

**Title:****Low-poly Mesh Generation from Low-quality Point Clouds Based on Projections****Authors:**

Shinichi Sano, sano.shinichi.b0@elms.hokudai.ac.jp, Hokkaido University

Hiroaki Date, hdate@ssi.ist.hokudai.ac.jp, Hokkaido University

Satoshi Kanai, kanai@ssi.ist.hokudai.ac.jp, Hokkaido University

Keywords:

Point cloud, 3D scanning, Low-poly mesh, projection

DOI: 10.14733/cadconfP.2025.223-228**Introduction:**

Recently, the acquisition of point clouds using small mobile devices such as iPads and iPhones has become common, and it is possible for non-experts in 3D scanning to acquire point clouds of objects and environments around them and to create virtual spaces easily. The point clouds are often generated by a mounted small LiDAR device or photogrammetry. However, the quality of point clouds varies depending on the method of point cloud acquisition. Low-quality point clouds with a large lack of points, variation of point densities, and high-level noise could often be acquired due to the surface properties of the target objects (Fig.1(a)). As a result, low-quality meshes with undesired bumps or large holes are obtained (Fig.1(b)). There is room for improvement in the rapid generation of useful meshes using existing methods [1].

In this paper, a simple and fast algorithm for low-poly mesh generation from low-quality point clouds is presented. Our method is based on Gao's method [2] for low-poly mesh generation from given high-density meshes of buildings. In our method, the low-quality point clouds are projected onto the 2D planes, and contours are created by image processing techniques, and the final mesh is obtained as an intersection of swept volumes of contours.



Fig. 1: Point clouds acquired by a mobile device and generated 3D meshes [3]: (a) Low-quality point clouds and (b) Meshes generated by existing methods.

Proposed method for generating low-poly meshes from low-quality point clouds:**Method Overview**

The input of the method is a low-quality point cloud of the target object acquired by small mobile devices, as shown in Fig. 1(a). The output of the method is a low-poly 3D mesh with regularities such as symmetry, parallelism, and orthogonality. In this study, a projection-based simplified mesh generation method for buildings proposed by Gao et al. [2] is extended to generate a low-poly mesh

from a low-quality point cloud. In Gao's method, the low-poly mesh is created by simplifying the mesh of the intersection of swept volumes of silhouettes obtained by projecting a given high-density mesh in multiple directions. This method has the advantage of allowing the application of 2D image and geometry processing. In our method, noise removal, hole filling, and regularization are done by 2D processing. Because the final 3D mesh is represented by the intersection of a set of swept volumes of 2D projections (silhouettes) of a point cloud, the target objects of our study are limited to the objects whose major surfaces appear in the silhouettes, such as office supplies, furniture, and home appliances.

Fig. 2 shows an overview of the method proposed in this study. First, the projection direction that can adequately capture the geometry of the major surfaces of the object is determined by evaluating the normals of the point cloud (A1). Then, the input point cloud is projected onto planes perpendicular to the projection directions to create a set of silhouette images. Noise removal and hole completion are performed by image processing techniques considering symmetries, and a set of silhouette contour polylines is extracted (A2). Next, the polylines are simplified and regularized (A3). Finally, a set of primitives is generated by sweeping the contour polylines along the projection directions, and the simplified intersection of primitives is extracted as a low-poly mesh (A4). Features of the proposed method include fast and robust generation of low-poly mesh with regularities by simple 2D image and contour processing from low-quality point clouds.

