

Article

An Efficient and High-Quality Mesh Reconstruction Method with Adaptive Visibility and Dynamic Refinement

Qingsong Yan ¹, Teng Xiao ^{2,3} , Yingjie Qu ¹ , Junxing Yang ⁴  and Fei Deng ^{1,3,*}

¹ School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China; yanqs_wuhu@whu.edu.cn (Q.Y.); quyj_wuhu@whu.edu.cn (Y.Q.)

² School of Computer Science, Hubei University of Technology, Wuhan 430068, China; xiao@hbut.edu.cn

³ Wuhan Tianjihang Information Technology Co., Ltd., Wuhan 430010, China

⁴ School of Geomatics and Urban Spatial Information, Beijing University of Civil Engineering and Architecture, Beijing 102616, China; yangjunxing@bucea.edu.cn

* Correspondence: fdeng@sgg.whu.edu.cn

Abstract: Image-based 3D reconstruction generates 3D mesh models from images and plays an important role in all walks of life. However, existing methods suffer from poor reconstruction quality and low reconstruction efficiency. To address this issue, we propose an improved optimization-based mesh reconstruction method with adaptive visibility reconstruction and dynamic photo-metric refinement. The adaptive visibility reconstruction adjusts soft visibility based on the observation and geometry structure of points to reconstruct details while suppressing noise in the rough mesh. The dynamic photo-metric refinement tunes the learning rate using historical gradients and stops to optimize converged triangles to speed up the mesh refinement. Experiments on BlendedMVS and real datasets showed that our method found a good balance between reconstruction quality and reconstruction efficiency. Compared with the state-of-the-art methods, OpenMVS and TDR, our method achieved higher reconstruction quality than OpenMVS and obtained competitive reconstruction quality with TDR, but required only one-third of the reconstruction time of OpenMVS and one-tenth of the reconstruction time of TDR. Our method balances reconstruction efficiency and reconstruction quality and can meet real-world application requirements.

Keywords: 3D reconstruction; mesh reconstruction; mesh refinement; graph-cut; photo-metric consistency; reconstruction efficiency



Citation: Yan, Q.; Xiao, T.; Qu, Y.; Yang, J.; Deng, F. An Efficient and High-Quality Mesh Reconstruction Method with Adaptive Visibility and Dynamic Refinement. *Electronics* **2023**, *12*, 4716. <https://doi.org/10.3390/electronics12224716>

Academic Editor: Pawel Strumillo

Received: 16 October 2023

Revised: 13 November 2023

Accepted: 19 November 2023

Published: 20 November 2023



Copyright: © 2023 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/>).

1. Introduction

Image-based 3D reconstruction is a core topic in computer vision, which can construct a 3D mesh model of the real world and plays a key role in many fields, such as urban planning and disaster relief. Typically, 3D reconstruction consists of four main components. First, structure from motion (SfM) [1] estimates intrinsic and extrinsic camera parameters from images. Then, multi-view stereo (MVS) [2,3] generates dense point clouds of the scene. Mesh reconstruction [4,5] transforms these point clouds into 3D mesh models. Finally, texture mapping [6] adds textures to meshes to enhance visual quality. However, of these four components, the mesh reconstruction determines the final shape of the mesh. Without a high-quality mesh reconstruction, the utility and fidelity of the generated 3D model will be significantly reduced, no matter how successful the other modules have been.

In general, mesh reconstruction can be categorized into two types of methods, the spline method [7] and the optimization method [4,5,8]. However, the spline method cannot deal with the noise in the point cloud and usually generates a poor-quality mesh that loses the details of the scene. In contrast, the optimization method, which can obtain higher-quality reconstruction results and is currently the dominant method, consists of two steps: visibility reconstruction and photo-metric refinement, where the visibility reconstruction

generates a rough mesh and the photo-metric refinement improves the quality of the rough mesh. Here, we briefly review the development of the optimization method.

The visibility reconstruction first uses the dense point cloud to construct several tetrahedrons, then employs the s-t graph-cut to classify these tetrahedrons based on the visibility information of the point cloud and, finally, obtains the 3D rough mesh [9–11]. However, how to use the visibility information is a problem that has long plagued researchers. Labatut et al. [12] proposed a robust algorithm to smooth the weighting process by introducing soft visibility, which can effectively deal with a large amount of noise in the dense point cloud. Jancosek and Pajdla [13,14] found that regions with few sampling points are hard to reconstruct and used the free-space-support to improve the reconstruction completeness by optimizing the weighting process. Zhou et al. [15] tried to increase the details in the mesh by removing the soft visibility and increasing the number of points. Zhou et al. [4] considered that the point cloud can only affect the tetrahedron of adjacent regions and optimized the weighting process to avoid the reduction of the surface integrity due to occlusion. Apart from that, introducing external constraints is a good idea. Labatut et al. [16] proposed a hierarchical algorithm to extract basic shape structures from the dense point cloud, such as planes, spheres, cones, and cylinders, and used them as constraints to guide the mesh reconstruction. Li et al. [17], on the other hand, constrained the mesh reconstruction based on the 3D line extracted by [18] to improve the reconstruction quality of the thin structure. However, these methods ignore the different quality and importance of each point in the dense point cloud and treat them equally, failing to balance between removing noise and preserving details on the rough mesh.

The photo-metric refinement uses the 3D mesh to render a virtual image in a new perspective and tries to improve the mesh quality by maximizing the photo-metric consistency between the virtual image and the real image [8]. However, while this method can obtain a high-quality mesh, it has high computational complexity and low reconstruction efficiency. Li et al. [19] and Zhang et al. [20] used an adaptive refinement algorithm, aiming to improve the reconstruction efficiency with as little loss of reconstruction quality as possible. On the other hand, Morreale et al. [21], Yan et al. [22], and Romanoni and Matteucci [23] believed that there are a large number of redundant images during the refinement and only selected the optimal image for each triangle in the mesh to avoid using images with poor observational conditions and improve the reconstruction efficiency. Meanwhile, other scholars have attempted to improve the performance of the photo-metric refinement. Blaha et al. [24] and Romanoni et al. [25] introduced semantic information into the photo-metric refinement. Fei et al. [26] used Line3D [18] to obtain 3D lines and used them as constraints for photo-metric refinement to prevent them from being too smooth. Romanoni and Matteucci [27] considered the self-occlusion problem caused by the depth consistency to select image pairs. Qu et al. [5] improved the photo-metric consistency of zero-normalized cross-correlation (ZNCC) through total differentiation and enhanced the mesh quality through adaptive mesh filtering. However, it is still challenging to balance reconstruction quality and reconstruction efficiency in photo-metric refinement. Increasing reconstruction quality means decreasing reconstruction efficiency, while increasing reconstruction efficiency means losing reconstruction quality.

In summary, existing 3D reconstruction methods struggle to find a balance between reconstruction quality and reconstruction efficiency. To address this issue, we propose an improved optimization-based method that can improve the reconstruction efficiency and reconstruction quality through adaptive visibility reconstruction and dynamic photo-metric refinement in this paper. The adaptive visibility reconstruction adjusts the soft visibility of each point in the point cloud according to the observation and geometry structure, thus retaining more-detailed information and effectively suppressing noise. The dynamic photo-metric refinement speeds up the convergence speed by changing the learning rate through the history gradient of the triangle and stopping to optimize the converged triangle to reduce the computational complexity. To validate the effectiveness of our method, we conducted quantitative and qualitative experiments on BlendedMVS [28]

and two real-world datasets. We compared our method with two state-of-the-art methods, OpenMVS [8,29] and TDR [5]. The results demonstrated that our method had a higher reconstruction quality than OpenMVS and achieved comparable reconstruction quality with TDR. Meanwhile, the reconstruction efficiency of our method was greatly improved. Our method only required around 1/3 of the time OpenMVS needed and 1/10 of the time TDR needed to complete the reconstruction, which proved that our method is more valuable for practical applications.

Our contributions are as follows:

1. We propose an improved optimization mesh-reconstruction method, and extensive experiments on BlendedMVS proved that our method can reconstruct a high-quality mesh with higher efficiency, taking only 1/3 of the reconstruction time of OpenMVS [8,29] and 1/10 of the reconstruction time of TDR [5] to complete the reconstruction.
2. We propose an adaptive visibility reconstruction, which analyzes the quality and importance of different points in the dense point cloud to maintain enough details and remove noise to obtain a better rough mesh.
3. We propose dynamic photo-metric refinement to improve the reconstruction quality and efficiency of the photo-metric refinement by utilizing the triangle gradient to adjust the learning rate and stop optimizing converged triangles dynamically.

