new vertex array. Finally, remove the labels of the triangular facets and consistent the orientations of triangular facets in $SM'_i (i = 1, 2, \dots, l)$ to obtain the repaired meshes without labels $M'_i (i = 1, 2, \dots, l)$. The pseudo-code of the algorithm flow for repairing voids is shown in Algorithm 3 (Repair voids).

end for

We implemented the mesh repairing algorithm in C++ and tested it on a Windows 64-bit PC with 3.80 GHz Intel(R) Core(TM) i7-10,700K and 32 GB RAM. To verify the reliability of our method, we tested the method on several geological datasets generated from geological sampling data using an implicit modeling method [4]. In addition, we compared the repairing results with the input meshes, as shown in Figures 6–9. Tables 1–4 show the parameters of the four experiments.

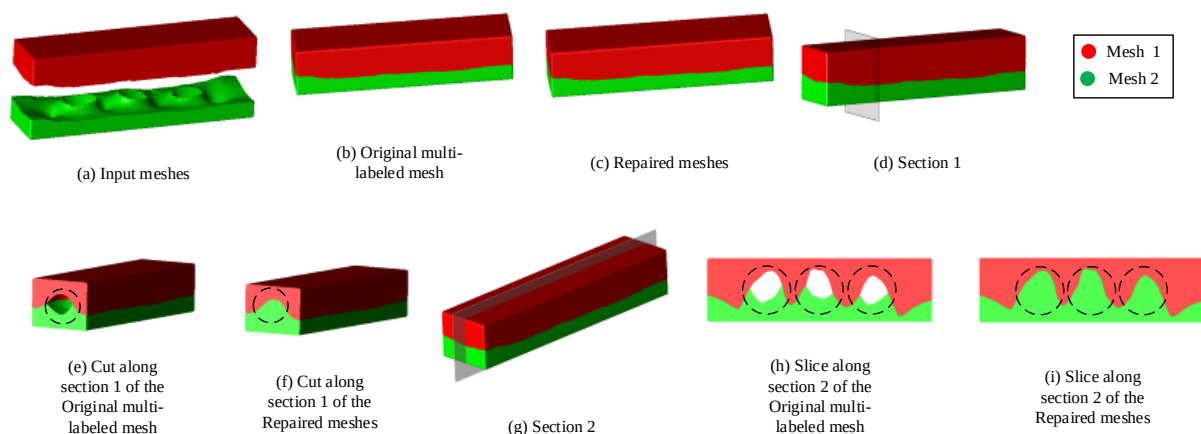


Table 1. Parameters of the multi-labeled mesh with input two sub-meshes. N_V is the number of voids, the Time is the time consumed in repairing the mesh, the Facets is the number of the triangular facets in multi-labeled mesh, the Vertexes is the number of mesh vertexes in multi-labeled mesh.

Meshes	Colors	N_V	Time (s)	Facets		Vertexes	
				Original	Repaired	Original	Repaired
Mesh 1	red	3	1.025	198,372	156,776	99,190	77,827
Mesh 2	green						



Meshes	Colors	N_V	Time (s)	Facets		Vertexes	
				Original	Repaired	Original	Repaired
Mesh 1	red	2	0.344	83,260	62,254	41,636	30,661
Mesh 2	blue						
Mesh 3	purple						

