

Structure-Preserving Mesh Simplification

Zhuo Chen, Xiaobin Zheng and Tao Guan*

School of Computer Science and Technology, Huazhong University of Science and Technology
[e-mail: cz_007@hust.edu.cn]

*Corresponding author: Tao Guan

*Received May 12, 2020; revised July 12, 2020; revised September 4, 2020; accepted November 6, 2020;
published November 30, 2020*

Abstract

Mesh model generated from 3D reconstruction usually comes with lots of noise, which challenges the performance and robustness of mesh simplification approaches. To overcome this problem, we present a novel method for mesh simplification which could preserve structure and improve the accuracy. Our algorithm considers both the planar structures and linear features. In the preprocessing step, it automatically detects a set of planar structures through an iterative diffusion approach based on Region Seed Growing algorithm; then robust linear features of the mesh model are extracted by exploiting image information and planar structures jointly; finally we simplify the mesh model with plane constraint QEM and linear feature preserving strategies. The proposed method can overcome the known problem that current simplification methods usually degrade the structural characteristics, especially when the decimation is extreme. Our experimental results demonstrate that the proposed method, compared to other simplification algorithms, can effectively improve the quality of mesh and yield an increased robustness on noisy input mesh.

Keywords: mesh simplification, mesh regularization, 3D reconstruction, 3D geometric

1. Introduction

Recovering 3D information from several images has always been a challenging task in geometric computer vision. Various multi-view 3D reconstruction methods have been proposed to generate high-resolution mesh models in last decades. However, plagued by some unavoidable issues, such as inaccurate camera geometry or object occlusion, their outputs usually come with a lot of distorted structures and incomplete geometry. On the other hand, considering the needs of 3D mesh storage, transmission, processing, and rendering, producing a concise, yet geometrically faithful model is also a necessary work. Although there are a wide range of simplification algorithms, degeneration of structure is still a common issue which remains to be solved. Motivated by these two problems, a structure-preserving mesh simplification method which can output a clean and compact mesh is desired.

There are a number of mesh simplification algorithms to keep the mesh model from degeneration during the decimation process [1-7]. The most common approach is *Edge Collapse* algorithm which minimizes the local geometric error brought by the decimation operation. It's not sufficient to consider local geometric error metric only, therefore global error controlling algorithms [8-10] are proposed. However, both the local and global error do not imply structure preservation. With the development in the field of feature detection [11-15], some researchers focus on utilizing additional constraints to guide the simplification. To better keep the structure characteristics, feature-preserving simplification methods [16-18] are proposed in these years. Most of them utilize the constraints of planar proxies or curvature to preserve the mesh structure from distorting, however, ignoring the linear features in the scenes.

It's known that both planes and lines are common geometric primitives and could be an effective abstract of 3D model. Especially, in large-scale urban scene, most of the objects are composed of planar parts. Moreover, line segments correspond to the edges of objects and contain important structural information that can be used to regularize the mesh and guide the simplification process. Inspired by this idea, our method consists of three steps: planar structure detection, structural line segments extraction and structure-preserving simplification, as shown in Fig. 1.

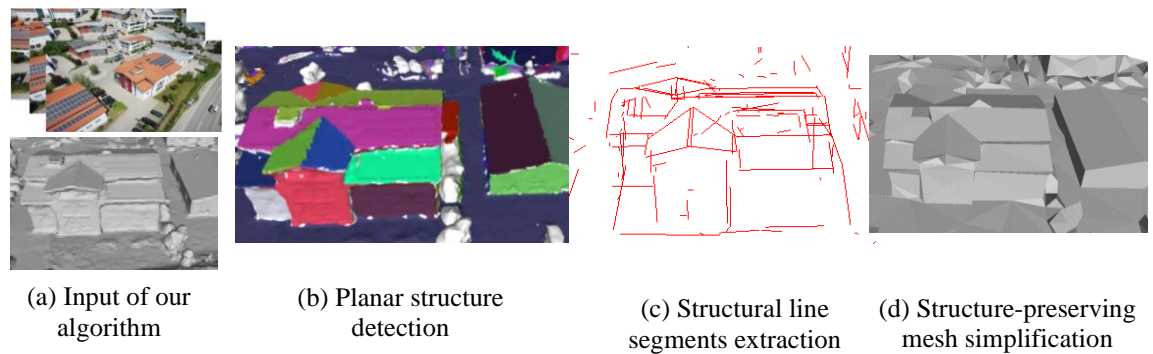


Fig. 1. Overview. (a) Input images and raw mesh. (b) Planar structure detection, which generates a set of planar structures using an iterative diffusion approach. (c) Structural line segments extraction, which filters out lots of noise and keeps the contour information. (d) Structure-preserving simplification with planar and linear feature constraints.

To be more specific, in the first step, the planar structure detection, our method generates a set of planar structures using an iterative diffusion approach based on *Region Seed Growing* algorithm. It grows iteratively the regions with fine-to-coarse constraints and finally merges planes which are near coplanar into the final set of planar structures. Next, a novel 3D line modeling method is applied to recover the 3D linear features from images and obtain a number of line segments. Compared to other methods, the proposed approach jointly exploits the extracted planar structure of the mesh to filter and improve these linear features, which could efficiently remove lots of noise and useless line segments that are inside bounded regions. After these works, we present a structure-preserving simplification approach that combines quadric error metric (QEM) minimization with the preservation of geometry, in which the planar and linear structure constraints are utilized.

In summary, we propose a novel method for mesh regularization and simplification. Our experiments on real scenes with rich linear features, such as urban buildings and furniture, illustrate the superiority of our method. For this paper, the main contributions are as follows:

1. An effective planar structure detecting approach, which generates a set of planar structures using an iterative diffusion strategy based on *Region Seed Growing* algorithm;
2. A novel structural line segments extracting algorithm which keeps the contour information of mesh and filters a lot of noise;
3. An improved structure-preserving simplification algorithm with planar and linear feature constraints.

2. Related Work

3D line modeling. There are various methods using line segments for reconstruction. For many built scenes composed of planes and line segments such as city buildings and tables, etc., 3D lines could be very useful. Micusik et al. [19] estimate motion and wiry 3D structure from imaged line segments across multiple views. SIFT keys placed at the end-points are exploited when doing image lines matching. Sugiura et al. [20] proposed a method that reconstructs surface from point-and-line cloud. They use Line Segment Detector (LSD) to detect 2D line segments, and then apply multiple-view correspondences computed by 2D line-to-line matching for pairs of images to reconstruct 3D line cloud. While these methods require explicit endpoint correspondences, Jain et al. [21] proposed a method that formulates the 3D lines modeling as an optimization problem, where the most probable 3D locations for the segment endpoints are computed by minimizing the re-projection error among several neighboring views. Although their method works well even under difficult lighting conditions, such an expensive-to-compute way is inefficient for large-scale datasets. Then Hofer et al. [22] replaced the continuous depth estimation with epipolar guided line segment matching which use purely geometric constraints and formulate the reconstruction procedure as a graph-clustering problem. These methods cannot directly be applied to our problem, but still provide useful ideas. In our case, instead of the segments for Structure-from-Motion(SfM) or reconstruction with lots of noise, what we need is only the contour line segments to regularize and clean our mesh.

Mesh regularization. Compared to refinement approaches, researches about improving regularity of the approximate reconstruction receive less interest in recent years. Most works focus on applying geometric primitives into reconstruction: Li et al. [23] recover a set of locally fitted basic primitives along with global mutual relations of objects, such as planes, cylinder, and sphere etc., and the global relations are iteratively learned. Lafarge et al. [24] further transform the various urban components interacting into a non-convex energy

minimization problem in which they are propagated under arrangement constraints over a planimetric map. There are some semi-automatic methods that usually need user assistance in image selecting and lines marking. Wang et al. [25] recently proposed an interactive approach that they construct a scaffold structure to regularize the mesh model, where users help to recognize the lines and the algorithm can use them to optimize the topology. However, the parameter-dependent primitive representation is not suitable for large-scale scenes and human interaction methods are not efficient.