2. Method

After obtaining the intrinsic and extrinsic camera parameters of a set of images \mathcal{I} by SfM and the dense point clouds \mathcal{P} from MVS, we reconstructed the mesh M through adaptive visibility reconstruction and dynamic photo-metric refinement, as shown in Figure 1. Each point p in \mathcal{P} has three types of attribution: position, color, and visibility v_p , where v_p represents the set of images that can see the p .

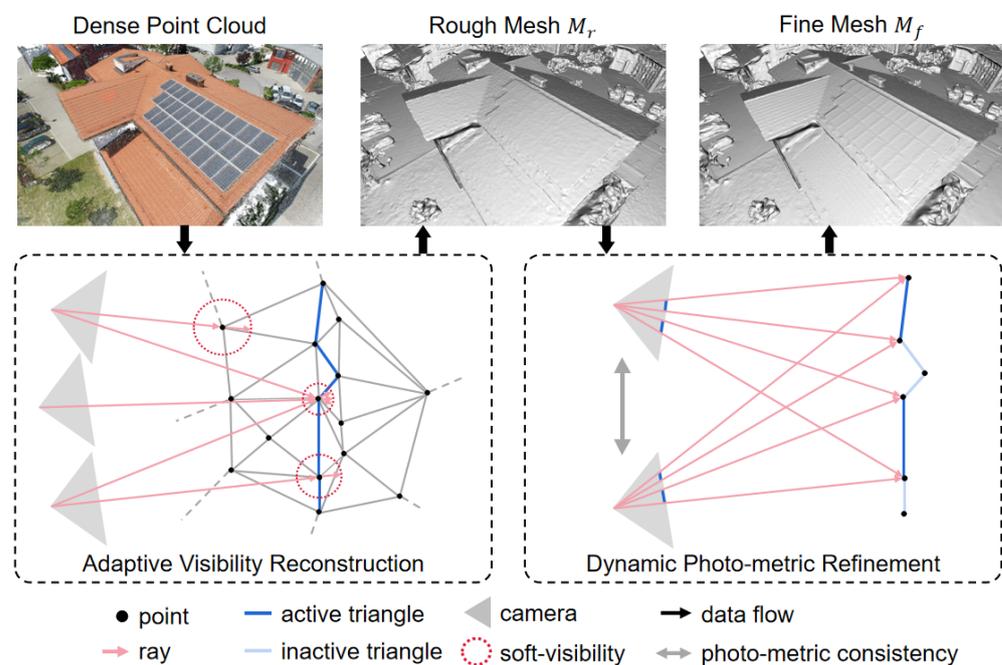


Figure 1. The pipeline of the proposed method. Given a dense point cloud \mathcal{P} , our method first reconstructs the rough mesh M_r by adaptive visibility reconstruction, where the weight of each edge in the s-t graph \mathcal{G} is determined by adaptive soft visibility. Then, our method refines M_r to obtain the fine mesh M_f through dynamic photo-metric refinement, which analyzes the convergence state of each triangle to speed up refinement.

In the adaptive visibility reconstruction, we reconstructed the rough mesh M_r by subdividing the space into several tetrahedrons \mathcal{T} and utilized the visibility of $p \in \mathcal{P}$ to

classify these tetrahedrons into two categories, including “inside the surface” and “outside the surface”. To deal with noise while maintaining details, we propose adaptive soft visibility to set the weight of the edges in the s-t graph \mathcal{G} based on the geometry structure and visibility of each point. In the photo-metric refinement, we used the photo-metric consistency to refine the rough mesh M_r and reconstruct the fine mesh M_f with less noise and more details. To speed up the refinement procedure, we checked the converging state of each triangle in the mesh and applied the dynamic learning rate strategy to automatically adjust the learning rate and the dynamic triangle selection to split triangles into the “active triangle” and the “inactive triangle”.

We will describe the details of the proposed method in the following section. Section 2.1 introduces how to reconstruct the rough mesh M_r using the adaptive visibility reconstruction, which can adjust the weight of edges in the s-t graph \mathcal{G} . Section 2.2 describes the details of the dynamic photo-metric refinement, which can accelerate the refinement procedure to obtain the fine mesh M_f .

2.1. Adaptive Visibility Reconstruction

Due to mismatches in MVS [2–4], \mathcal{C} contains much noise. To deal with this noise, Vu et al. [8] utilized the s-t graph-cut to obtain the rough mesh M_r . The core idea of this method is to divide the whole space into a series of tetrahedrons \mathcal{T} and classify them into “inside the surface” and “outside the surface” via the visibility information of each point in the point clouds. Furthermore, Labatut et al. [12] proposed soft visibility to model the uncertainty of each point and generate a smoother mesh. However, they ignored that the quality and importance of each point are different and treated them equally, resulting in an over-smoothed mesh. Therefore, we propose an adaptive visibility reconstruction, which can find a balance between reducing noise and maintaining details.

Following Vu [8] and Labutu [12], we constructed tetrahedrons \mathcal{T} by \mathcal{P} and built an s-t graph \mathcal{G} to classify these tetrahedrons. Based on the space relationship between tetrahedrons, we can define the node \mathcal{N} and edge \mathcal{E} in \mathcal{G} . There are three types of nodes in \mathcal{G} . The $n_{in} \in \mathcal{N}$ node and the $n_{out} \in \mathcal{N}$ node represent “inside the surface” and “outside the surface”, respectively. The $n_t \in \mathcal{N}$ node indicates each tetrahedron t in \mathcal{T} . There are three types of edges in \mathcal{G} . The $e_{in \rightarrow t}$ edge denotes the possibility that a tetrahedron belongs to n_{in} , and the $e_{out \rightarrow t}$ edge means the potential that a tetrahedron belongs to n_{out} , while the e_f edge denotes the potential that two adjacent tetrahedrons belong to the same category, where f is the adjacent face between two tetrahedrons. We show a demo of building the s-t graph \mathcal{G} from five points in Figure 2.

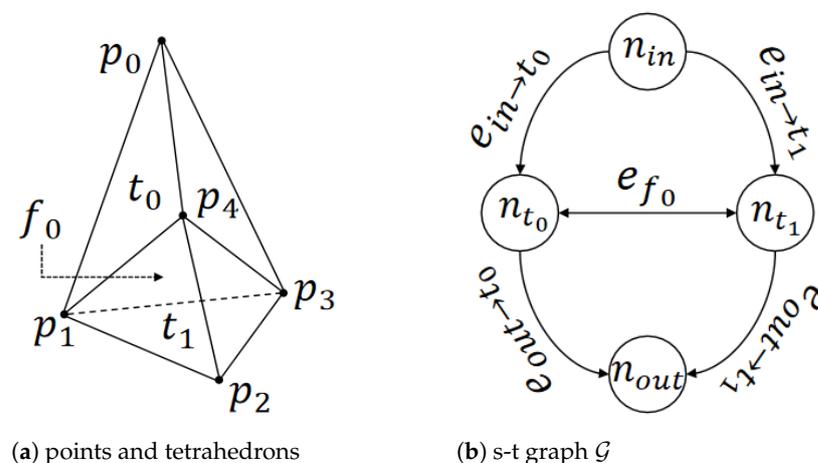


Figure 2. The tetrahedrons \mathcal{T} and the s-t graph \mathcal{G} . Given five points p_0, p_1, p_2, p_3, p_4 , we can build two adjacent tetrahedrons t_0 and t_1 , where f_0 is the adjacent face between t_0 and t_1 . Based on the tetrahedrons, the corresponding s-t graph \mathcal{G} has two n_t nodes n_{t_0}, n_{t_1} , one e_f edge e_{f_0} , two $e_{in \rightarrow t}$ edges, and two $e_{out \rightarrow t}$ edges.

To classify the node n_t in the graph \mathcal{G} , we used the visibility of \mathcal{P} to determine the weight of each edge \mathcal{E} in \mathcal{G} . The visibility comes from the depth map fusion in MVS [2–4,30], where each point $p \in \mathcal{P}$ has a set of visible images v_p . For each image $I_i \in v_p$, we can construct a ray $\vec{r}_{I_i \rightarrow p}$ that traverses several tetrahedrons. Assuming p is located on the surface, the tetrahedrons between I_i and p are “outside the surface” and the tetrahedrons behind p are “inside the surface”, as Figure 3 shows.