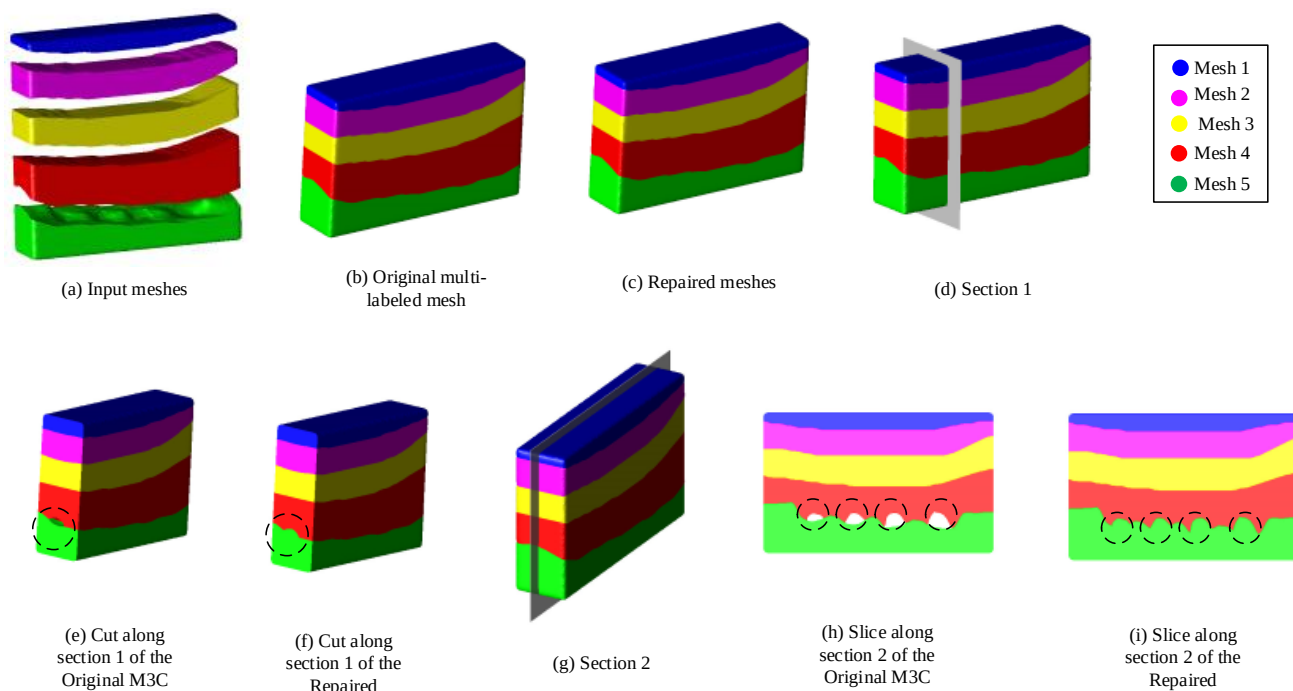


Figure 9. The multi-labeled mesh with input five sub-meshes.

Table 3. Parameters of the multi-labeled mesh with input four sub-meshes. N_V is the number of voids, the Time is the time consumed in repairing the mesh, the Facets is the number of the triangular facets in multi-labeled mesh, the Vertexes is the number of mesh vertexes in multi-labeled mesh.

Meshes	Colors	N_V	Time (s)	Facets		Vertexes	
				Original	Repaired	Original	Repaired
Mesh 1	red	4	0.337	83,406	60,458	41,711	29,670
Mesh 2	green						
Mesh 3	blue						
Mesh 4	purple						

Table 4. Parameters of the multi-labeled mesh with input five sub-meshes. N_V is the number of voids, the Time is the time consumed in repairing the mesh, the Facets is the number of the triangular facets in multi-labeled mesh, the Vertexes is the number of mesh vertexes in multi-labeled mesh.

Meshes	Colors	N_V	Time (s)	Facets		Vertexes	
				Original	Repaired	Original	Repaired
Mesh 1	blue	4	0.347	78,590	57,770	39,305	28,239
Mesh 2	purple						
Mesh 3	yellow						
Mesh 4	red						
Mesh 5	green						

Figure 6 shows an example of the repairing results using our method. Figure 6b is the original multi-labeled model generated from input two sub-meshes, and Figure 6c is the repaired multi-labeled model. Compared with Figure 6b,c, it can be seen that the repairing process will not cause apparent changes in the appearance of the model. It can be seen from Figure 6e,h that there are voids in the original multi-labeled model. Compared with Figure 6f,i, it can be seen that the voids inside the models have been repaired. As some invalid facets will be removed, it can be seen from the data in Table 1 that after the model is repaired, the number of facets will be reduced and the number of vertexes will also be reduced.

To test the universality of the algorithm, we implemented another three sets of experiments. Figures 7–9 shows the examples of the repairing results using the multi-labeled mesh generated from more sub-meshes. With the increase of the input meshes, the repairing algorithm can still repair the voids in the multi-labeled model.

The performance of the repairing method mainly depends on the number of facets and the meshes vertexes in the multi-labeled mesh, as shown in Tables 1–4. The experimental results of several datasets show that our method can repair the voids in multi-labeled mesh efficiently.

6. Conclusions and Discussion

The goal of our work is to design a simple algorithm for generating topologically consistent meshes for several valid input meshes. In this paper, we present a robust method for repairing voids in multi-labeled triangular mesh. Given several closed and manifold meshes without labels, our method always produces topologically consistent meshes without voids. The method is suitable for triangular meshes obtained by any three-dimensional modeling algorithm. The model repairing results are basically consistent with the original model, and only the labels of some triangular facets are removed or modified in the set of triangular facets where the voids are located. Therefore, the method retains the shape features of the original model to the maximum extent. The experimental results of several data sets show that the model repairing algorithm is efficient and reliable.