Mesh Simplification. The research of simplification is fairly matured, and numerous mesh simplification algorithms have been previously proposed, such as Vertex Cluster [3, 5, 6], Vertex Removal [1], Surface Re-tiling [2], and the most common approach is the edge collapse with QEM [4]. However, for highly simplified mesh, minimizing a local geometric error metric is not sufficient. Therefore, a number of global error controlling algorithms [9, 10] are proposed to prevent the accumulation of approximation error. Regrettably, these methods didn't solve the problem of structure preservation. Therefore Salinas et al. [16] suggested explicitly detecting the structural characteristics to induce the structure-aware mesh decimation. They first extract a set of planar proxies and compute a graph to store the relation between planar structures. With structural information, constrained QEM computation and other strategies would produce a better result.

In the following, we will present our method in Section 3, which includes the planar structure detection, structural line segments extraction and our improved structure-preserving simplification algorithm with both planar and linear feature constraints. Finally, the experimental results are demonstrated in Section 4.

3. The Proposed Structure-Preserving Mesh Simplification

Both planes and lines are significant geometric primitives and could be an effective abstract of 3D model. In large-scale urban scene, most of the objects are composed of planar parts. Moreover, line segments correspond to the edges of objects and contain important structural information that can be used to regularize the mesh and guide the simplification process.

In this section, we present a novel simplification method, which exploits both the planar structure and linear features. Our method starts with planar structure detection (Section 3.1), which an iterative diffusion approach based on *Region Seed Growing* algorithm is demonstrated; the estimated planar structures are then used in the effective structural line segments extraction method (Section 3.2), which could recover the 3D lines and filter out incorrect and useless segments; finally, the proposed structure-preserving simplification algorithm (Section 3.3) considering both the planar and linear feature constraints will output the compact and clean mesh model.

3.1 Planar Structure Detection

Planar structure is one of the most important primitives in 3D scenes. Especially, in the general urban scenes, most of the objects are composed of planar parts. These planar structures can be extracted by common shape detection approaches. Considering the topological information of the mesh, we propose an iterative diffusion approach based on *Region Seed Growing* algorithm, which can segment the connected regions with the same planar features and provide good boundary information.

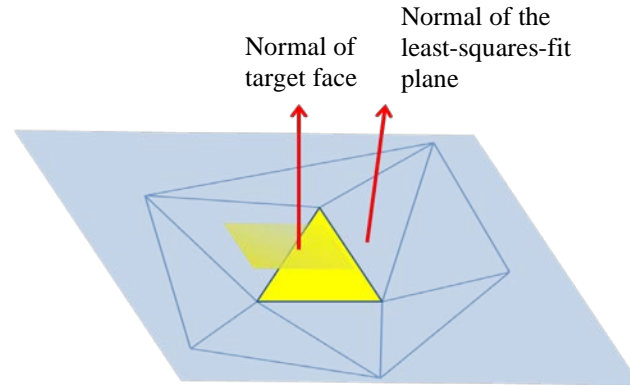


Fig. 2. Illustration of planarity score. It is measured by the angle between the normal of the specific face and the least-squares-fit plane of its neighboring faces.

At initial stage of the detection, a triangular face with the best planarity is selected as the seed plane during each growing, and then the region is grown over its adjacent triangles while keeping the error tolerance. To be more specific, the algorithm can be divided into three steps: seeds priority queue initialization, iterative region growing and the final merging.

Seeds Priority Queue Initialization. For Region Growing based algorithm, the choice of the region seed is a significant part. Therefore we establish the priority queue of the region seed, which determine the start face of each region growing. This priority queue should select the seed face that is most likely on a planar structure. For this goal, a score of planarity for every triangular face is computed as its priority. The planarity score is measured by the angle between the normal of the specific face and the least-squares-fit plane of its neighboring faces, as shown in **Fig. 2**. The smaller the angle, means the flatter this region and the higher the planarity score of this triangle as well as its priority. During each of the region growing process, we pick the top element in priority queue, which has the highest planarity score, as the seed to grow.

Iterative Region Growing. Our approach applies a fine-to-coarse constraint to guide the iterative region growing. In the growing with the finest constraint, the regions are growing under a normal and distance error tolerance, which are denoted as γ and d . To be more specific, a selected triangle will be added into current region only if its normal deviates less than γ and its projection distance to the seed is shorter than d , which filters out those triangular faces whose normal deviates too much or centroid is too far to the seed plane. When no more triangular face is eligible, this region stops growing; when the seeds priority queue is empty, this iteration ends. Note that if the area of the growing region is too small after an iteration (under a specified threshold), this growing is considered invalid. In the next iteration, regions grow with a coarser constraint and the growing skips faces which are already visited. For the iteration with the coarsest constraint, the regions are growing under only the normal error tolerance.

Planar Structures Merging. After the iterative diffusion, a set of planar structures are obtained. However, some of them are adjacent and near coplanar. For the simplicity of following processing, we conduct a final merging of the planar structures. First, these planar

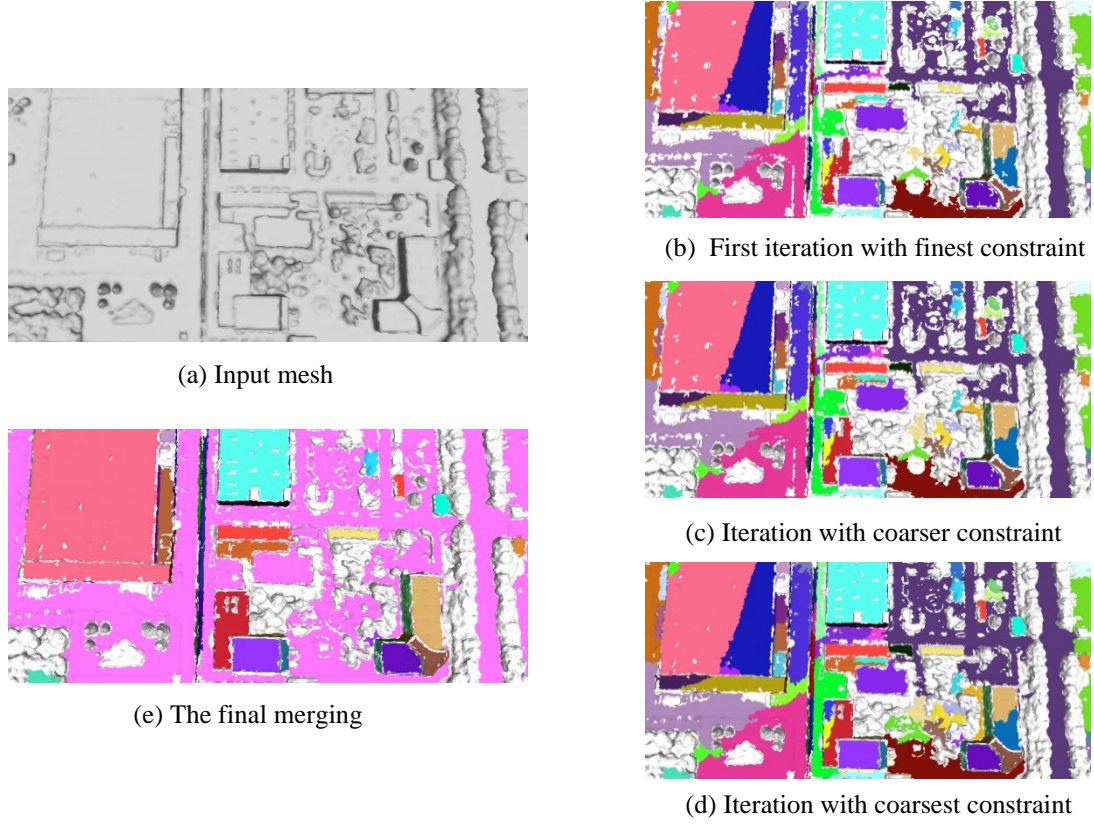


Fig. 3. The pipeline of our Planar Structure Detection.

structures are sorted by their areas and the similarity of their normal are computed as well. If the normal of target plane deviates less than the normal tolerance to the source planar structure, it will be merged into one planar structure. The merging is also an iterative diffusion process. **Fig. 3** demonstrates the entire process of planar structure detection.