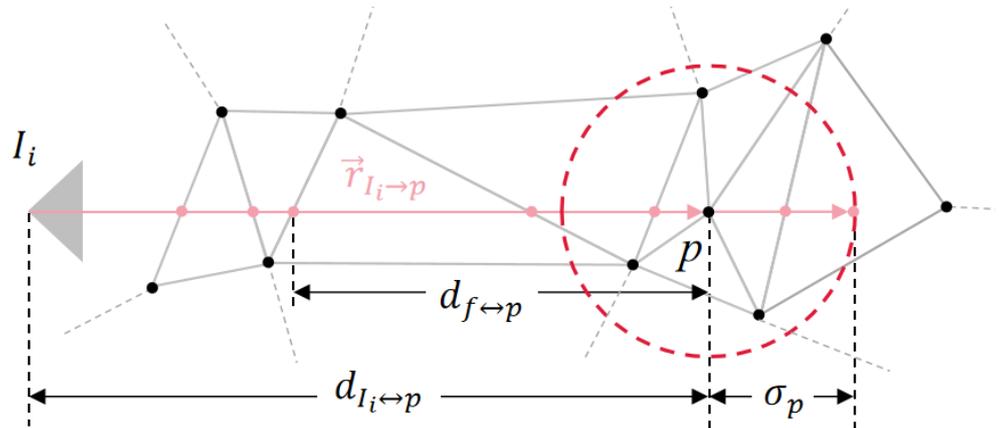


Figure 3. The visibility of a point. Given a point p and a visible image $I_i \in v_p$, we can construct a ray $\vec{r}_{I_i \rightarrow p}$ that traverses a series of tetrahedrons and intersects many faces in \mathcal{T} . To deal with the noise in p , we used the adaptive soft visibility by extending the ray with a distance σ_p to find the tetrahedron belonging to n_{in} . Meanwhile, we determined the weight of e_f based on the distance $d_{f \rightarrow p}$ from the intersection point of the ray and the face f to p .

However, there is noise in the position of the point from the dense point cloud; p is located near the surface, rather than precisely on the surface. To solve this problem, we used adaptive soft visibility by extending the ray with an adaptive distance σ_p for each point. The adaptive soft visibility is based on the fact that the importance m_p of each point $p \in \mathcal{P}$ is different in the process of reconstructing the mesh. Obviously, points that contain details, such as edges, are more important than those located in flat regions. Moreover, points with more visible images are more reliable than those with fewer visible images and contain less noise. Therefore, we calculated the importance m_p as Equation (1) shows, where \vec{n}_p represents the normal of p , $|v_p|$ is the size of v_p , and N_p is the points near p . The normal \vec{n}_p shows the direction of the point, and regions with large variations in the normal contain more details. However, since the normal is calculated by principal component analysis (PCA), it is sensitive to the noise in the point clouds. For this reason, we also introduced the size of the visible image v_p to reduce the effect of the noise in p since a p with higher $|v_p|$ generally has lower noise.

$$m_p = \frac{\sum_{p_n \in N_p} 1 - \vec{n}_p \vec{n}_{p_n}}{|v_p|(|v_p| - 2)^2} \tag{1}$$

Based on the importance of the point m_p , we can define the adaptive soft visibility σ_p , as shown in Equation (2), where σ is the median distance of the point clouds \mathcal{P} that roughly reflects the noise level [4,8]. The σ_p differs for each point and can find a better balance between noise and details.

$$\sigma_p = m_p \sigma \tag{2}$$

After determining the adaptive soft visibility σ_p , we can set the weight of each edge in the s-t graph \mathcal{G} , as Figure 3 shows. For $p \in P$ and a visible image $I_i \in v_p$, we built a ray $\vec{r}_{I_i \rightarrow p}$, which crosses a series of tetrahedrons and intersects the faces in \mathcal{T} . At the same time, we extended the ray $\vec{r}_{I_i \rightarrow p}$ by the distance σ_p to traverse another series of tetrahedrons

behind p . Assuming p is located on the surface, the tetrahedrons between I_i and p are “outside the surface” and the tetrahedrons between p and p_{σ_p} are “inside the surface”.

More precisely, we define the weight of three types of edges in \mathcal{G} as follows. Firstly, we directly set the tetrahedron t where the image I_i is located as “outside the surface” and set $e_{out \rightarrow t} = \infty$, which is because the image cannot be located “inside the surface”. Secondly, we set the tetrahedron t where p_{σ_p} is located as “inside the surface” and set $e_{out \rightarrow t} = |v_p|$, as the higher the $|v_p|$, the higher the likelihood that t belongs to “inside the surface”. Thirdly, we set the weight of e_f based on the distance $d_{f \leftrightarrow p}$ of the intersection point of face f and point p , as shown in Equation (3). The closer two tetrahedrons are to p , the more likely they belong to the same category.

$$e_f = (1 - \exp^{-\frac{d_{f \leftrightarrow p}^2}{2\sigma_p^2}}) |v_p| \tag{3}$$

After setting the weight of all edges in the s-t graph \mathcal{G} , we obtained the rough mesh M_r through the s-t graph-cut as Vu et al. [8] did. It should be noted that M_r still needs post-processing to improve the mesh quality, such as mesh smoothing, simplification, and hole filling.

2.2. Dynamic Photo-Metric Refinement

The rough mesh M_r obtained from the s-t graph-cut still suffers from noise and needs further optimization. Vu et al. [8] and Qu et al. [5] proposed an optimization algorithm based on photo-metric consistency by rendering a virtual image through triangles in the rough mesh M_r and comparing it with the ground truth image to optimize the geometry of M_r . However, they ignored that triangles in different regions have different convergence speeds and processed all triangles all the time, which is computationally inefficient. We propose a dynamic photo-metric refinement algorithm that analyses the convergence state of each triangle in M_r to speed up the refinement procedure.

Given a mesh M and a pair of images $I_i, I_j \in \mathcal{I}$, we can re-render a virtual image I_j^M in view of I_j based on I_i , as shown in Figure 4. Following Vu [8], we first built a ray for each pixel in I_j and intersecting with M to calculate the depth of the pixel, and then, we re-projected this pixel to the 3D space and projected it to I_i to sample the color. For the sake of the later introduction, we denote this procedure as \mathcal{R} , as shown in Equation (4).

$$I_j^M = \mathcal{R}(I_i, M) \tag{4}$$

After generating I_j^M , we calculated the photo-metric consistency between I_j and I_j^M . If M is closer to the ground truth (GT), the photo-metric consistency should be larger. For each pixel in I_j and I_j^M , it is unreliable to calculate the photo-metric consistency by directly comparing the difference of the color. Following Shen [2] and Vu [8], we set a window for each pixel and computed the photo-metric consistency using the ZNCC, as shown in Equation (5), where s is the size of the pixel window and $\bar{I}_j(p)$ and $\bar{I}_j^M(p)$ are the average color of the window around the pixel p in I_j and I_j^M , respectively. h ranges from 0 to 2, and a smaller h means a higher photo-metric consistency.

$$h(I_j, I_j^M) = 1 - \sum_{p \in I_j} \frac{\sum_{|p-x| <= s} (I_j(x) - \bar{I}_j(p))(I_j^M(x) - \bar{I}_j^M(p))}{\sqrt{\sum_{|p-x| <= s} (I_j(x) - \bar{I}_j(p))^2 \sum_{|p-x| <= s} (I_j^M(x) - \bar{I}_j^M(p))^2}} \tag{5}$$

Combining Equations (4) and (5), we can obtain the loss function \mathcal{L} , as shown in Equation (6). Since the whole process is differentiable, we can compute the gradient $\partial \mathcal{L} / \partial M$ to update the geometry of the mesh M to improve the photo-metric consistency.

$$\mathcal{L}_{I_i \rightarrow I_j}(M) = h(I_j, I_j^M) = h(I_j, \mathcal{R}(I_i, M)) \tag{6}$$

