

Article

# Total Differential Photometric Mesh Refinement with Self-Adapted Mesh Denoising

Yingjie Qu <sup>1</sup>, Qingsong Yan <sup>1</sup>, Junxing Yang <sup>2</sup>, Teng Xiao <sup>3,4</sup> and Fei Deng <sup>1,4,\*</sup><sup>1</sup> School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China<sup>2</sup> School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture, Beijing 102616, China<sup>3</sup> School of Computer Science, Hubei University of Technology, Wuhan 430068, China<sup>4</sup> Wuhan Tianjihang Information Technology Co., Ltd., Wuhan 430010, China

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

**Abstract:** Variational mesh refinement is a crucial step in multiview 3D reconstruction. Existing algorithms either focus on recovering mesh details or focus on suppressing noise. Approaches with consideration of both are lacking. To address this limitation, we proposed a new variational mesh refinement method named total differential mesh refinement (TDR), which mainly included two improvements. First, the traditional partial-differential photo-consistency gradient used in the variational mesh refinement method was replaced by the proposed total-differential photo-consistency gradient. With consideration of the photo-consistency correlation between adjacent pixels, our method can make photo-consistency achieve a more effective convergence than traditional approaches. Second, we introduced the bilateral normal filter with a novel self-adaptive mesh denoising strategy into the variational mesh refinement. This strategy maintains a balance between detail preservation and effective denoising via the zero-normalized cross-correlation (ZNCC) map. Various experiments demonstrated that our method is superior to traditional variational mesh refinement approaches in both accuracy and denoising effect. Moreover, compared with the mesh generated by open-source and commercial software (Context Capture), our meshes are more detailed, regular, and smooth.

**Keywords:** variational mesh refinement; photometric mesh refinement; surface reconstruction; mesh denoising; photogrammetry

**Citation:** Qu, Y.; Yan, Q.; Yang, J.; Xiao, T.; Deng, F. Total Differential Photometric Mesh Refinement with Self-Adapted Mesh Denoising. *Photonics* **2023**, *10*, 20.

<https://doi.org/10.3390/photonics10010020>

Received: 26 October 2022;

Revised: 20 December 2022;

Accepted: 21 December 2022;

Published: 24 December 2022



**Copyright:** © 2022 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

A well-established pipeline of image-based 3D reconstruction technology mainly includes structure from motion (SfM), multiview stereo (MVS), surface reconstruction, mesh refinement, and texture. In this pipeline, the surface reconstruction step reconstructs an initial coarse 3D mesh that may lack details and contain noise due to occlusion and non-Lambertian materials. Photometric stereo [1,2] and mesh refinement are popular methods to reconstruct high-quality mesh shapes. Photometric stereo recovers pixel-wise surface normals from a fixed scene under varying shading cues, which are widely used in the industrial field [3–5]. The mesh refinement method evolves the initial mesh to fine details using multiview images. In this paper, we mainly talk about the mesh refinement method for large-scale reconstruction. Variational mesh refinement is the most commonly used refinement method, and it improves mesh details and accuracy by iteratively updating all vertex positions to maximize the photo-consistency between images [6]. However, due to its isotropic regularization term, this method tends to smooth sharpened structures and cannot sufficiently remove excessive mesh noises in texture-less and non-Lambertian regions.

To solve this problem, Li et al. [7] used a content-aware mesh denoising approach as a regularization term for mesh refinement, which was effective in suppressing mesh noise while preserving sharp features. However, without the assistance of image information, some distinct errors and noises of the mesh may be wrongly identified as sharp features and preserved.

Other studies aimed to improve the accuracy of the variational mesh refinement: some researchers selected the best image pairs to refine the mesh [8]. Blaha et al. [9] and Romanoni et al. [10] used semantic information to improve mesh accuracy. However, the improvement is limited when there are few images and little semantic information. In addition, existing variational mesh refinement methods calculate the gradient of each pixel independently and do not consider the photo-consistency of neighboring pixels.