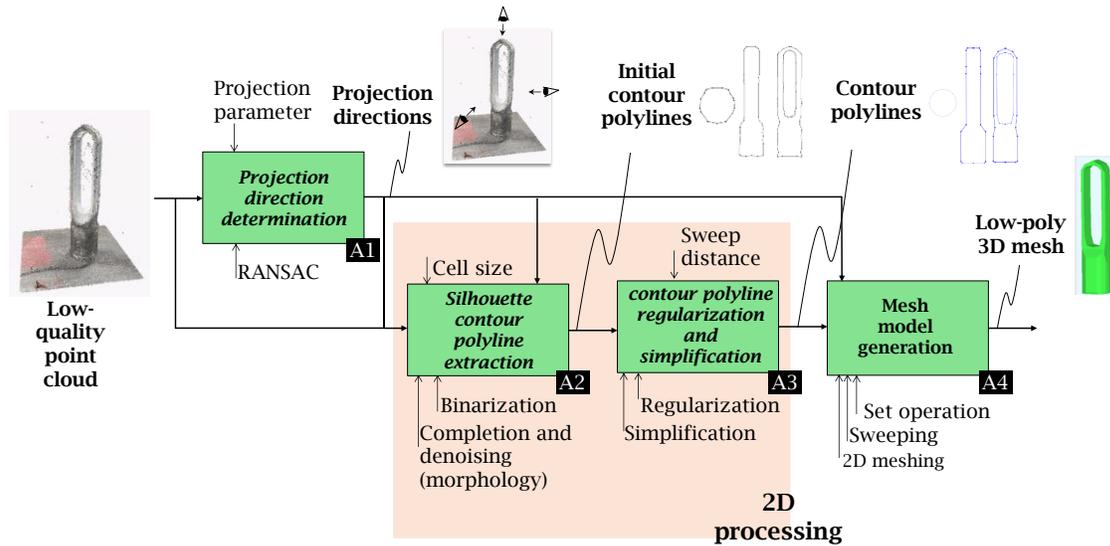


Fig. 2: Overview of the proposed method.

Projection direction determination

In this study, it is assumed that the silhouettes can capture the geometry of the major surfaces of the object from three directions of the top, front, and side views of the input object (Fig.3). The projection direction to obtain the top view is denoted by \mathbf{v}_1 , and it is extracted as the normal of the floor detected by plane fitting using RANSAC. Next, the second direction \mathbf{v}_2 is determined by using projected normals of the object on a plane p_1 perpendicular to \mathbf{v}_1 . First, normals $\{\mathbf{n}_i\}$ of the points that satisfy $|\mathbf{n}_i \cdot \mathbf{v}_1| < \tau_n$ are projected onto the plane p_1 . Second direction \mathbf{v}_2 is determined as the direction in which the projected normals $\{\hat{\mathbf{n}}_i\}$ are concentrated. This direction can be obtained by Equation (1) (Fig.3).

$$\mathbf{v}_2 = \arg \max_{\hat{\mathbf{n}}_i \in N} |S_i|, \quad S_i = \{\hat{\mathbf{n}}_j | \text{angle}(\hat{\mathbf{n}}_i, \hat{\mathbf{n}}_j) < \tau_\theta\} \quad (1)$$

Where, N is a set of projected normals, $\text{angle}(\mathbf{a}, \mathbf{b})$ is the angle between \mathbf{a} and \mathbf{b} , τ_θ is a threshold. The third direction \mathbf{v}_3 is calculated by the outer product of \mathbf{v}_1 and \mathbf{v}_2 .

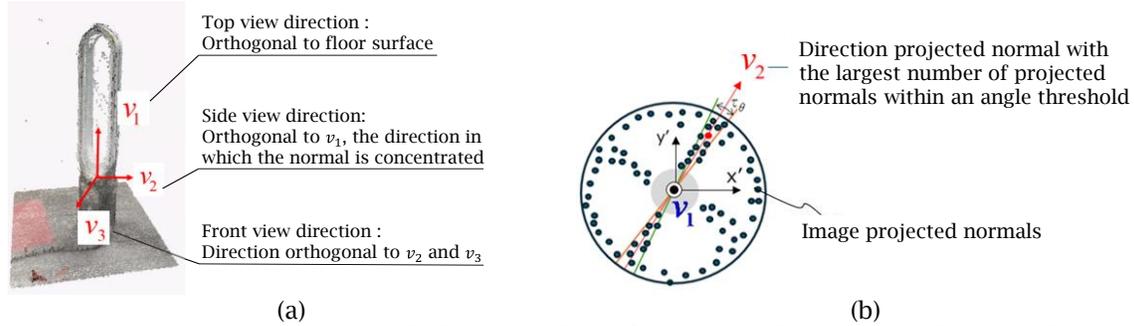


Fig. 3: Projection directions: (a) Calculation method of projection direction and (b) Normal voting.

Silhouette contour polyline extraction

First, the SOR filter for noise removal is applied to the input point cloud and the resulting point cloud is projected onto a plane perpendicular to the projection directions v_d (where $d \in \{1,2,3\}$) and silhouette images I_s^d are created. Here, each I_s^d is a binary image where white cells represent the foreground (object) including projected points and black cells are the background as shown in Fig. 4(a). Next, a closing operation based on morphology is applied to I_s^d to remove small holes and interpolate the lack of points. Then, holes are detected as the connected background pixels, and if the size of the holes is smaller than the given threshold, the holes are filled. Finally, the opening operation based on morphology is applied in order to remove noises, and the modified silhouette images I_m^d are obtained as shown in Fig. 4(b).

Symmetries of each I_m^d are detected. First, each image size is adjusted to the AABB of the foreground of modified images, and the degree of the symmetry r_{sym} is evaluated using Equation (2).

$$r_{sym} = \frac{|R_{org} \cap R_{mir}|}{|R_{org}|} \quad (2)$$

Here, R_{org} is a pixel set of foregrounds of the image and R_{mir} is one for the mirrored image. r_{sym} is the rate of common foreground pixels in the original and mirrored images. If $r_{sym} > \tau_s$ (τ_s : threshold), the silhouette is considered symmetrical and the half image of the intersection between the original and mirrored images is extracted as I_h^d as shown in Fig. 4(c). Finally, the contour polylines (point sequences) of I_h^d for symmetric silhouette or I_m^d for the others are extracted as shown in Fig. 4(d).