The pseudo code shows the main processes of planar structure detection:

Pseudo code of Planar Structure Detection

```

1: function plane_detector( $M$ );
   Input: mesh model  $M$  with  $N$  faces
   Output: marked faces list
   // Seeds Priority Queue Initialization
2:   Let  $SPQ<index, score>$  be the Priority Queue
3:   for  $i = 1$  to  $N$  do
4:      $S \leftarrow$  compute the planarity score for  $face(i)$ 
5:     insert  $(i, S)$  into  $SPQ$ 
6:   end for
   // Iterative Region Growing
7:   Let  $PS$  be the set of Planar Structure
8:   for  $i = 1$  to 3 do
9:      $\gamma \leftarrow$  normal error tolerance ( $i$ )

```

```

10:    $d \leftarrow$  distance error tolerance (i)
11:   while  $SPQ$  is not empty do
12:      $Seed \leftarrow$  top of  $SPQ$ 
13:     pop  $SPQ$ 
14:      $P \leftarrow$  Region Growing for  $face(Seed)$  with  $(\gamma, d)$ 
15:     insert  $P$  into  $PS$ 
16:   end while
17: end for
    //Merge planar structures
18: sort  $PS$  with the area in descending order
19:  $\theta \leftarrow$  normal error tolerance of merging
20: for  $i = 1$  to  $N_{PS}$  do
21:   for  $j = i+1$  to  $N_{PS}$  do
22:     if normal of  $PS(i)$  deviates less  $\theta$ 
23:       merge  $PS(j)$  into  $PS(i)$ 
24:       mark  $PS(j)$  as  $PS()$  end forend for
25:     end if
26:   end for
27: end for

```

3.2 Structural Line Segments Extraction

With the help of estimated planar structures, our method could effectively extract structural line segments to guide following mesh decimation. It first recovers the 3D linear features from images (section 3.2.1) and obtains a number of 3D structural segments. Furthermore, compared to other methods, we jointly exploit the detected planar structure (section 3.2.2) to filter and improve these linear features, which could efficiently remove lots of noise and useless line segments that are inside bounded regions.

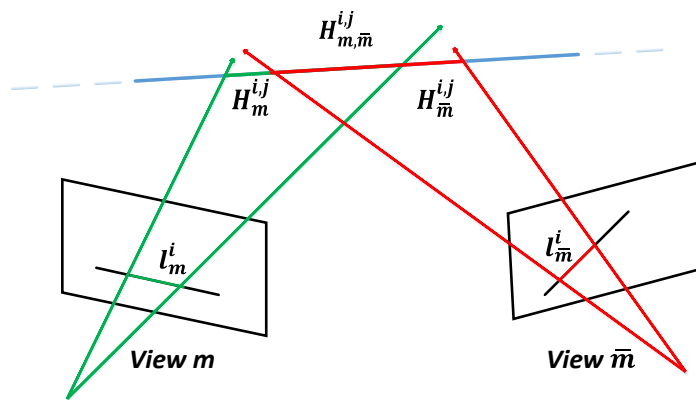


Fig. 4. Example of recovering 3D lines from matched 2D segments. The matched segment pair $(l_m^i, l_{\bar{m}}^j)$ in view m (the green) and view \bar{m} (the red) generates a 3D line $H_{m,\bar{m}}^{i,j}$, on which the two 3D line segment $H_m^{i,j}$ and $H_{\bar{m}}^{i,j}$ is corresponding to them respectively.

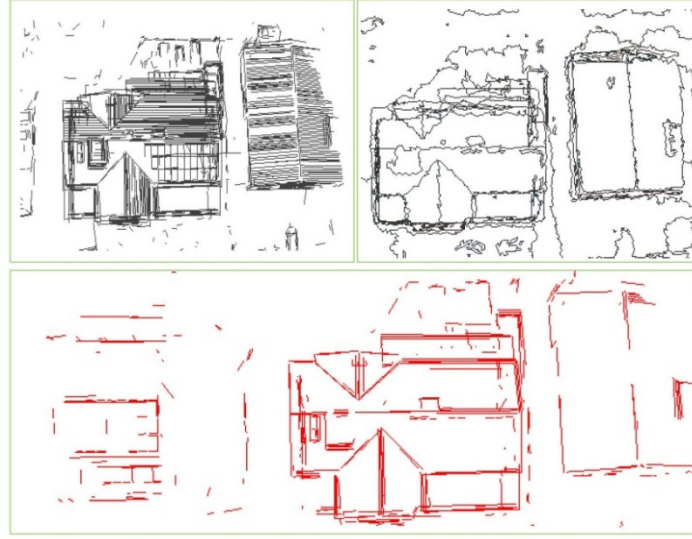


Fig. 5. Comparison of obtained 3D line segment set before and after our line filtering method. **Top Left:** 3D line modeling from only images. **Top Right:** The boundary of planar structure we extracted. **Bottom:** Structural line segments after our line filtering. Most of the noise and useless segments are filtered out, with the structural segments reserved.

3.2.1 3D Line Modeling

The input of this algorithm is a set of images $I = \{ I_1, I_2, \dots, I_N \}$, where N is the number of cameras. Then the corresponding camera calibration $P = \{ P_1, P_2, \dots, P_N \}$ and a set of sparse 3D points $X = \{ X_1, X_2, \dots, X_K \}$ can be obtained by arbitrary robust structure and motion algorithm.

In the first step, a set of 2D line segments $L_i = \{ l_1^i, l_2^i, \dots, l_M^i \}$ are detected for every input image I_i using Line Segment Detector (LSD) [26] which is a parameter-free algorithm that extracts 2D line segments robustly. With these line segments at hand, the most important work is to establish potential correspondences between them. In order to avoid numerous unnecessary computations, similar to [27], we first find the probable neighbors for each image instead of matching all images with each other. The Dice's similarity coefficient that Hofer et al. used is not sufficient and the proposed approach uses an improved version of computation to better estimate scores of the potential neighbors:

$$\text{Score}(i, j) = \frac{2 \cdot |X(i) \cap X(j)|}{|X(i)| + |X(j)|} \cdot \frac{\cos \angle(I_i, I_j)}{D(I_i, I_j)}, \quad (1)$$

where $\angle(I_i, I_j)$ denotes the angle between the view directions of I_i and I_j , D the difference between their scales.

Now it's time to match our 2D line segments between these neighbor images. This process follows the simple but effective way proposed in [22], which use the epipolar matching constraints for segment pairs. For line segment l_m^i in a selected segment pair (l_m^i, l_m^j) , its two end points are corresponding to two epipolar lines in another image I_j respectively. We then intersect the two lines with line passing through l_m^j , and overlap $O_{i,j}$ of l_m^j with the truncated

segment can be computed easily. $O_{j,i}$ is obtained in the same way. Thus, initial matches could be established as :

$$M = \left\{ (l_m^i, l_m^j) \mid \frac{\min(|O_{i,j}|, |O_{j,i}|)}{\max(|l_m^i|, |l_m^j|)} \geq \tau \right\}, \quad (2)$$

which two line segments are considered to be matched only if their relative overlap is large enough.