In this study, a total differential mesh refinement (TDR) approach was developed to address the abovementioned problems. First, we proposed to use the total-differential photo-consistency gradient (TDPG) to replace the partial-differential photo-consistency gradient (PDPG) calculation in the variational mesh refinement approach. The TDPG considers the influence of the gradient between adjacent pixels so that the photo-consistency can obtain more effective convergence than the PDPG via gradient descent. Second, we incorporated bilateral normal filtering [11] into the variational mesh refinement to improve the denoising and edge-preserving capabilities. A self-adaptive mesh denoising strategy was adopted to balance detail preservation and effective denoising. Specifically, the zero-normalized cross-correlation (ZNCC), which measures the photo-consistency in the image domain, was transformed to mesh vertices to form a ZNCC map indicative of the uncertainty of the mesh vertices. Then, denoising gradients are self-adaptive and weighted depending on their ZNCC. Thus, our method can remove mesh noise while preserving the details of the mesh. An overview of our method is shown in Figure 1.

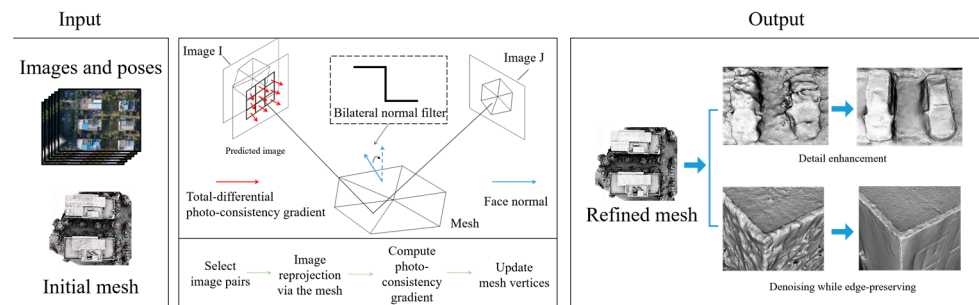


Figure 1. The flow diagram for the proposed TDR algorithm.

Our contributions are as follows:

We proposed the TDPG method, which considers the partial derivative of all pixels in the neighborhood, makes the photo-consistency error converge to a low level, and obtains a fine-details mesh model.

We introduced the bilateral normal filtering [11] to the variational mesh refinement and adopted the self-adaptive mesh denoising strategy that utilized a ZNCC map to guide mesh denoising. This strategy enabled effective denoising while preserving mesh details (Section 2.3).

We used photo-consistency information to guide mesh denoising, which provided a new idea for the study of feature-preserving denoising.

## 2. Methodology

We enhanced the variational mesh refinement from two aspects. First, the total-differential photo-consistency gradient (TDPG) calculation was proposed for more effective convergence of the photo-consistency. Second, bilateral normal filtering was utilized in mesh refinement for mesh denoising, flattening planes, and sharpening edges. In order to avoid the loss of details that may be caused by denoising, we proposed the self-adaptive

mesh denoising strategy. In this section, we first briefly introduce the variational mesh refinement approach (Section 2.1). Then, we present our mesh refinement method (TDR), including TDPG calculation (Section 2.2) and self-adaptive mesh denoising (Section 2.3).

### 2.1. Preliminaries on Variational Mesh Refinement

The variational mesh refinement approach was introduced by Pons et al. [12] and expanded by Vu et al. [13]. This method minimizes the photo-consistency error between image pairs by iteratively refining the mesh vertex. For one image pair  $I_p$  and  $I_q$ , image  $I_q$  can be projected onto the mesh  $S$  and then reprojected to image  $I_p$  to form a predicted image  $I_{pq}^S$  [7,14]. The photo-consistency between the predicted image and the reference image measures the correctness of the mesh, and the goal of variational mesh refinement is to minimize this photo-consistency error between all image pairs.