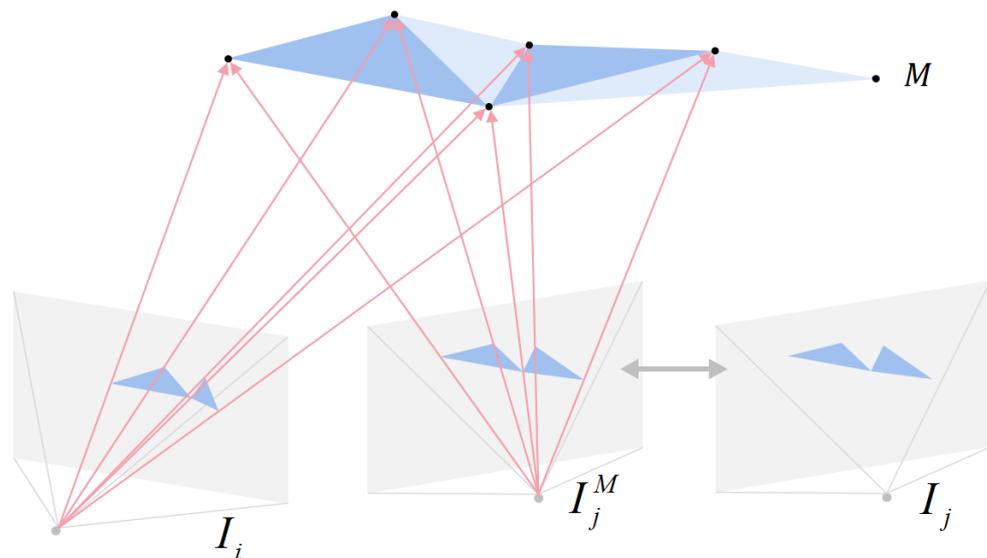


Figure 4. Photo-metric refinement. Given a mesh M and one pair of images I_i, I_j , we first use I_i to re-render a virtual image I_j^M in view of I_j , where M provides the depth value. We then optimize the geometry of M to maximize the photo-metric consistency between I_j and I_j^M . During the dynamic photo-metric refinement, we select partial triangles, i.e., blue triangles, to calculate photo-metric consistency and ignore light blue triangles.

Once the gradient is obtained from the photo-metric consistency, it is also important to know how to utilize the gradient to update M . Although the most-straightforward way is to use the steepest descent method, it requires a precise learning rate setting to prevent falling into a local minimum [5,8]. Considering this problem, we followed the optimization method in deep learning and used Adam [31] to update M , which dynamically adjusts the learning rate based on the current gradient and the historical gradient, as shown in Equation (7), where $t - 1, t$ is the iteration number, η is the learning rate, ϵ is set to the default value of 1×10^{-6} , and $\hat{\alpha}$ and $\hat{\beta}$ are the momentum and the second moment of the gradient, respectively.

$$M_t = M_{t-1} - \frac{\eta}{\sqrt{\hat{\beta} + \epsilon}} \hat{\alpha} = M_{t-1} - dM_{t-1} \tag{7}$$

Instead of using a fixed learning rate, Adam has a flexible optimization procedure, as Equation (8) shows, where λ_α and λ_β are set to default values of 0.9 and 0.999, respectively. During optimization, Adam dynamically adjusts the learning rate of each triangle in M , thus making refinement more efficient and accurate.

$$\begin{cases} \alpha_t &= \lambda_\alpha \alpha_{t-1} + (1 - \lambda_\alpha) \frac{\partial \mathcal{L}}{\partial M} \\ \beta_t &= \lambda_\beta \beta_{t-1} + (1 - \lambda_\beta) \frac{\partial \mathcal{L}}{\partial M}^2 \\ \hat{\alpha} &= \frac{\alpha_t}{1 - \lambda_\alpha} \\ \hat{\beta} &= \frac{\beta_t}{1 - \lambda_\beta} \end{cases} \tag{8}$$

Except for the learning rate, another issue in photo-metric refinement is that each triangle needs to calculate the photo-metric consistency in each iteration. However, due to the inconsistent convergence speed of different triangles, some triangles may be converged while others still need further optimization. To address this problem, we used an early stopping strategy, which determines whether a triangle needs optimization based on changes between the current position dM_t and the last position dM_{t-1} , as Equation (9) shows, where T_d is the stopping optimization threshold. If all three vertices of a triangle

satisfy Equation (9), we considered that this triangle has converged and stopped the optimization of it.

$$dM_t \leq dM_{t-1}T_d \quad (9)$$

In addition, we employed a coarse-to-fine optimization strategy to improve the convergence speed and prevent falling into the local minimum. Meanwhile, we subdivided the mesh to add triangles in M to facilitate the representation of more details.

3. Experiments

3.1. Datasets

We conducted quantitative and qualitative experiments to verify the effectiveness and reliability of the proposed algorithm. The quantitative experiment used ten scenes from the BlendedMVS dataset [28] with an image resolution of 2048×1536 , including five aerial scenes and five close-range scenes, as Figures 5 and 6 show. The qualitative experiment used two real-world datasets: one aerial scene, P36, with an image resolution of 4533×3016 captured by an unmanned aerial vehicle (UAV) from Pix4D [32] and one close-range scene, P146, captured by a hand-held camera with an image resolution of 4016×2005 , as Figure 7 shows. BlendedMVS provides the ground truth intrinsic and extrinsic camera parameters and the 3D models of the scene, and the real-world dataset uses accurate intrinsic and extrinsic parameters provided by COLMAP [1]. To avoid the influence of dense point clouds, we uniformly used the same MVS method to calculate the dense point cloud [2,3] during the evaluation for all mesh-reconstruction methods.

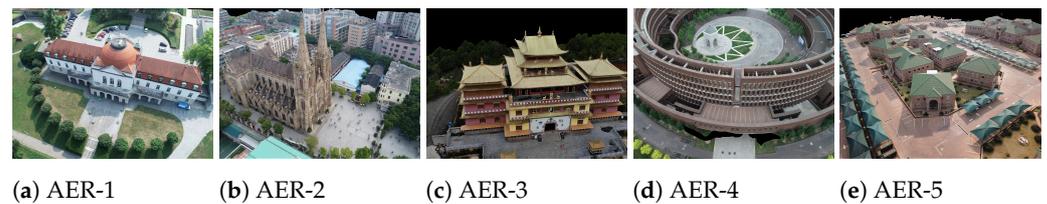


Figure 5. Images of five aerial scenes. We selected five aerial scenes from BlendedMVS [28] that capture buildings. We show one image from each scene, where the black pixels indicate areas outside the reconstruction range.



Figure 6. Images of five close-range scenes. We selected five close-range scenes from BlendedMVS [28] that capture sculptures. We visualize one image in each scene, where the invisible region is black.

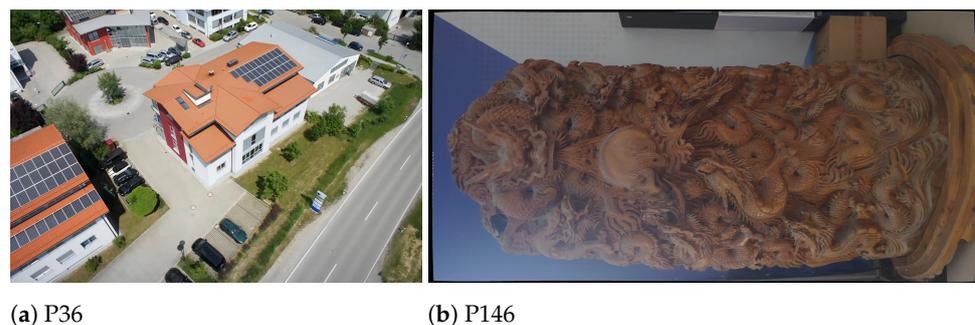


Figure 7. Images of two real-world scenes. We selected two real-world scenes for the quantitative evaluation, including one aerial scene and one close-range scene.

3.2. Implementation

We implemented our method in C++ in Visual Studio 2015. For all experiments, we used a personal computer with Windows 10, which has a 3.2 GHz i7-8700 central processing unit (CPU) and 64G of random access memory (RAM). We used the same hyper-parameters for all experiments and set $s = 5$, $T_d = 0.01$, $\eta = 1.0$.

3.3. Evaluation Metrics

In the qualitative evaluation, we compared our method with two state-of-the-art methods, OpenMVS [29], which is an open-source implementation of Vu et al. [8], and TDR [5]. For the ease of presentation, we note that V_r and V_f are the mesh reconstruction results from OpenMVS in the visibility reconstruction and photo-metric refinement, Q_r the mesh reconstruction results from TDR in the photo-metric refinement, and M_r and M_f the mesh reconstruction results from our method in adaptive visibility reconstruction and dynamic photo-metric refinement. The qualitative evaluation consisted of two metrics: reconstruction efficiency E and reconstruction quality Q .

Reconstruction efficiency E measures the time a method needs to reconstruct the mesh from dense point clouds. The less time a method requires, the more efficient it is. Typically, reconstruction efficiency is measured in seconds.