Each matched 2D line segment pair $\{l_m^i, l_m^j\}$ could generate a 3D line $H_m^{i,j}$ by intersecting two planes which pass through the segments and camera centers P_i, P_j , as Fig. 4 shows. The initial result contains lots of inaccurate and redundant segments, and every 2D segment is corresponding to many 3D segments. By using the image data, we re-project 3D segments to all the images, and evaluate the confidence with the angle and distance between the original 2D segment and the re-projected one. For each 2D segment, the 3D segment with the highest confidence is set as its spatial representation. Finally, 2D segments which correspond to the same entity need to be clustered as well as their 3D representations. So after filtering lines with 2D image information, 3D spatial information (angular and positional similarity) could be utilized to compute the affinity of matching segments [22]. Through graph-based clustering approach proposed by Felzenszwalb [28], we finally obtain an abstract 3D line model, as the top left of Fig. 5 shows.

3.2.2 Lines Filtering

The 3D line segment model generated by the image set contains a large number of mis-matching noise and useless line segments within planar regions. We need to carry out line filtering to obtain valid 3D line segments with structure features. Given the 3D line model and the planar structures extracted in the previous step, their spatial information could be utilized to keep only the segments on the boundary of planes and find anchor points on the mesh which correspond to the end points of segments.

Firstly, the correspondence between 3D segments and plane boundary should be established. For each 3D segment, we make radius neighbor search near its two end-points and mid-point and obtain a number of neighbor boundary vertices. The planar structure which has the most boundary vertices will be assigned to this segment. If the neighboring boundary vertices of the 3D segments are not sufficient, the line segment will be filtered out. To accelerate the nearest neighbor queries, a kd-tree is built with boundary vertices of planar structures. After this segment evaluating, most noise and invalid line segments inside planes are removed.

Further more, we use the mesh information to filter out redundant and imprecise 3D segments corresponding to the same entity, which is difficult to be recognized by [22, 27] due to the registration error of image data. The result 3D line segments contain different segments, some of which however depict the same boundary of the mesh planar structure. Therefore the similarity coefficient of two segments (s_i, s_j) can be computed as:

$$C(s_i, s_j) = \exp\left(\frac{\cos\angle(s_i, s_j)}{\sigma^2}\right) \cdot O(s_i, s_j), \quad (3)$$

where $\angle(s_i, s_j)$ denotes the angle between (s_i, s_j) , and O computes the overlap between them by projecting. σ is the user-specified regularization parameter. $C(s_i, s_j)$ is valid only when it is over 0.5. The two segments with high similarity coefficient will be considered redundant, and we keep only the one which passes through more mesh vertices.

After final filtering, our extracting algorithm is finished and produces a set of structural line segments $S = \{s_1, s_2, \dots, s_L\}$, where L is the number of segments, as the example in [Fig. 5](#) demonstrates. These lines correspond to the boundary of planar structures and could be used to regularize the mesh model and guide the structure-preserving simplification.

3.3 Structure-Preserving Mesh Simplification

To generate a concise model with a well-preserved structure, both the planar structures and line features are utilized to guide the mesh simplification. In this section, we present an improved mesh simplification method based on the edge collapse with QEM. Our approach has two key novel improvements: first, we compute an effective error quadric for edge

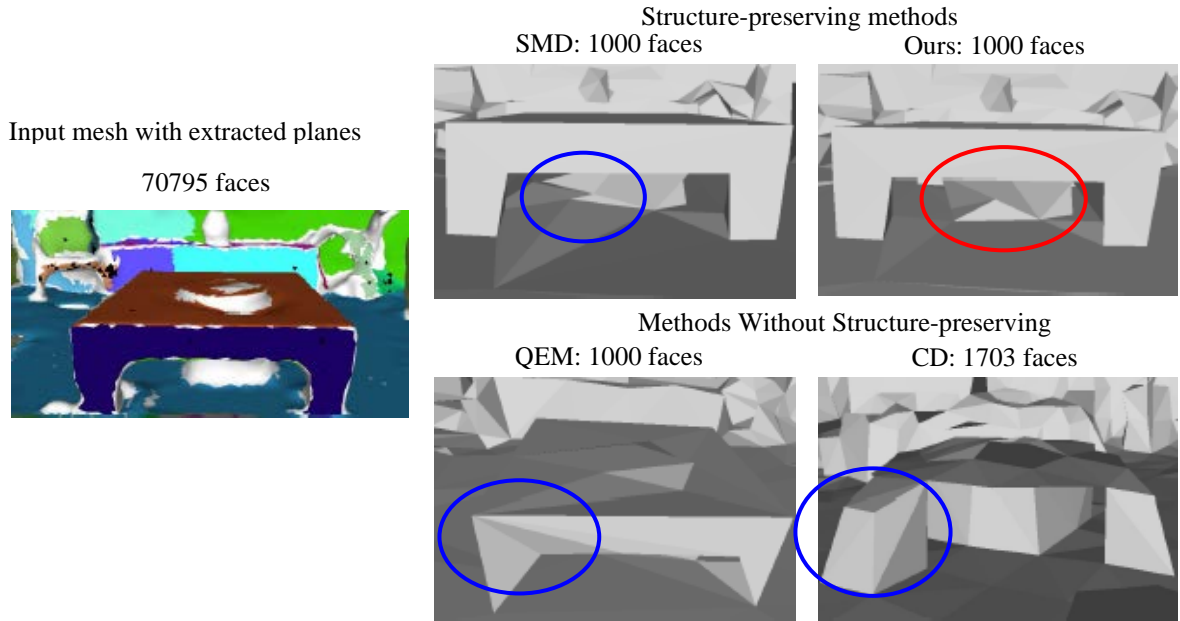


Fig. 6. Illustration of structure-preserving simplification method. **Left:** input mesh. **Top Right:** meshes simplified by structure-preserving methods (SMD[16] and ours). **Bottom Right:** meshes simplified by common methods, such as QEM[4] and CD[18], which are structure-unaware.

collapse, which the planar structures are taken into account as plane constraint(section 3.3.1); second, for vertices on the 3D contour constraint structure, linear feature preserving strategies are applied to avoid mesh degeneration(section 3.3.2).

3.3.1 Plane Constraint QEM

The origin method first proposed by Garland et.al [4] uses iterative edge collapse to simplify models and maintains surface error approximations with QEM. Every vertex is associated with a set of planes which pass through the triangles that meet at this vertex. The error approximation of a vertex v is defined as the sum of squared distances to its associated set of planes P :

$$\Delta v = \sum_{p \in P(v)} (p^T v)^2, \quad (4)$$

where $p = [a \ b \ c \ d]^T$ denotes a plane $ax + by + cz + d = 0$ and note that $a^2 + b^2 + c^2 = 1$.

The error metric above can be rewritten as a quadratic form, which the coefficients are arranged into a 4×4 symmetric matrix. And the sum of squared distances are represented as a sum of quadrics:

$$\Delta v = \sum_{p \in P(v)} v^T (pp^T) v = v^T \left(\sum_{p \in P(v)} Q_p \right) v, \quad (5)$$

where Q_p is the quadric matrix of plane p :

$$Q_p = \begin{pmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{pmatrix}$$

Garland et al. only compute the set of planes to each vertex during the initialization and do not update the plane set for the new vertex of a collapse by simply adding the associated quadric matrices. To be more specific, they defined a symmetric 4×4 matrix $Q_v = \sum_{p \in P(v)} Q_p$ for vertex v . To approximate the error for a given collapse edge $(v_1, v_2) \rightarrow \tilde{v}$, they simply add the two quadric matrices of v_1 and v_2 and assign it to the new vertex \tilde{v} .

However, as experiments in [7] show that memoryless simplification methods usually yield lower mean errors, we also adopt the memoryless method that updates the plane set and recomputes the quadric matrix for vertices when a collapse is over. Fortunately, each edge collapse is a local operation, we thus do not have to recompute all the vertices of the whole mesh but only the vertices related to the edge, which avoid the expensive computations.