The energy function is expressed as:

$$E(S) = E_{photo}(S) + E_{regularization}(S) \tag{1}$$

$S$  is the mesh surface,  $E(S)$  is the total energy function,  $E_{regularization}(S)$  enforces the smoothness of the surface. The  $E_{photo}(S)$  is defined as:

$$E_{photo}(S) = - \sum_{p,q} \int_{\Omega_{pq}^S} M_{zncc}(I_p, I_{pq}^S)(x_i) dx_i \tag{2}$$

$M_{zncc}(I_p, I_{pq}^S)(x_i)$  is the ZNCC measurement between images  $I_p$  and  $I_{pq}^S$  at pixel  $x_i$ .  $\Omega_{pq}^S$  is the map of the reprojection from image  $I_p$  to image  $I_q$  via the surface.

To minimize  $E_{photo}(S)$ , the gradient is calculated using chain rules (see [13]):

$$g_{photo}(V) = \frac{dE_{photo}(S)}{dV} = \sum_{p,q} \int_{\Omega_{pq}^S} \phi(X_i) \partial M(x_i) DI_q(x_j) D\Pi_q(X_i) \frac{d_i}{N^T d_i} N dx_i \tag{3}$$

$V$  represents all mesh vertices,  $g_{photo}(V)$  is the vertices gradient induced by the photo in each iteration.  $M$  is the abbreviation for  $M_{zncc}(I_p, I_{pq}^S)$ ,  $\partial M(x_i)$  is photo-consistency gradient at pixel  $x_i$ .  $X_i$  is the intersection of  $S$  and the ray from the camera center of  $I_p$  to  $x_i$ .  $\phi(X_i)$  is the barycentric coordinate weight of  $X_i$ .  $\Pi_q$  represents the projection from the world to the  $I_q$ ,  $x_j = \Pi_q(X_i)$  and  $DI_q(x_j)$  is image gradient at  $x_j$ .  $D$  denotes the Jacobian matrix of a function.  $d_i$  is the vector joining the camera center of  $I_p$  and point  $X_i$ .  $N$  is the outward surface normal at  $X_i$ .

For mesh regularization, this method adds thin plate energy [15] that penalizes mesh bending to prevent excessive bending of the mesh and excessive deviation of the gradient flow.

$$E_{regularization}(S) = \int_S (k_1^2 + k_2^2) dS \tag{4}$$

where  $k_1$  and  $k_2$  are the principal curvatures of the mesh. The linear combination of Laplacian and Bi-Laplacian operators minimizes this energy [15].

### 2.2. Total-Differential Photo-Consistency Gradient Calculation

A key step in variational mesh refinement is the calculation of the vertex gradient (Equation (3)), which determines how the vertices are updated to minimize photo-consistency error. In the baseline method [13], the photo-consistency gradient ( $\partial M$  in (Equation (3))) is calculated separately at each pixel and the potential contradictions between pixel gradients are ignored. In our approach, we proposed to calculate the photo-consistency gradient for each pixel with consideration of the surrounding related pixels. To distinguish those two methods, we denote the baseline method by  $\partial M_{PDPG}$  and our method by  $\partial M_{TDPG}$ . The following describes the differences between the two methods.

ZNCC is a ubiquitously used photo-consistency measurement in variational mesh refinement, and its formula is as follows:

$$M_{zncc}(x_i) = \frac{1}{n} \left( \frac{\sum_{m=1}^n (a_m - \mu_A)(b_m - \mu_B)}{\sigma_A \sigma_B} \right), M_{zncc}(x_i) \in (-1, 1) \quad (5)$$

where  $x_i$  is the position of a pixel in the reference image,  $M_{zncc}(x_i)$  is the ZNCC measurement at  $x_i$ , while  $A$  and  $B$  are two image patches of equal size from the reference and predicted images at  $x_i$  (see Figure 2).  $a_m$  and  $b_m$  ( $m \in (1, n)$ ) denote the  $m$ -th pixel in  $A$  and  $B$ , respectively, and  $n$  is the number of pixels in each image patch;  $\mu_A/\mu_B$  and  $\sigma_A/\sigma_B$  are the mean and standard deviations of the pixel values in  $A$  and  $B$ , respectively.