Generally, we calculated the distance D between the reconstructed mesh and the ground truth mesh to measure the reconstruction quality Q . However, the distance between the two meshes is meaningful since the mesh provided by BlendedMVS lacks an absolute scale. Therefore, we measured the reconstruction quality Q by comparing the distance changes with V_r , as Equation (10) shows, where G is the ground truth mesh and M is the mesh to evaluate. The closer between M and G , the higher Q . We used CloudCompare [33] to calculate the distance D between two meshes, which randomly samples the points on the mesh and calculates the average distance between these points.

$$Q(M) = \frac{D(GT, V_r) - D(GT, M)}{D(GT, V_r)} \quad (10)$$

3.4. Aerial Scenes

Table 1 shows the quantitative results for the five aerial scenes. In terms of reconstruction quality, V_f , Q_f , M_r , and M_f had higher reconstruction quality than V_r . V_f had better reconstruction quality than V_r , with the reconstruction quality Q ranging from 0.76 to 4.65, which is unsurprising since photo-metric refinement can improve the mesh quality. Meanwhile, the rough reconstruction results of our method, M_r , had a higher reconstruction quality than V_r , with the reconstruction quality Q ranging from 0.50 to 1.60, which proved the effectiveness of the adaptive soft visibility. Our method utilized the adaptive soft visibility to reconstruct a higher-quality rough mesh, while OpenMVS [8,29] lost many details of the scene by directly using fixed soft visibility for all points. In addition, the dynamic mesh refinement further improved the reconstruction quality. M_f had higher reconstruction quality than V_f in four scenes, with an improvement between 0.10 and 0.83. The reconstruction quality of M_f was only slightly lower than V_f on AEA-1. Moreover, regarding reconstruction efficiency, our method was much better than OpenMVS on all scenes. Although the reconstruction efficiency of both our method and OpenMVS decreased as the number of images increased, our method required only 34% to 43% of the reconstruction time of OpenMVS. Compared to TDR [5], our method achieved comparable reconstruction quality, but only needed 3% to 14% of the reconstruction time of TDR. Although TDR achieved the highest reconstruction quality among all methods, its reconstruction efficiency was too low to be used in practical applications.

Table 1. Quantitative experimental results on five aerial scenes.

Dataset	Number	OpenMVS [8,29]		TDBI [5]		$E(M_f)$	Ours	
		$E(V_f)$	$Q(V_f)$	$E(Q_f)$	$Q(Q_f)$		$Q(M_r)$	$Q(M_f)$
AER-1	77	1708	0.76	7245	4.56	736	0.50	0.59
AER-2	125	2958	4.65	13,016	13.67	1220	1.60	5.47
AER-3	132	3140	3.62	8549	7.08	1189	1.37	3.71
AER-4	149	3516	2.07	17,968	8.11	1457	1.08	2.47
AER-5	186	6125	1.14	67,568	5.98	2106	1.08	1.97

To further compare the different methods, we visualize the reconstruction results of the five aerial scenes in Figure 8 and zoom in on some regions to compare the reconstruction details in Figure 9. It struck us that, although TDR [5] achieved the highest reconstruction quality, the quality of its edge areas was poor, and it could not reconstruct the thin structure of buildings. Compared with OpenMVS and TDR, our method successfully reconstructed the thin structures of buildings and recovered more details of the scene. Our approach is more suitable for 3D reconstruction tasks in urban areas containing a large number of man-made buildings.

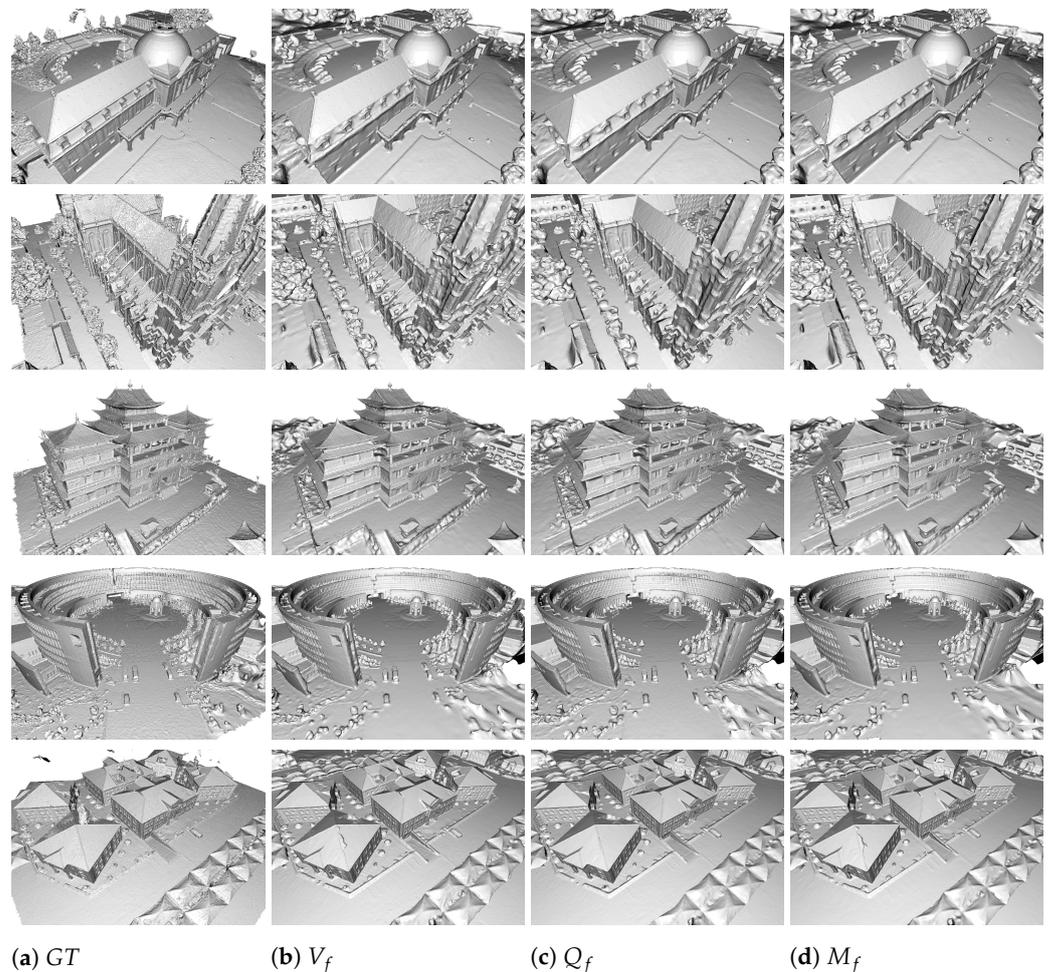


Figure 8. Results on aerial scenes. We show the reconstruction results from AER-1, AER-2, AER-3, AER-4, and AER-5 from top to bottom. We visualize the ground truth mesh provided by BlendMVS [28] in (a), as well as the results from the state-of-the-art methods OpenMVS [8,29] in (b) and TDR [5] in (c). (d) is the results from our method. Overall, our method achieved comparable reconstruction quality to OpenMVS and TDR, but the reconstruction efficiency of our method was much higher.