In addition to the origin error quadrics, another term for planar structure constraint is considered as following:

$$Q_e = (1 - \alpha)(Q_{v_1} + Q_{v_2}) + \alpha \cdot Q_{PC}, \quad (6)$$

with α being the trade-off parameter and Q_{PC} being the quadrics of planar structure constraint defined as:

$$Q_{PC} = \begin{cases} \sum_{p \in PS(e)} Q_p & \text{if } PS(e) \neq \emptyset \\ 0 & \text{otherwise} \end{cases}, \quad (7)$$

where $PS(e)$ denotes the set of planes corresponding to planar structures that the edge e belongs to.

Similar to [16], during each edge collapse our metric utilizes both the local planes of triangles and the global planes of planar structures associated with the target edge, which increase the robustness of the simplification. $|PS(e)|$, the size of our plane set, is no more than two with our non-overlapping planar structure extraction. We can thus define following 3 types for it: 0 means the edge do not belong to any planar structure; 1 means it is inside a planar structure; 2 means it is on the boundary shared by two planar structures. In this way, we simplify the computation so as to improve the efficiency.

3.3.2 Linear Features Preserving

For edges associated to the structural line segments, simply applying the QEM edge collapse on them will lead to structure degeneration, which is a serious problem especially in the extrem mesh simplification. Vertices on the boundary edges have significant influence to

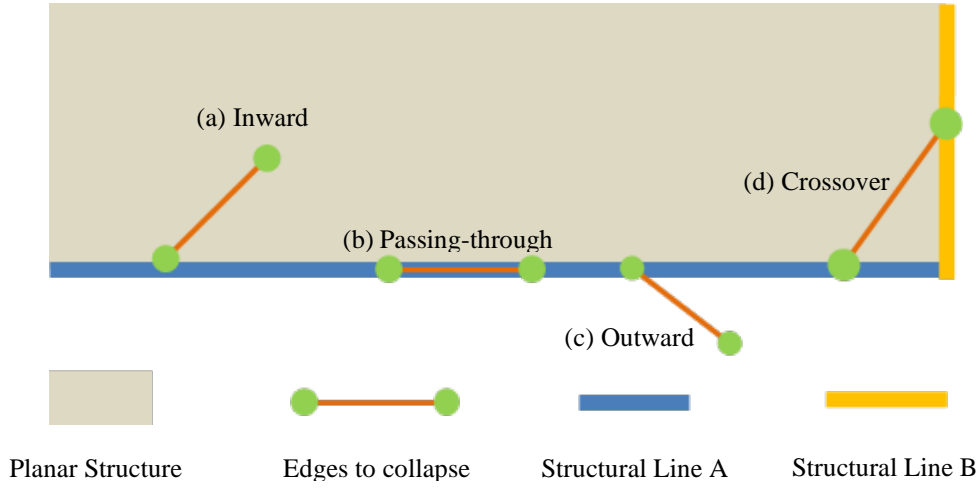


Fig. 7. An illustration of linear features preserving. Different edges related to the boundary of planar structure are defined as being of 4 types: Inward, outward, passing-through, crossover. For these structural edges, we apply specific preserving strategy.

the model structure, moving vertices of boundary edges away from the boundary structure, without considering the structural constraint, will cause a discontinuous and deformed result.

To exploit the 3D line information obtained by the Structural Line Segments Extraction described in section 3.2, we present a linear feature preserving strategy. As mentioned before, the structural line segments are filtered by the boundary of the planar structure and correspond to the 3D contour of the mesh model, therefore it could be utilized to guide the mesh simplification. Different edges related to the boundary of planar structure are defined as being of 4 types (Fig. 7 is an example for the following illustration):

Inward. Edges with a vertex inside the planar structure and another on the contour, like Fig. 7 (a). We do not compute the QEM optimized position and directly move the new vertex to the contour vertex of the target edge.

Outward. Edges with a vertex that belongs to none of the planar structures and another on the contour, like Fig. 7 (c). The same as the type of inward, we directly move the new vertex to the contour vertex of the target edge.

Passing-through. Edges that both vertices of them are on the same line segment of the contour structure, like Fig. 7 (b). If both of the edge vertices are the end points of the same line segment in the contour structure, this collapse operation will be penalized with high penalty weight. If only one of the vertices is the end point, we move the new vertex to the vertex that represents the end point. Otherwise, for edges that neither of the vertices are end points, the midpoint of this edge is set as the new vertex of the collapse.

Crossover. Edges that both vertices of them are on the contour structure but belong to different segments, like Fig. 7 (d). We penalize this collapse operation which will destroy the origin topology of the structure.

Our approach on these structural edges could skip lots of computation, and keep the structure of the mesh from degeneration caused by these collapse operations. Due to these linear feature preserving rules, we can keep the straightness and sharpness of the mesh while applying the edge collapse based on plane constrained QEM, even the mesh is noisy, which will be demonstrated in the experiments section.

4. Experimental Results and Discussion

In this section, we conduct several experiments to valid the effectness of our structure preserving mesh simplification. We compare our method to the most common used simplification method: quadric error metric(QEM) method and its variants implemented in VCG library [4], to the Clustering Decimation (CD) method[18], and to the Structure-aware Mesh Decimation (SMD) approach [16].

4.1 Implemented Details

Our algorithm is implemented with C++ and tested on a PC with i7 3.60GHz processor and 16GB RAM. We choose COLMAP[29] to estimate the camera pose and sparse 3D points which provide the input of 3D line modeling. The Multi-View Stereo reconstruction (MVS) is completed with OpenMVS, which is a open-source 3D reconstruction library based on [30-32]. For structural segments extraction, we set the 2D segments matching threshold to $\tau = 0.3$. In the process of simplification, the trade-off parameter of plane constraint QEM to $\alpha = 0.2$.



Fig. 8. Dataset *Factories*. Input mesh generated by [32] with 280 images, composed of 3M faces. The mesh contains different kinds of buildings with rich planar and linear features. Simplification results are demonstrated in Fig. 9 and compared to results from different simplification methods.

4.2 Experimental results

We first compare our test results to different simplified mesh results generated by other methods, as illustrated in [Fig. 8](#), [Fig. 9](#), [Fig. 10](#) and [Fig. 11](#), to valid the effectiveness of our simplification approach.

In [Fig. 8](#) and [Fig. 9](#), comparisons are made on a large-scale outdoor urban scene, which is reconstructed from 280 aerial images and composed of 3M faces. We pick four typical regions to demonstrate the simplification results and the mesh is decimated with varying target numbers of vertices. Our method has similar results with other methods when the target number is large enough, as shown in Colum 1, 2, 3 which is corresponding to 50%, 20%, 10% vertices. However, in the output with 1% vertices, our method could generate a better mesh with preserving structure (marked in red circles), while results by QEM, QEM with Planar Simplification (PS) and SMD[16] have varying degrees of degradation (marked in blue circles). Although SMD is the best one among them, which uses planar proxies to preserve the structure, in some cases its results cannot preserve the boundary of planar structures without linear feature constraints.

[Fig. 10](#) shows that our method could enhance the structural features of raw mesh and preserve them from degeneration during the simplification, while QEM, QEM with normal preservation (NP) and SMD have varying degrees of degradation (marked in blue circles). Especially, the linear boundary of our result is straight and sharp, and the details associated with the linear features are preserved well, as marked in red circle.

As demonstrated in [Fig.11](#), the Metro geometric comparison tool is used in order to measure surface deviation. As [\[8, 16\]](#), we use a symmetric Hausdorff error to evaluate the quality of the simplified mesh. QEM method could have low mean error at the beginning since its greedy optimization idea, however, without structural constraints, its error metric will increase rapidly at the coarser resolution. Among structure-preserving methods, SMD with constraints of planar proxies is better than other QEM variants, while our method exploiting

Table 1. Timing comparison. All timings are measured on an Intel i7-8700@3.2 GHz 8 cores CPU with 8GB RAM.