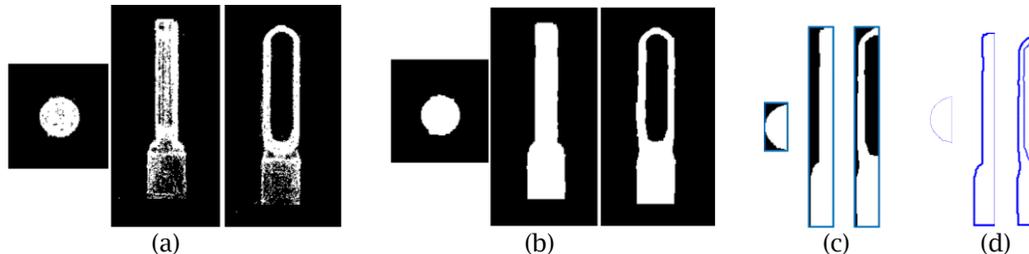


Fig. 4: Projected image and contour polyline: (a) Projected image, (b) Silhouette image, (c) Half image, and (d) Contour point sequences.

Shaped contour generation

To remove small undesired bumps on the contours, polyline smoothing is applied as shown in Fig. 5(a). In our method, an extension of the λ - μ algorithm [4] is used. In this method, the smoothing strength is adaptively controlled to keep the small shapes of the input object. We assume that the small shapes to be preserved are represented by multiple points, and unfewer points in the point clouds represent undesired bumps caused by the noise. Under this assumption, the number of projected points s_i in the image cell corresponding to the vertex i of the contour polyline is used to control the strength of the smoothing. In our method, a weak smoothing operation is applied to vertices that have large s_i by adjusting the coefficients in the iteration of the smoothing operation.

After polyline smoothing, the simplified contour polylines are extracted using the Douglas-Peucker algorithm as shown in Fig. 5(b).

Then, the regularization is done for simplified contour polylines. First, circular arcs of the polylines are detected by RANSAC, and vertex positions on the detected circular arcs are moved to the detected circles. For the others, an algorithm for regularizing 2D polygon proposed by Takahashi et al. [5] is used. In the method, regularities of parallel, orthogonal, co-planar, and regular intervals are imposed by using a graph-based representation of the regularities and the constrained least square fitting (Fig. 5(c)). If symmetrization has been performed, the contour polylines are duplicated and mirrored to generate the entire contour polylines as shown in Fig. 5(d).

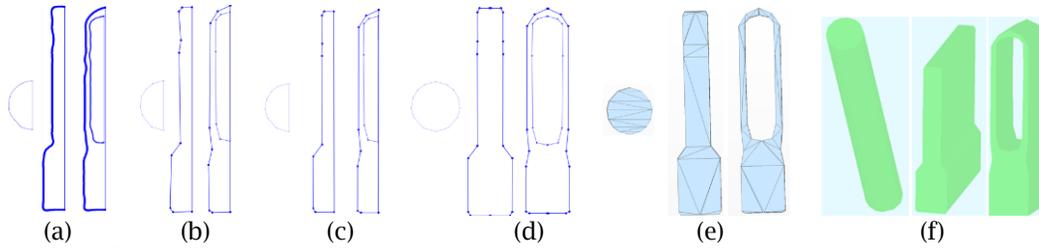


Fig. 5: Flow of primitive generation: (a) Smoothed polylines, (b) Simplified contour polylines, (c) Regularized polylines, (d) Entire contour polylines, (e) 2D triangular meshes, and (f) primitives.

Low-poly mesh generation

2D triangular meshes of the silhouettes are created by applying two-dimensional constrained Delaunay triangulation (Fig. 5(e)) to the resulting contours, and then primitives are generated by swept operation of each 2D mesh along the corresponding projection direction (Fig. 5(f)). Finally, a 3D low-poly mesh is obtained by calculating the mesh of the intersection of generated primitives and simplifying the intersection mesh.