Even though we take the triangular mesh as an example to analyze the void repairing method of the multi-labeled mesh, the algorithm can be extended to the arbitrary polygon mesh by simple modification. It is worth noting that the algorithm proposed in this paper assumes that the input sub-mesh models are closed manifolds. When the input sub-mesh model is an unclosed manifold, the judgment rules of voids are not applicable anymore. Moreover, when the input sub-mesh model is non-manifold, the process of void searching may go wrong.

There are still some limitations to our work that need to be improved. The void is mainly determined according to the topological relationship of the triangular mesh and the adjacency relationship between the mesh models. In the process of practical application, it is found that the method requires that the quasi-voids are not adjacent, that is, there is no common point between any two quasi-voids. When there are common points between quasi-voids, the adjacent quasi-voids will be regarded as a void in the process of void searching, and errors will be made in the subsequent void repairing processes. Due to the simplicity of the repairing process, however, the repaired voids may not look optimal with respect to the surrounding meshes, particularly in the regions of large voids with several different labels. In this case, it is necessary to implement reliable method to fit better voids rather than replacing the labels of existing triangular facets in the future.

Author Contributions: D.Z., B.L., T.S. and Z.L. conceived, designed and performed the experiments; L.W. and L.B. analyzed the data and revised the methodology; D.Z. and B.L. wrote the paper; all authors discussed the results and revised the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the National Natural Science Foundation of China (52034001).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We thank the reviewers for their comments and suggestions to improve the quality of the paper.

Conflicts of Interest: The authors declare no competing interest.