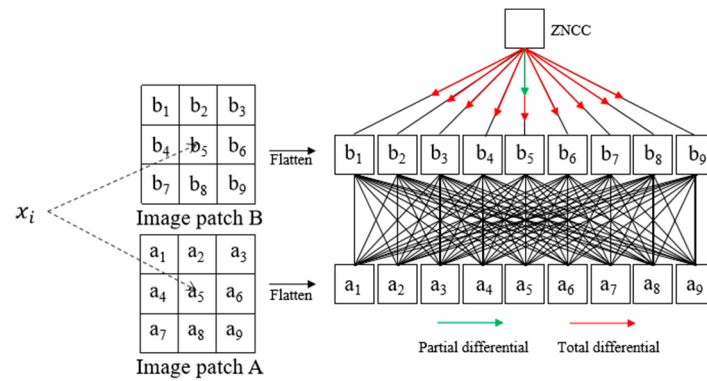


Figure 2. Schematic diagram of the minimization of ZNCC error.

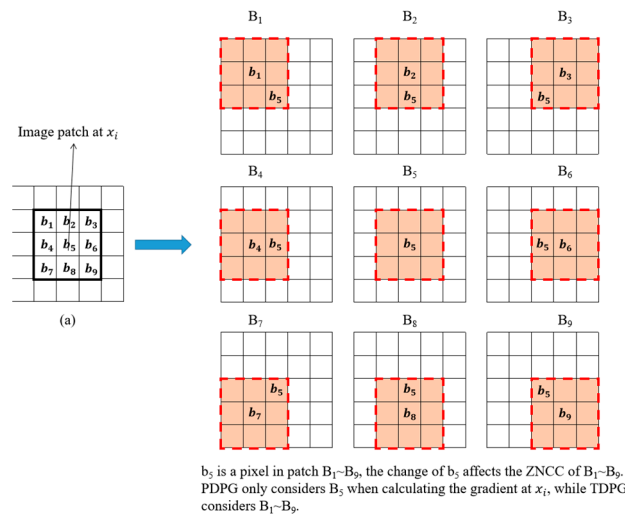
$A$  and  $B$  are two image patches from the reference and predicted images at  $x_i$ . The black lines indicate that the two image patches are combined to calculate the ZNCC value. The arrows represent the differential of ZNCC with respect to the pixels in  $B$ . The right part of Figure 2 shows that ZNCC is jointly calculated by  $a_m$  and  $b_m$ . To minimize the ZNCC error, the partial differential only changes the center pixel, while the total differential changes all the pixels in  $B$ .

The ZNCC error is defined as  $1 - M_{zncc}$ . The variational mesh refinement [13] changes the center pixel value to reduce this error, and the partial derivative is calculated [12,14]:

$$\partial M_{PDPG}(x_i) = \frac{\partial M_{zncc}(x_i)}{\partial b_{center}} = \frac{1}{n} \left( \frac{a_{center} - \mu_A}{\sigma_A \sigma_B} - M_{zncc}(x_i) \frac{b_{center} - \mu_B}{\sigma_B^2} \right) \quad (6)$$

where  $\partial M_{PDPG}(x_i)$  denotes the PDPG at position  $x_i$ ,  $a_{center}/b_{center}$  is center pixels of image patches  $A$  and  $B$ .

On the one hand, PDPG only calculates the partial derivative of the center pixel. As in Figure 2, the ZNCC is determined by all patch pixels, and the total differential should be considered. On the other hand, the PDPG is individually calculated in each image patch. As in Figure 3, pixel  $x_i$  is contained by nine image patches, but the gradient is only determined by the center patch. The gradient calculated in this way is expected to reduce the ZNCC error of  $x_i$ . However, it may increase the ZNCC error of the neighboring pixels, which is not conducive to the convergence of the photo-consistency on the entire image.