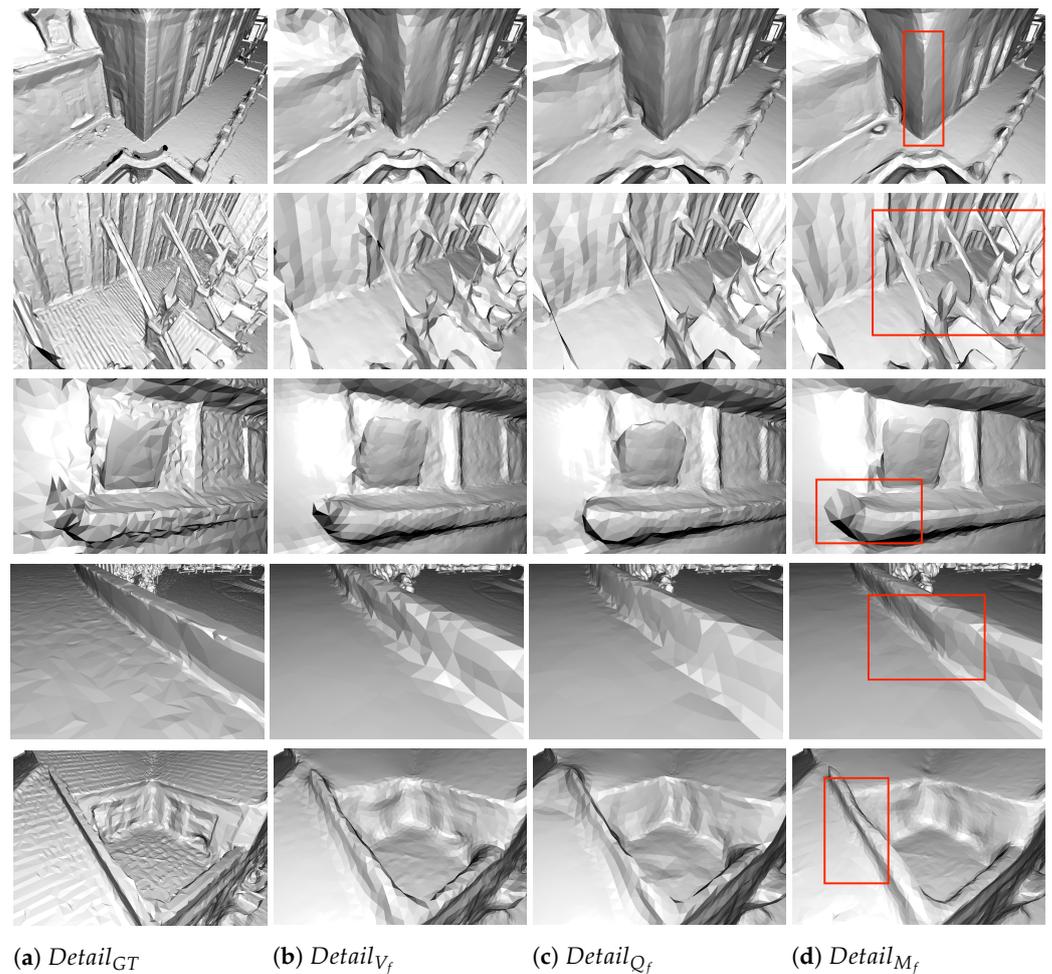


Figure 9. Details on aerial scenes. In order to clearly compare the differences between methods, we show the details of the reconstruction results from AER-1, AER-2, AER-3, AER-4, and AER-5 from top to bottom. (a) $Detail_{GT}$ shows the details from the ground truth mesh. (b) $Detail_{V_f}$ shows the results from OpenMVS [8,29]. (c) $Detail_{Q_f}$ shows the results from TDR [5]. (d) $Detail_{M_f}$ shows the results from our method. Compared with OpenMVS and TDR, our method can reconstruct more sharp edges of the scene. We use red boxes to mark areas where our method achieved better results.

In conclusion, in terms of reconstruction quality, our method is comparable to OpenMVS [8,29] and TDR [5], but significantly improved the reconstruction efficiency, and it better meets the needs of practical applications.

3.5. Close-Range Scenes

Table 2 shows the quantitative reconstruction results for the five close-range scenes. In terms of reconstruction quality, all methods had higher quality than V_r . Compared with V_r , the reconstruction quality of M_r ranged from 0.46 to 1.92, and the reconstruction quality of M_f ranged from 1.43 to 10.05. Meanwhile, M_f showed a more-significant improvement over V_f compared to aerial scenes, which was 1.12 to 2.93 higher than V_f . The main reason for this improvement is that the observation of the image was clearer in the close-range case, and the photo-metric consistency can provide more-reliable gradient information to adjust the learning rate and select triangles. In terms of reconstruction efficiency, our method still outperformed OpenMVS [8,29] in all scenes, requiring only about 39% to 49% of the reconstruction time of OpenMVS. Compared with TDR [5], our method achieved competitive reconstruction qualitatively on four scenes while obtaining better reconstruction quality on CLO-5. Overall, the reconstruction quality of TDR only

improved a little over our method, but the reconstruction efficiency of TDR was so low that our method only needed 0.3% to 8% of its reconstruction time.

Table 2. Quantitative experimental results on close-range scenes.

Dataset	Number	OpenMVS [8,29]		TDBI [5]		$E(M_r)$	Ours $Q(M_r)$	$Q(M_f)$
		$E(V_f)$	$Q(V_f)$	$E(V_f)$	$Q(V_f)$			
CLO-1	51	1073	8.70	20,680	17.52	497	0.65	10.05
CLO-2	64	1172	0.03	155,636	6.8	466	0.57	1.43
CLO-3	91	1820	0.58	9893	4.11	834	1.92	2.81
CLO-4	100	2286	4.72	28,340	8.52	888	1.57	7.65
CLO-5	117	2078	2.84	46,047	1.13	794	0.46	3.96

We show the reconstruction results in Figure 10 and the details of the reconstruction results in Figure 11. Our method also reconstructed more thin structures of the scenes, especially in CLO-2, where OpenMVS and TDR failed to rebuild the walking stick. Considering the fact that TDR obtained the best reconstruction quality, the reason may be that it pays too much attention to the flat regions while ignoring the boundary regions, which may point the way to improve our approach.

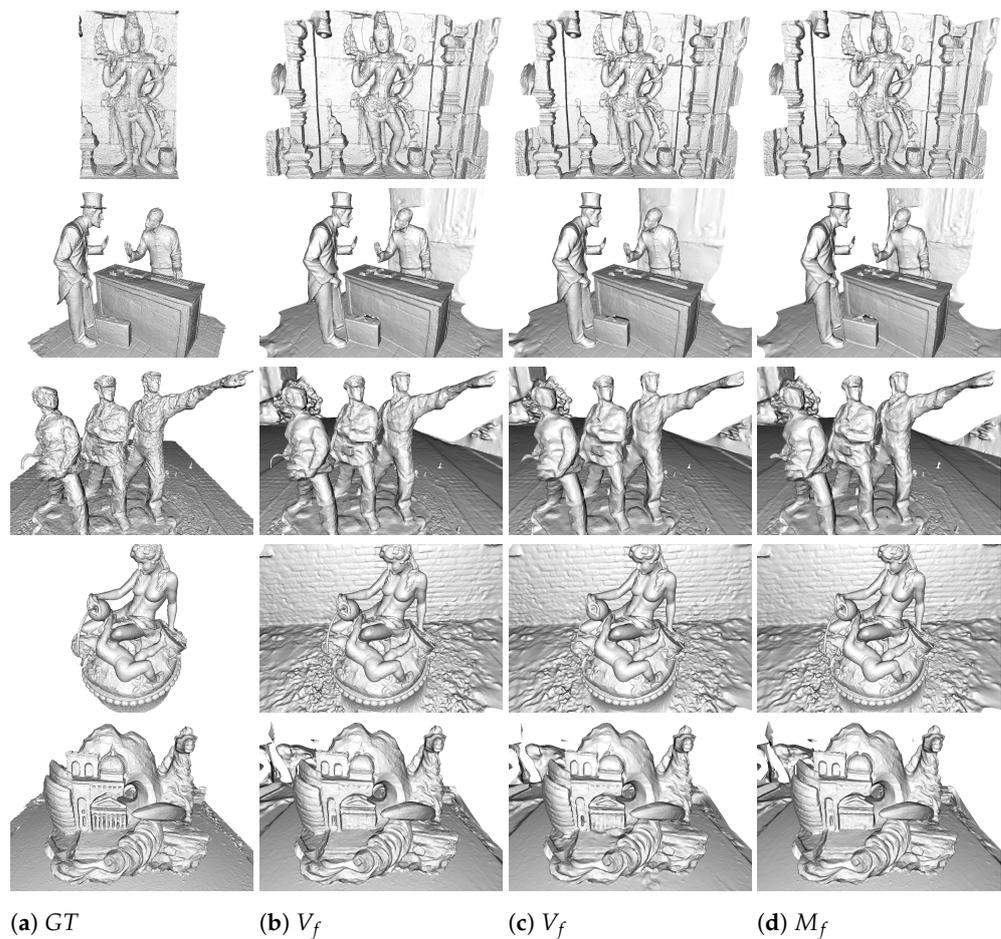


Figure 10. Results on the close-range scenes. From top to bottom, we visualize the reconstruction results of CLO-1, CLO-2, CLO-3, CLO-4, and CLO-5. We show the ground truth mesh of Blended-MVS [28] in (a) and compare the results of the state-of-the-art method OpenMVS [8,29] in (b) and TDR [5] in (c) with our method in (d). As the reconstruction range of OpenMVS, TDR, and our method is larger than the ground truth, we excluded regions that were out of scope during the evaluation.