Input Mesh	#faces (before)	#faces (after)	Timing(s)			Methods
			Plane detection	Line extraction	simplification	
<i>Factories</i>	3,034,211	30,000	/	/	77	QEM
			/	/	83	QEM with NP
			/	/	86	QEM with PS
			18	/	234	SMD
			22	31	183	Ours
<i>Sofa</i>	30,000	3,000	/	/	0.43	QEM
			/	/	0.46	QEM with NP
			/	/	0.48	QEM with PS
			0.11	/	1.31	SMD
			0.14	0.18	1.03	Ours

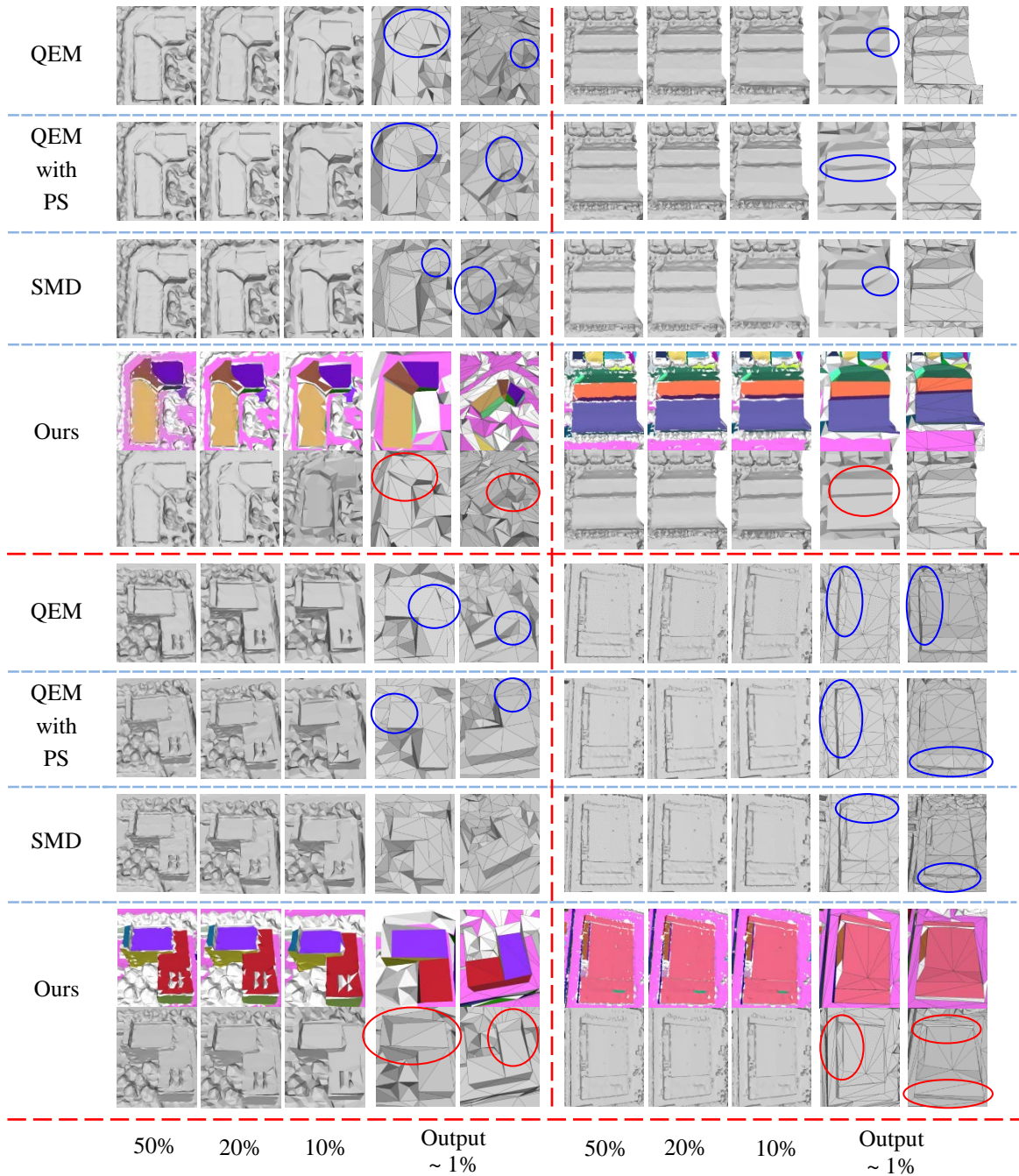


Fig. 9. Comparisons on dataset *Factories*. The top-left, the top-right, the bottom-left and the bottom-right respectively correspond to typical regions (a), (b), (c), (d) in Fig. 8. The input mesh is decimated with 50%, 20%, 10% and final 1% vertices (the last two columns). Experimental results show that our method could generate a better mesh with preserving structure (marked in red circles), while results by QEM, QEM with Planar Simplification (PS) and SMD have varying degrees of degradation (marked in blue circles).

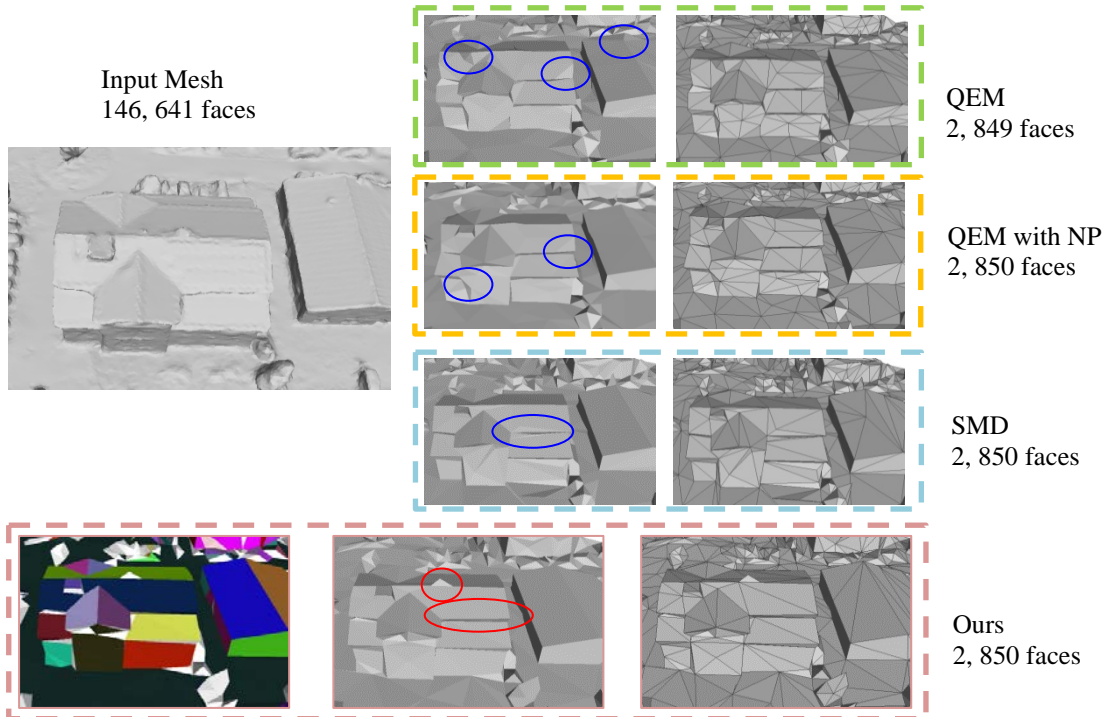


Fig. 10. Comparisons. The top-left is the input mesh with 146, 641 faces. The right and the bottom are outputs at coarse resolution of 2, 850 faces. Our method could preserve main structure and details near the linear features, while others cannot.