**Figure 3.** The difference between PDPG and TDPG.

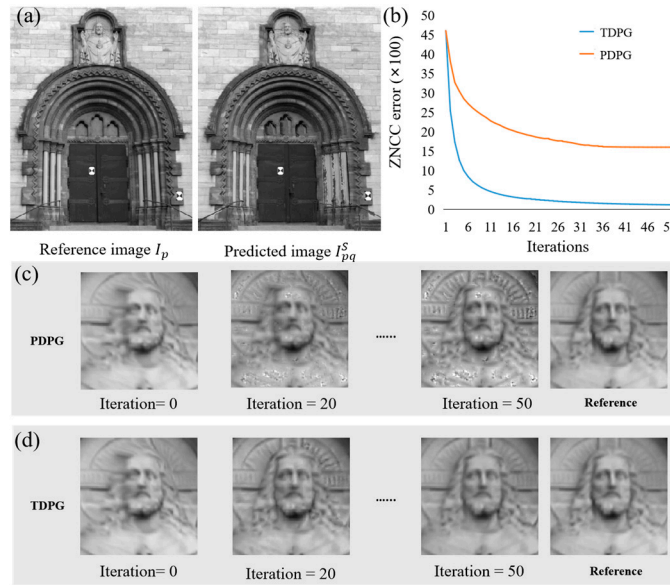
In our method, the photo-consistency gradient of a pixel is jointly determined by all image patches that contain it. Based on this idea, we propose the TDPG calculation method:

$$\partial M_{TDPG}(x_i) = \sum_{m=1}^n W_g(d(x_i^{b_m}, x_i)) \frac{\partial M_{zncc}(x_i^{b_m})}{\partial b_{center}} \quad (7)$$

$$\frac{\partial M_{zncc}(x_i^{b_m})}{\partial b_{center}} = \frac{1}{n} \left( \frac{\alpha_m - \mu_{A_m}}{\sigma_{A_m} \sigma_{B_m}} - M_{zncc}(x_i^{b_m}) \frac{b_m - \mu_{B_m}}{\sigma_{B_m}^2} \right) \quad (8)$$

where  $\partial M_{TDPG}(x_i)$  is the TDPG at  $x_i$  and  $x_i^{b_m}$  is the position of  $b_m$  in the image patch of  $x_i$  (see Figure 3a),  $d(x_i^{b_m}, x_i)$  is the Euclidean distance between  $x_i^{b_m}$  and  $x_i$ .  $W_g(d)$  is the Gaussian weight.  $\frac{\partial M_{zncc}(x_i^{b_m})}{\partial b_{center}}$  is the partial derivative of ZNCC at  $x_i^{b_m}$  with respect to  $b_{center}$ .  $A_m$  and  $B_m$  are two image patches from the reference and predicted images at  $x_i^{b_m}$ .  $\mu_{A_m}/\mu_{B_m}$  and  $\sigma_{A_m}/\sigma_{B_m}$  are the mean and standard deviations of the pixel values in  $A_m$  and  $B_m$ , respectively. The equation shows that the gradient of each pixel is jointly determined by the partial derivative of all pixels within the image patch and that pixels closer to  $x_i$  have a more significant impact.

Figure 4 shows the difference between the convergence process of the PDPG and TDPG on the entire image. We found that TDPG converges after ~25 iterations while PDPG takes ~35 iterations (Figure 4b). TDPG achieves a more effective convergence in that it considers the partial derivative of all pixels in the neighborhood and increases the area affected by the gradient, which thereby facilitates photo-consistency convergence on the entire image. Furthermore, TDPG also yielded a much lower ZNCC error than PDPG. The local magnification area in Figure 4c,d shows that the TDPG method makes the predicted image closer to the reference image in the iterative process.



**Figure 4.** An example of the convergence process of PDPG and TDPG for the entire image. (a) shows the reference (left) and predicted image (right), and (b) shows the photo-consistency convergence curve of the PDPG and TDPG. (c) and (d) show the changes in the predicted image at different iterations by PDPG and TDPG, respectively.