References

- Attene, M.; Campen, M.; Kobbelt, L. Polygon mesh repairing: An application perspective. *ACM Comput. Surv.* **2013**, *45*, 15. [\[CrossRef\]](#)
- Calcagno, P.; Chilès, J.P.; Courrioux, G.; Guillen, A. Geological modelling from field data and geological knowledge: Part i. Modelling method coupling 3d potential-field interpolation and geological rules. *Phys. Earth Planet. Inter.* **2008**, *171*, 147–157. [\[CrossRef\]](#)
- Hillier, M.J.; Schetselaar, E.M.; de Kemp, E.A.; Perron, G. Three-dimensional modelling of geological surfaces using generalized interpolation with radial basis functions. *Math. Geol.* **2014**, *46*, 931–953.
- Li, B.; Zhong, D.; Wang, L. Repair of geological models based on multiple material marching cubes. *Mathematics* **2021**, *9*, 2207. [\[CrossRef\]](#)
- Lorensen, W.E.; Cline, H.E. Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH Comput. Graph.* **1987**, *21*, 163–169. [\[CrossRef\]](#)
- Bohn, J. Removing zero-volume parts from cad models for layered manufacturing. *Comput. Graph. Appl. IEEE* **1995**, *15*, 27–34. [\[CrossRef\]](#)
- Barequet, G.; Sharir, M. Filling gaps in the boundary of a polyhedron. *Comput. Aided Geom. Des.* **1995**, *12*, 207–229. [\[CrossRef\]](#)
- Pérez, E.; Salamanca, S.; Merchán, P.; Adán, A. A comparison of hole-filling methods in 3d. *Int. J. Appl. Math. Comput. Sci.* **2016**, *26*, 885–903. [\[CrossRef\]](#)
- Casciola, G.; Lazzaro, D.; Montefusco, L.B.; Morigi, S. Fast surface reconstruction and hole filling using radial basis functions. *Numer. Algorithms* **2005**, *39*, 289–305. [\[CrossRef\]](#)
- Franchini, E.; Morigi, S.; Sgallari, F. Implicit shape reconstruction of unorganized points using pde-based deformable 3d manifolds. *Numer. Math. Theory Methods Appl.* **2010**, *3*, 405–430. [\[CrossRef\]](#)
- Paulsen, R.R.; Baerentzen, J.A.; Larsen, R. Markov random field surface reconstruction. *IEEE Trans. Vis. Comput. Graph.* **2010**, *16*, 636–646. [\[CrossRef\]](#) [\[PubMed\]](#)
- Xie, H.; McDonnell, K.T.; Qin, H. Surface reconstruction of noisy and defective data sets. In Proceedings of the Conference on Visualization '04, Austin, TX, USA, 10–15 October 2004.
- Davis, J.; Marschner, S.R.; Garr, M.; Levoy, M. Filling holes in complex surfaces using volumetric diffusion. In Proceedings of the First International Symposium on 3d Data Processing, Padua, Italy, 19–21 June 2002.
- Ju, T. Robust repair of polygonal models. *ACM Trans. Graph.* **2004**, *23*, 888–895. [\[CrossRef\]](#)
- Nooruddin, F.S.; Turk, G. Simplification and repair of polygonal models using volumetric techniques. *IEEE Trans. Vis. Comput. Graph.* **2003**, *9*, 191–205. [\[CrossRef\]](#)
- Bischoff, S.; Kobbelt, L. *Structure Preserving Cad Model Repair*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2010; pp. 527–536.
- Chalmovianský, P.; Jüttler, B. Filling holes in point clouds. In *Mathematics of Surfaces, Proceedings of the 10th IMA International Conference, Leeds, UK, 15–17 September 2003*; Springer: Berlin/Heidelberg, Germany, 2013.
- Hoppe, H.; Deroose, T.; Duchamp, T.; McDonald, J.; Stuetzle, W. Surface reconstruction from unorganized points. *ACM SIGGRAPH Comput. Graph.* **1992**, *26*, 71–78. [\[CrossRef\]](#)
- Wang, J.; Oliveira, M.M. A hole-filling strategy for reconstruction of smooth surfaces in range images. In Proceedings of the Brazilian Symposium on Computer Graphics & Image Processing, Sao Carlos, Brazil, 12–15 October 2003.
- Wu, X.; Chen, W. A scattered point set hole-filling method based on boundary extension and convergence. In Proceedings of the 11th World Congress on Intelligent Control and Automation, Shenyang, China, 29 June–4 July 2015.
- Doria, D.; Radke, R.J. Filling large holes in lidar data by inpainting depth gradients. In Proceedings of the IEEE Computer Society Conference on Computer Vision & Pattern Recognition Workshops, Providence, RI, USA, 16–21 June 2012.
- Wang, J.; Oliveira, M.M. Filling holes on locally smooth surfaces reconstructed from point clouds-sciencedirect. *Image Vis. Comput.* **2007**, *25*, 103–113. [\[CrossRef\]](#)
- Pfeifle, R.; Seidel, H.P. Triangular b-splines for blending & filling of polygonal holes. In Proceedings of the Graphics Interface 1996 Conference, Toronto, ON, Canada, 22–24 May 1996.
- Branch, J.W.; Prieto, F.; Boulanger, P. Automatic hole-filling of triangular meshes using local radial basis function. In Proceedings of the International Symposium on 3d Data Processing, Chapel Hill, NC, USA, 14–16 June 2006; pp. 97–111.
- Liepa, P. Filling holes in meshes. In Proceedings of the First Eurographics Symposium on Geometry Processing, Aachen, Germany, 23–25 June 2003.
- Attene, M. A lightweight approach to repairing digitized polygon meshes. *Vis. Comput.* **2010**, *26*, 1393–1406. [\[CrossRef\]](#)
- Li, Z.; Meek, D.S.; Walton, D.J. Polynomial blending in a mesh hole-filling application. *Comput. Aided Des.* **2010**, *42*, 340–349. [\[CrossRef\]](#)
- Wang, X.C.; Liu, X.P.; Lu, L.F.; Li, B.J.; Cao, J.J.; Yin, B.C.; Shi, X.Q. Automatic hole-filling of cad models with feature-preserving. *Comput. Graph.-UK* **2012**, *36*, 101–110. [\[CrossRef\]](#)
- Ngo HT, M.; Lee, W.S. Feature-first hole filling strategy for 3d meshes. In *International Conference on Computer Vision, Imaging and Computer Graphics, Proceedings of the International Joint Conference, VISIGRAPP 2011, Vilamoura, Portugal, 5–7 March 2011*; Springer: Berlin/Heidelberg, Germany, 2011.
- Kraevoy, V.; Sheffer, A. Template-based mesh completion. In *Eurographics Symposium on Geometry Processing*; Citeseer: Princeton, NJ, USA, 2005.

31. Park, S.Y.; Guo, X.; Shin, H.Y.; Qin, H. Shape and appearance repair for incomplete point surfaces. *Korean J. Comput. Des. Eng.* **2007**, *12*, 330–343.
32. Harary, G.; Tal, A.; Grinspun, E. Context-based coherent surface completion. *ACM Trans. Graph.* **2014**, *33*, 1–12. [[CrossRef](#)]
33. Gao, Y.; Yao, Y.; Jiang, Y. Multi-target 3d reconstruction from rgb-d data. In Proceedings of the 2nd International Conference on Computer Science and Software Engineering, Xi'an, China, 24–26 May 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 184–191.
34. Ramos, T.L.C.; Vargas, A.J.C. Repairing non-manifold boundaries of segmented simplicial meshes. In Proceedings of the 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Niteroi, Brazil, 17–20 October 2017; pp. 39–46.