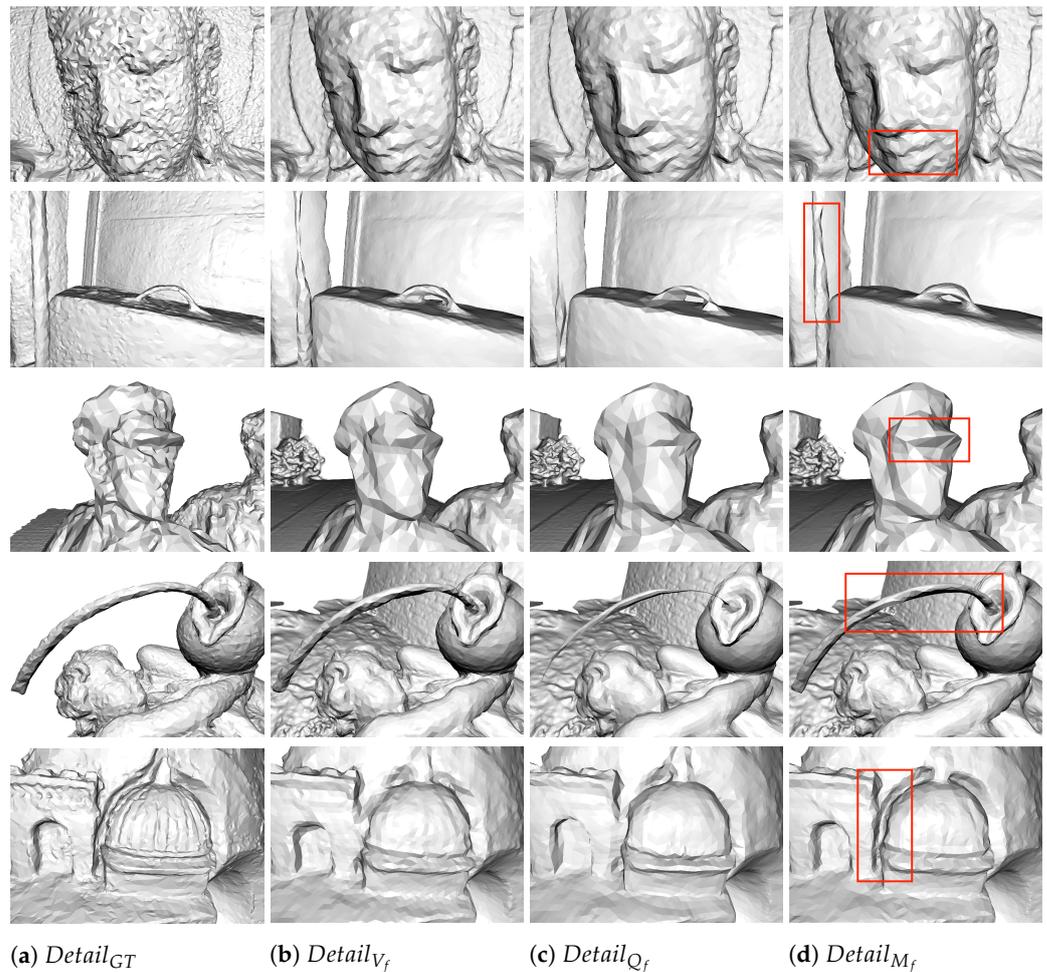


Figure 11. Details on the close-range scenes. From top to bottom, we visualize the details of reconstruction from CLO-1, CLO-2, CLO-3, CLO-4, and CLO-5, where (a) $Detail_{GT}$ shows the ground truth mesh, (b) $Detail_{V_f}$ shows the results from OpenMVS [8,29], (c) shows the results from TDR [5], and (d) shows the results from our method. On close-range scenes, our method can significantly retain more details. We use red boxes to mark areas where our method achieved better results.

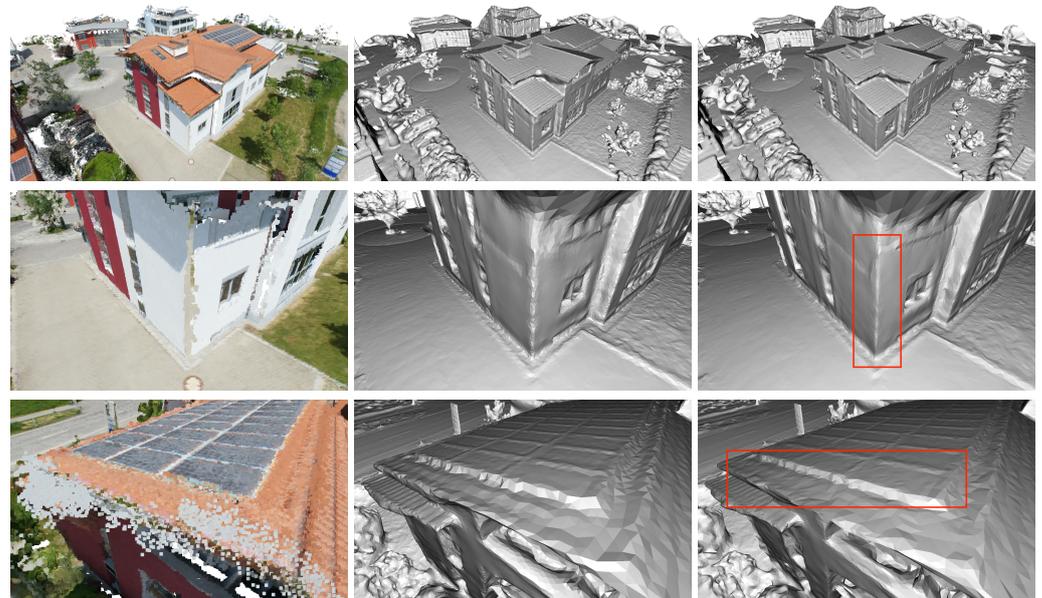
Overall, our method found a good balance between the reconstruction quality and reconstruction efficiency in aerial images and close-range scenes, while OpenMVS and TDR needed more reconstruction time with comparable reconstruction quality.

3.6. Real-World

To fully validate the effectiveness of our method, we also evaluated our method and OpenMVS [8,29] on two real-world datasets. Due to the lack of ground truth meshes, we only compared the reconstruction efficiency with OpenMVS, as shown in Table 3. Consistent with the results of BlendedMVS, our method had higher reconstruction efficiency on real-world datasets, taking only one-third of the reconstruction time of OpenMVS. We further qualitatively compare them in Figure 12, which shows the dense point cloud from MVS and the reconstructed mesh. We show the details of some regions, and while both OpenMVS and our method were effective at dealing with noise in the dense point cloud, the OpenMVS generated an over-smoothed mesh, while our method reconstructed the mesh with more details, especially in the noise-filled regions.

Table 3. Quantitative experimental results on real-world scenes.

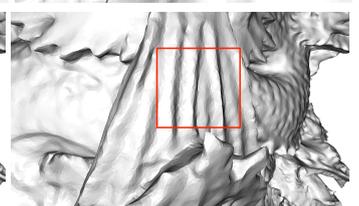
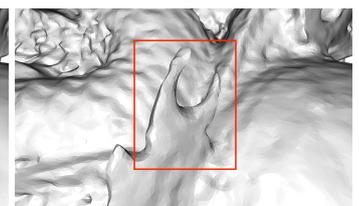
Dataset	Number	OpenMVS [8,29] $E(V_f)$	Ours $E(M_f)$
P36	36	4026	1698
P146	146	10,035	3429



(a) P36-Cloud

(b) P36- V_f

(c) P36- M_f



(d) P146-Cloud

(e) P146- V_f

(f) P146- M_f

Figure 12. Results of P36 and P146. On real-world datasets, we compare results of different mesh reconstruction methods with the same dense point cloud, and our method can maintain more details. For each scene, we visualize the 3D scene in the first row and the details in the following two rows, where (a) and (d) show dense point clouds from P36 and P146, respectively, (b,e) show the results from OpenMVS [8,29] on two scenes, and (c,f) are the results from our method. OpenMVS and our method can handle the noise in the dense point cloud, but OpenMVS reconstructs smoother meshes. We use red boxes to mark areas where our method achieves better results.

4. Conclusions

This paper proposed an improved optimization-based mesh reconstruction method that balances reconstruction quality and reconstruction efficiency. Our method first reconstructs high-quality rough meshes that preserve details and suppress noise by analyzing point quality and importance. Then, our method employs dynamic photo-metric refinement to speed up the convergence by changing the learning rate and stopping the optimization of converged triangles. Extensive experiments on both BlendedMVS and real-world datasets demonstrated that our method outperformed the state-of-the-art OpenMVS in reconstruction quality while requiring only one-third of the time and achieved competitive reconstruction quality with TDR with only one-tenth of the time. In the future, we plan to extend our method to city-scale reconstruction tasks and further optimize the efficiency of our method to meet the application requirements in scenarios such as emergency rescue.