Results and Evaluations:

The point clouds of objects for experiments were acquired using iPhone 15 Pro and 3D scanning software [3]. Photos, point clouds, and generated 3D meshes are shown in Fig. 6, including a number of points, and a number of vertices and triangles of the generated meshes. The large-scale noises and large lack of points can be seen in the point clouds. The results showed that large lacks of points are appropriately filled and low-poly meshes with regularities are generated by using our method. Also, it was confirmed that the projections from three directions derived by the point normals could capture the geometries of major surfaces of the objects. In Fig. 6, the meshes obtained by the popular implicit surface reconstruction method (PSR) [6] and a 3D reconstruction software [3] are also shown. It was observed that the resulting meshes have incorrect surfaces, inappropriate holes, and small bumps. On the other hand, the low-poly meshes without such problems can be obtained using our method. The computation times were less than 3 seconds on a desktop PC (CPU: Intel(R) Core(TM) i7-14700F). Experimental results for other objects are shown in Fig. 7. Although due to the nature of the method, some concave portions of the object cannot be recovered appropriately, low-poly meshes that capture the major surfaces of the objects could be obtained. To improve the detail features and concavities of the resulting meshes by increasing the projection directions and using depth information at each view are included in future works.

Conclusions:

In this paper, we proposed a method for generating low-poly meshes from low-quality point clouds. The method is based on 2D projection of points, image and contour processing, and intersection extraction of primitives generated by contours. In image processing, symmetrization, noise removal, and hole filling are achieved, and in contour processing, regularization and simplification are realized. Through experiments, it was confirmed that the proposed method enables the generation of low-poly meshes from low-quality point clouds of small, simple objects. Future work includes the improvement of detail reproducibility by increasing the projection directions and using depth information at each view.

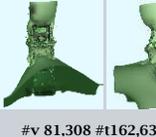
	Proposed method	Existing method	
		PSR[5]	Scaniverse[3]
 points 102,782	 #v 519 #t 1,038	 #v 46,095 #t 92,384	 #v 15,397 #t 30,540
 points 126,814	 #v 406 #t 808	 #v 81,308 #t 162,635	 #v 15,715 #t 31,058
 points 112,904	 #v 544 #t 1084	 #v 77,936 #t 155,906	 #v 29,998 #t 59,414

Fig. 6: The resulting low-poly meshes and a comparison with existing methods.

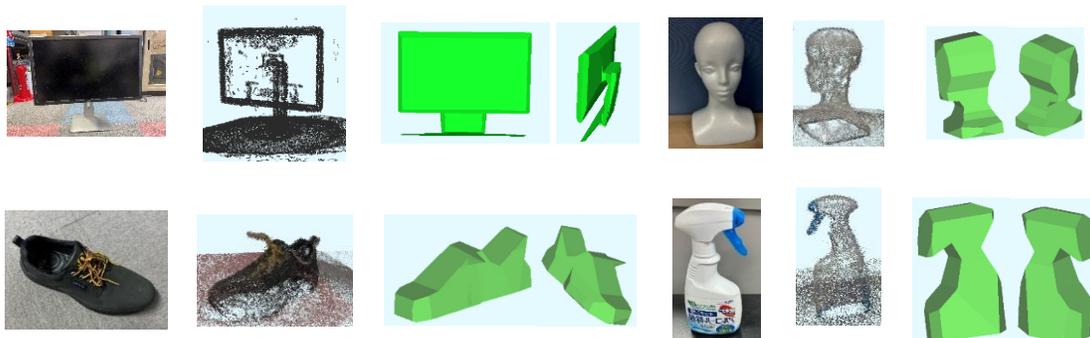


Fig. 7: Resulting low-poly meshes of other objects.

Shinichi Sano, <https://orcid.org/0009-0003-1614-8536>

Hiroaki Date, <https://orcid.org/0000-0002-6189-2044>

Satoshi Kanai, <https://orcid.org/0000-0003-3570-1782>

References:

- [1] Sulzer, R.; Marlet, R.; Vallet, B.; Landrieu, L.: A Survey and Benchmark of Automatic Surface Reconstruction From Point Clouds, IEEE Transactions on Pattern Analysis and Machine Intelligence, 47, 2025, 2000-2019. <https://doi.org/10.1109/TPAMI.2024.3510932>
- [2] Gao, X.; Wu, K.; Pan, Z.: Low-poly Mesh Generation for Building Models, Proc. ACM SIGGRAPH 2022, 2022, 3:1-9.
- [3] Scaniverse, <https://scaniverse.com/>
Taubin, G.: A signal processing approach to fair surface design, Proc. SIGGRAPH 1995, 1995, 351-358. <https://doi.org/10.1145/218380.218473>

- [4] Takahashi, H.; Date, H.; Kanai, S.: Automatic Indoor Environment Modeling from Laser-scanned Point Clouds using Graph-based Regular Arrangement Recognition Proc. ICCBEI 2019, 2019, 368-375.
- [5] Kazhdan, M.; Hoppe, H.: Screened poisson surface reconstruction, ACM Transactions on Graphics, 32(3), 2013, 29:1-13. <https://doi.org/10.1145/2487228.2487237>