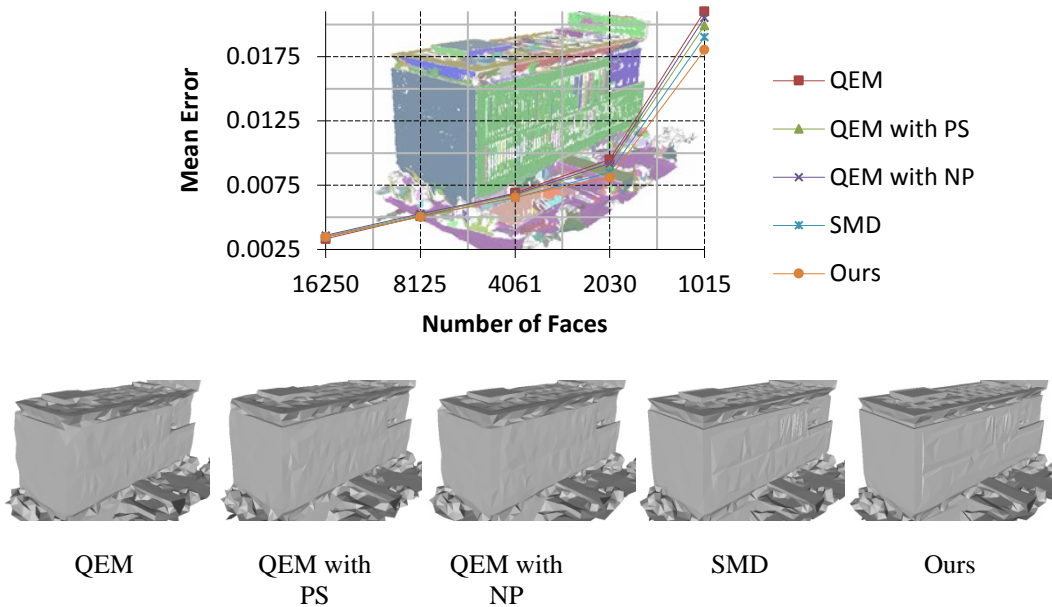


Fig. 11. Comparisons. The input mesh is obtained from OpenMVS and composed of 267,846 faces. We compared the mean error of simplification methods at different levels of mesh complexity. Our method has similar performance with other methods at a fine level of detail, but at a coarser level of resolution, our method will show better performance by enhancing and preserving main structural features.

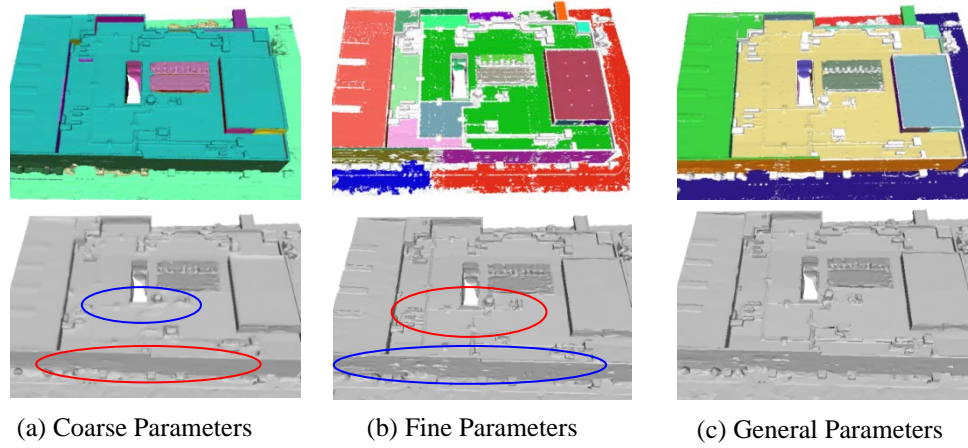


Fig. 12. Results of our simplification method with different levels of error tolerance. The input mesh is composed of 619, 736 faces, and the output has 10, 000 faces.

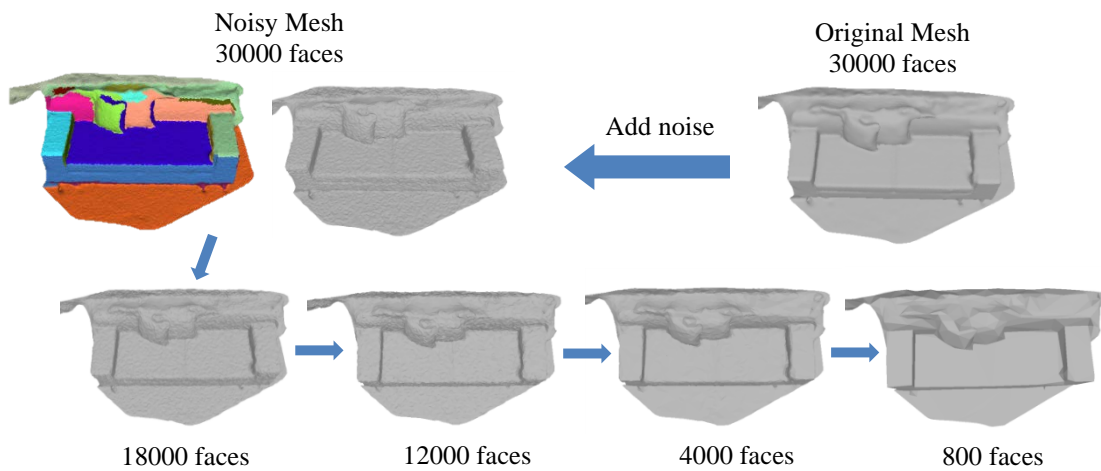


Fig. 13. Structure-preserving mesh simplification on dataset *Sofa* with uniform noise. Top Right: original input mesh. Top Left: noisy mesh with 10% noise. We move the vertices along random directions by 0~10% of the average edge length. Bottom: results of the mesh with different face number during our simplification.

both the planar and linear features could best preserve the structures from degeneration.

We also make experiments to study the influence of our iterative plane structure detection. Results of our simplification method with different level of error tolerance are shown in Fig. 12. Coarse level of error tolerance could better merge different parts of planar structures (marked in red circles) and cover the noisy regions but also might include the wrong planes (marked in blue circles), as (a) shows. Fine level of error tolerance could avoid the incorrect fusion (marked in red circles) but might cause some planar structures are filtered out (marked in blue

circles), as demonstrated in (b). In general, the following normal error tolerances of the planar structure detection perform well in most cases: 15, 30, 45 degrees in the three different iterations, and 15 degrees in the final fusion, as shown in (c).

In [Table 1](#), we demonstrate the timing comparisons on different datasets. Compared to other methods, we have an extra processing time on extracting structural information, however, it may not be a problem since the processing bottle-neck may lie with the throughout of final simplification. Experiments show that our method has the lower running time than SMD while achieving the best performance.

We finally test the robustness of our method to the noise. [Fig. 13](#) shows our experimental result on dataset *Sofa*. In this experiment, we add noise to the original mesh by moving the vertices along random directions by 0~10% of the average edge length and simplify the noisy mesh with our approach. Our planar structure detection could overcome these noise so that our planar and linear structure preservation still works well and is able to keep the mesh model from degeneration. Experimental result shows that our iterative edge collapse could clean the noise and improve the quality.

5. Conclusion

Mesh model generated from 3D reconstruction usually comes with lots of noise, which challenges the performance and robustness of mesh simplification approaches. In this paper, we propose a novel method for mesh simplification to overcome this problem. It consists of three parts: an effective planar structure detecting approach, an automated structural line segments extracting algorithm and the structure-preserving simplification algorithm with planar and linear feature constraints. We have done our experiments on different kinds of datasets to demonstrate that our method could generate a clean, compact and structure-preserving mesh from a noisy raw mesh. In addition the qualitative and quantitative comparisons between our method and other methods show that our method, compared to other simplification algorithms, could efficiently improve the quality of mesh at coarse level of detail and yield an increased robustness on noisy input mesh.