Author Contributions: Conceptualization, Q.Y.; methodology, Q.Y.; writing—original draft preparation, Q.Y.; writing—review and editing, Y.Q., T.X., J.Y. and F.D.; funding acquisition, T.X. and F.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundation of China (No. 42301491) and the Hubei Key Research and Development Project (No. 2022BAA035).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: BlendedMVS can be obtained from <https://github.com/YoYo000/BlendedMVS> (accessed on 10 September 2023). The P36 dataset can be obtained from <https://doc.arcgis.com/en/drone2map/latest/get-started/sample-data.htm> (accessed on 10 September 2023). The P146 dataset is available from the corresponding author upon reasonable request.

Acknowledgments: The authors are grateful to the providers of the BlendedMVS dataset and the P36 dataset. We would also like to thank the researchers who published open-source code or programs, including OpenMVS, COLMAP, and CloudCompare.

Conflicts of Interest: Teng Xiao and Fei Deng were employed by Wuhan Tianjihang Information Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Schonberger, J.L.; Frahm, J.-M. Structure-from-motion revisited. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4104–4113. [CrossRef]
2. Shen, S. Accurate multiple view 3d reconstruction using patch-based stereo for large-scale scenes. *IEEE Trans. Image Process.* **2013**, *22*, 1901–1914. [CrossRef] [PubMed]
3. Fei, D.; Qingsong, Y.; Teng, X. A GPU-PatchMatch multi-view dense matching algorithm based on parallel propagation. *Acta Geod. Cartogr. Sin.* **2020**, *49*, 181–190. [CrossRef]
4. Zhou, L.; Zhang, Z.; Jiang, H.; Sun, H.; Bao, H.; Zhang, G. DP-MVS: Detail Preserving Multi-View Surface Reconstruction of Large-Scale Scenes. *Remote. Sens.* **2021**, *13*, 4569. [CrossRef]
5. Qu, Y.; Yan, Q.; Yang, J.; Xiao, T.; Deng, F. Total Differential Photometric Mesh Refinement with Self-Adapted Mesh Denoising. *Photonics* **2022**, *10*, 20. [CrossRef]
6. Waechter, M.; Moehrle, N.; Goesele, M. Let there be color! Large-scale texturing of 3D reconstructions. In *Computer Vision—ECCV 2014, Proceedings of the 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Part V 13*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 836–850. [CrossRef]
7. Kazhdan, M.; Chuang, M.; Rusinkiewicz, S.; Hoppe, H. Poisson surface reconstruction with envelope constraints. *Comput. Graph. Forum* **2020**, *39*, 173–182. [CrossRef]
8. Vu, H.H.; Labatut, P.; Pons, J.P.; Keriven, R. High accuracy and visibility-consistent dense multiview stereo. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *34*, 889–901. [CrossRef] [PubMed]
9. Labatut, P.; Pons, J.P.; Keriven, R. Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts. In Proceedings of the 2007 IEEE 11th International Conference on Computer Vision, Rio De Janeiro, Brazil, 14–21 October 2007; pp. 1–8. [CrossRef]
10. Vogiatzis, G.; Esteban, C.H.; Torr, P.H.; Cipolla, R. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 2241–2246. [CrossRef] [PubMed]

11. Sinha, S.N.; Mordohai, P.; Pollefeys, M. Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In Proceedings of the 2007 IEEE 11th International Conference on Computer Vision, Rio De Janeiro, Brazil, 14–21 October 2007; pp. 1–8. [[CrossRef](#)]
12. Labatut, P.; Pons, J.P.; Keriven, R. Robust and efficient surface reconstruction from range data. *Comput. Graph. Forum* **2009**, *28*, 2275–2290. [[CrossRef](#)]
13. Jancosek, M.; Pajdla, T. Multi-view reconstruction preserving weakly-supported surfaces. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; pp. 3121–3128.
14. Jancosek, M.; Pajdla, T. Exploiting visibility information in surface reconstruction to preserve weakly supported surfaces. *Int. Sch. Res. Not.* **2014**, *2014*, 798595. [[CrossRef](#)] [[PubMed](#)]
15. Zhou, Y.; Shen, S.; Hu, Z. Detail preserved surface reconstruction from point cloud. *Sensors* **2019**, *19*, 1278. [[CrossRef](#)] [[PubMed](#)]
16. Labatut, P.; Pons, J.P.; Keriven, R. Hierarchical shape-based surface reconstruction for dense multi-view stereo. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, Kyoto, Japan, 27 September–4 October 2009; pp. 1598–1605. [[CrossRef](#)]
17. Li, S.; Yao, Y.; Fang, T.; Quan, L. Reconstructing thin structures of manifold surfaces by integrating spatial curves. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2887–2896. [[CrossRef](#)]
18. Hofer, M.; Maurer, M.; Bischof, H. Efficient 3D scene abstraction using line segments. *Comput. Vis. Image Underst.* **2017**, *157*, 167–178. [[CrossRef](#)]
19. Li, S.; Siu, S.Y.; Fang, T.; Quan, L. Efficient multi-view surface refinement with adaptive resolution control. In *Computer Vision—ECCV 2016, Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 349–364. [[CrossRef](#)]
20. Zhang, C.; Zhang, M.; Guo, B.; Peng, Z. Adaptive Fast Mesh Refinement of 3D Reconstruction Based on Image Information. *Geomat. Inf. Sci. Wuhan Univ.* **2020**, *45*, 411–418.
21. Morreale, L.; Romanoni, A.; Matteucci, M. Predicting the next best view for 3d mesh refinement. In *IAS 2018: Intelligent Autonomous Systems 15, Proceedings of the 15th International Conference IAS-15, Baden-Baden, Germany, 11–15 June 2018*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 760–772. [[CrossRef](#)]
22. Yan, Z.; Qingsong, Y.; Yingjie, Q.; Xin, C.; Fei, D. View selection strategy for photo-consistency refinement. *Acta Geod. Cartogr. Sin.* **2020**, *49*, 1463–1472. [[CrossRef](#)]
23. Romanoni, A.; Matteucci, M. Facetwise Mesh Refinement for Multi-View Stereo. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 6794–6801. [[CrossRef](#)]
24. Blaha, M.; Rothermel, M.; Oswald, M.R.; Sattler, T.; Richard, A.; Wegner, J.D.; Pollefeys, M.; Schindler, K. Semantically informed multiview surface refinement. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 3819–3827. [[CrossRef](#)]
25. Romanoni, A.; Ciccone, M.; Visin, F.; Matteucci, M. Multi-view stereo with single-view semantic mesh refinement. In Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 22–29 October 2017; pp. 706–715. [[CrossRef](#)]
26. Fei, D.; Xin, C.; Qingsong, Y.; Yingjie, Q. Variational refinement of mesh with line constraint for photogrammetry. *Acta Geod. Cartogr. Sin.* **2020**, *49*, 469–479. [[CrossRef](#)]
27. Romanoni, A.; Matteucci, M. Mesh-based camera pairs selection and occlusion-aware masking for mesh refinement. *Pattern Recognit. Lett.* **2019**, *125*, 364–372. [[CrossRef](#)]
28. Yao, Y.; Luo, Z.; Li, S.; Zhang, J.; Ren, Y.; Zhou, L.; Fang, T.; Quan, L. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, DC, USA, 13–19 June 2020; pp. 1790–1799. [[CrossRef](#)]
29. Cernea, D. OpenMVS: Multi-View Stereo Reconstruction Library. Available online: <https://cdcseacave.github.io/openMVS> (accessed on 1 October 2023).
30. Merrell, P.; Akbarzadeh, A.; Wang, L.; Mordohai, P.; Frahm, J.M.; Yang, R.; Nistér, D.; Pollefeys, M. Real-time visibility-based fusion of depth maps. In Proceedings of the 2007 IEEE 11th International Conference on Computer Vision, Rio De Janeiro, Brazil, 14–21 October 2007; pp. 1–8. [[CrossRef](#)]
31. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
32. Christoph, S. Pix4D. Available online: <https://www.pix4d.com> (accessed on 1 October 2023).
33. CloudCompare. CloudCompare: 3D Point Cloud and Mesh Processing Software Open Source Project. Available online: <https://www.cloudcompare.org> (accessed on 1 October 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.