### 2.3. Self-Adaptive Mesh Denoising

Although the photo-consistency gradient improves accuracy and enriches the mesh details, noise and errors in the initial mesh cannot be effectively removed, especially in the texture-less and non-Lambertian regions. The mesh regularization of variational mesh refinement is a combination of the Laplacian and Bi-Laplacian operation [15], an isotropic and one-step mesh denoising method. The one-step property means that the method cannot effectively remove the mesh noise in limited iterations, and the isotropic property makes it hard to retain high-frequency details [11,16]. Therefore, we propose to use the two-step and anisotropic bilateral normal filtering as a regularization term for mesh refinement. However, directly applying the bilateral normal filtering with its strong mesh deformation capability is inappropriate because the small mesh details that are difficult to distinguish from noise will be erased.

We utilized the image ZNCC metric, which indicates the mesh accuracy, to guide mesh denoising. For image pair  $I_p$  and  $I_q$ ,  $x_i(p, q)$  is the position of a pixel in image  $I_p$ . The ray formed by the camera center of  $I_p$  and  $x_i(p, q)$  intersects the mesh at the 3D position  $X_i(p, q)$ . The ZNCC value at face  $f_k$  can be calculated from all the image pairs visible to it (Figure 5):

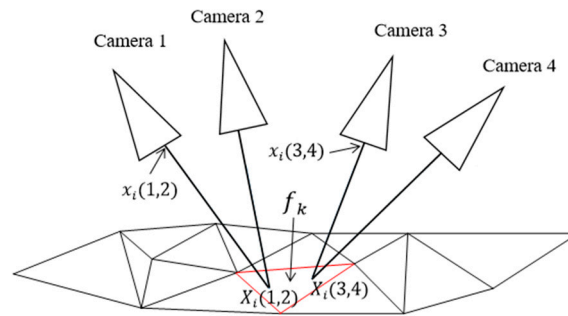
$$C(f_k) = \begin{cases} \frac{\sum_{p,q} \sum_{X_i(p,q) \in f_k} \alpha_{p,q}^{vis}(X_i(p, q)) \cdot M_{zncc}(x_i(p, q))}{\sum_{p,q} \sum_{X_i(p,q) \in f_k} \alpha_{p,q}^{vis}(X_i(p, q))}, & \sum_{p,q} \sum_{X_i(p,q) \in f_k} \alpha_{p,q}^{vis}(X_i(p, q)) \neq 0 \\ 0, & \sum_{p,q} \sum_{X_i(p,q) \in f_k} \alpha_{p,q}^{vis}(X_i(p, q)) = 0 \end{cases} \quad (9)$$

where  $C(f_k)$  represents the ZNCC value of face  $f_k$ .  $\alpha_{p,q}^{vis}(X_i(p, q))$  describes whether the 3D point  $X_i(p, q)$  is simultaneously visible by image  $I_p$  and image  $I_q$ . If it is visible,  $\alpha_{p,q}^{vis}(X_i(p, q)) = 1$ ; otherwise,  $\alpha_{p,q}^{vis}(X_i(p, q)) = 0$ .  $X_i(p, q) \in f_k$  denotes  $X_i(p, q)$  is on face  $f_k$ .  $M_{zncc}(x_i(p, q))$  is the ZNCC value at  $x_i(p, q)$ . Then,  $C(f_k)$  is transferred from mesh face to mesh vertex to form a ZNCC map:

$$C(V_i) = \frac{\sum_{k \in N(V_i)} A(f_k) C(f_k)}{\sum_{k \in N(V_i)} A(f_k)} \quad (10)$$



in which,  $V_i$  is a vertex of the mesh.  $C(V_i)$  represents the ZNCC value of the mesh vertex  $V_i$ .  $N(V_i)$  is the one-ring face neighborhood of  $V_i$ .  $A(f_k)$  is the area of the face  $f_k$ .