In future work, we plan to additionally consider semantic information or texture information as constraint rules in our regularization and simplification.

References

- [1] Schroeder, William J., Jonathan A. Zarge, and William E. Lorensen, "Decimation of triangle meshes," *ACM siggraph computer graphics*, Vol. 26. No. 2, 1992. [Article \(CrossRef Link\)](#).
- [2] Turk, Greg, "Re-tiling polygonal surfaces," *ACM SIGGRAPH Computer Graphics*, Vol. 26. No. 2, 1992. [Article \(CrossRef Link\)](#).
- [3] Rossignac, Jarek, and Paul Borrel, "Multi-resolution 3D approximations for rendering complex scenes," *Modeling in computer graphics*, Springer, Berlin, Heidelberg, pp. 455-465, 1993. [Article \(CrossRef Link\)](#).
- [4] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proc. of the 24th annual conference on Computer graphics and interactive techniques*, pp. 209-216, 1997. [Article \(CrossRef Link\)](#).
- [5] Low, Kok-Lim, and Tiow-Seng Tan, "Model simplification using vertex-clustering," in *Proc. of the 1997 symposium on Interactive 3D graphics*. ACM, 75-ff, 1997. [Article \(CrossRef Link\)](#).
- [6] Luebke, David, and Carl Erikson, "View-dependent simplification of arbitrary polygonal environments," in *Proc. of COMPUTER SCIENCE*, 2006. [Article \(CrossRef Link\)](#).

- [7] P. Lindstrom and G. Turk, "Evaluation of memoryless simplification," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 2, pp. 98-115, 1999. [Article \(CrossRef Link\)](#).
- [8] D. Cohen-Steiner, P. Alliez, and M. Desbrun, "Variational shape approximation," *ACM Transactions on Graphics (ToG)*, vol. 23, no. 3, 2004. [Article \(CrossRef Link\)](#).
- [9] H. Borouchaki, P. J. C. m. i. a. m. Frey, and engineering, "Simplification of surface mesh using Hausdorff envelope," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 48-49, pp. 4864-4884, 2005. [Article \(CrossRef Link\)](#).
- [10] E. Ovreiu, J. G. Riveros, S. Valette, and R. Prost, "Mesh simplification using a two-sided error minimization," in *Proc. of International Conference on Image, Vision and Computing*, 2012.
- [11] Y. Chen et al., "Research of improving semantic image segmentation based on a feature fusion model," *Journal of Ambient Intelligence and Humanized Computing*, 2020. [Article \(CrossRef Link\)](#).
- [12] Y. Chen et al., "Single-image super-resolution algorithm based on structural self-similarity and deformation block features," *IEEE Access*, vol. 7, pp. 58791-58801, 2019. [Article \(CrossRef Link\)](#).
- [13] Y. Chen, W. Xu, J. Zuo, and K. J. C. C. Yang, "The fire recognition algorithm using dynamic feature fusion and IV-SVM classifier," *Cluster Computing*, vol. 22, pp. 7665-7675, 2019. [Article \(CrossRef Link\)](#).
- [14] Y. Chen et al., "Multiscale fast correlation filtering tracking algorithm based on a feature fusion model," *Concurrency and Computation: Practice and Experience*, 2019. [Article \(CrossRef Link\)](#).
- [15] Y. Chen et al., "The visual object tracking algorithm research based on adaptive combination kernel," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, pp. 4855-4867, 2019. [Article \(CrossRef Link\)](#).
- [16] Salinas, David, Florent Lafarge, and Pierre Alliez, "Structure-Aware Mesh Decimation," in *Proc. of Computer Graphics Forum*, Vol. 34, No. 6, pp. 211-227, 2015. [Article \(CrossRef Link\)](#).
- [17] Liu, Shaohui, et al., "Feature-preserving mesh denoising based on guided normal filtering," *Multimedia Tools and Applications*, vol. 77, pp. 23009-23021, 2018. [Article \(CrossRef Link\)](#).
- [18] Z. Hua, Z. Huang, J. J. I. J. o. M. Li, and U. Engineering, "Mesh simplification using vertex clustering based on principal curvature," *International Journal of Multimedia and Ubiquitous Engineering*, Vol.10, No.9, pp.99-110, 2015. [Article \(CrossRef Link\)](#).
- [19] B. Micusik and H. Wildenauer, "Structure from motion with line segments under relaxed endpoint constraints," *International Journal of Computer Vision*, vol. 124, pp. 65-79, 2017. [Article \(CrossRef Link\)](#).
- [20] Sugiura, Takayuki, Akihiko Torii, and Masatoshi Okutomi, "3D surface reconstruction from point-and-line cloud," in *Proc. of 3D Vision (3DV), 2015 International Conference on. IEEE*, 2015. [Article \(CrossRef Link\)](#).
- [21] Jain, Arjun, et al., "Exploiting global connectivity constraints for reconstruction of 3D line segments from images," in *Proc. of CVPR*, 2010. [Article \(CrossRef Link\)](#).
- [22] M. Hofer, M. Maurer, and H. Bischof, "Efficient 3D scene abstraction using line segments," *Computer Vision and Image Understanding*, vol. 157, pp. 167-178, 2017. [Article \(CrossRef Link\)](#).
- [23] Li, Yangyan, et al., "Globfit: Consistently fitting primitives by discovering global relations," *ACM Transactions on Graphics (TOG)*, Vol. 30, No. 4, 2011. [Article \(CrossRef Link\)](#).
- [24] Lafarge, Florent, et al., "A hybrid multiview stereo algorithm for modeling urban scenes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 5-17, 2013. [Article \(CrossRef Link\)](#).
- [25] J. Wang et al., "Image-based building regularization using structural linear features," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 6, pp. 1760-1772, 2016. [Article \(CrossRef Link\)](#).
- [26] R. G. Von Gioi, J. e. r. e. m. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: a line segment detector," *Image Processing On Line*, vol. 2, pp. 35-55, 2012. [Article \(CrossRef Link\)](#).
- [27] Hofer, Manuel, Michael Maurer, and Horst Bischof, "Line3d: Efficient 3d scene abstraction for the built environment," in *Proc. of German Conference on Pattern Recognition. Springer, Cham*, pp. 237-248, 2015. [Article \(CrossRef Link\)](#).

- [28] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International journal of computer vision*, vol. 59, pp. 167-181, 2004. [Article \(CrossRef Link\)](#).
- [29] Schönberger, Johannes L., et al., "Pixelwise view selection for unstructured multi-view stereo," in *Proc. of European Conference on Computer Vision*. Springer, Cham, pp. 501-518, 2016. [Article \(CrossRef Link\)](#).
- [30] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM ToG*, pp. 1-11, 2009. [Article \(CrossRef Link\)](#).
- [31] M. Jancosek and T. J. I. s. r. n. Pajdla, "Exploiting visibility information in surface reconstruction to preserve weakly supported surfaces," vol. 2014, 2014.
- [32] H.-H. Vu, P. Labatut, J.-P. Pons, and R. Keriven, "High accuracy and visibility-consistent dense multiview stereo," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 5, pp. 889-901, 2012. [Article \(CrossRef Link\)](#).



Zhuo Chen received the B.S. degree and is currently pursuing a Phd degree at Huazhong University of Science and Technology, Hubei, China. His major research interests include computer graphics and computer vision.



Xiaobin Zheng received the M.S. degree at Huazhong University of Science and Technology, Hubei, China. His major research interests include signal processing, video coding and compressive sensing.



Tao Guan received the Ph.D. degree from the Huazhong University of Science and Technology, Wuhan, China in 2008. He is currently a professor in the School of Computer Science and Technology, Huazhong University of Science and Technology. His research interests include mobile visual search and mobile augmented reality.