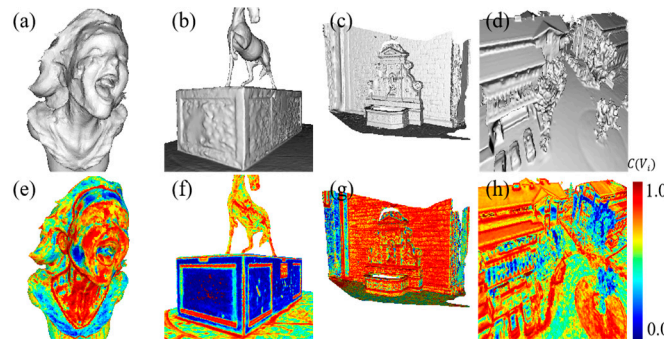


**Figure 5.** The schematic diagram for the calculation of the ZNCC value of face  $f_k$ . The red triangle represents  $f_k$ . All the image pairs visible to  $f_k$  are considered for the ZNCC calculation, indicated as images from Camera 1 to 4 in this figure.

The ZNCC maps of different meshes are shown in Figure 6. We found that the noise area in the mesh has a low value of  $C(V_i)$  due to the error of the mesh shape. In addition, in the area with a high  $C(V_i)$ , the mesh shape conforms to multiview photo-consistency. We use a  $C(V_i)$ -weighted denoising gradient to achieve self-adaptive mesh denoising:

$$g_{regularization}(V_i) = (1 - C(V_i)) \cdot g_{bilateral}(V_i) \tag{11}$$

$g_{regularization}(V_i)$  is the regularization gradient at each vertex, and  $g_{bilateral}(V_i)$  is the bilateral denoising gradient at each vertex which is described in [11].



**Figure 6.** ZNCC maps of different meshes. (a–d) are the initial meshes, and (e–h) are the corresponding ZNCC maps.

It is worth noting that general mesh denoising methods only remove noise based on geometric information. In contrast, this paper adaptively applies a denoising gradient based on the photo-consistency metric, which is conducive to removing significant errors in the initial mesh (see Figure 1).

Finally, our TDR combines the photometric gradient and regularization gradient by  $\beta$ :

$$g_{TDR}(V) = g_{photo}^{TDR}(V) + \beta g_{regularization}(V) \tag{12}$$

$g$  is an abbreviation for gradient.  $g_{photo}^{TDR}(V)$  is the photometric gradient of our TDR method in each iteration.  $g_{photo}^{TDR}(V)$  replaces the  $\partial M(x_i)$  as  $\partial M_{TDPG}(x_i)$  in (Equation (3)).

#### 2.4. Initialization and Implementation Details

We implemented the variational mesh refinement method [13], where ray tracing is used to calculate the projection between images, the image patch size is  $5 \times 5$ ,  $\beta$  is set to 0.2. Those parameters are the same as that used in [13]. The Gaussian weights  $W_g(d)$  are normalized according to the image patch size. The mesh denoising scheme is local for the bilateral normal filtering algorithm, and the normal iterations and the vertex iterations are set to 20 and 10, respectively. To solve the nonconvex problem, we utilize the L-BFGS optimization algorithm [17,18]. The parameter setting of the TDR method is the same as the variational mesh refinement. Our algorithm is implemented with C++. All experiments were conducted on a single PC machine with Intel(R) Core(TM) i7-8700 CPU (12-core), 64 GB RAM, and Nvidia GTX 2070 GPU.

We used a variety of mainstream meshes as the initial inputs, including the OpenMVS [19] mesh, which is reconstructed by the built-in surface reconstruction function, the COLMAP [20] mesh, and the CMP-CMVS mesh [21]. We used the OpenMVS mesh as the initial mesh by default, as the mesh faces are uniformly sized, and the scene reconstruction is complete.

### 3. Experiments

#### 3.1. Dataset and Evaluation Metrics

**Dataset:** We use six datasets that cover different scenes, including UAV (unmanned air vehicle) scenes, close-range scenes, and simulation scenes. Information about these datasets is given in Table 1.

**Table 1.** Introduction to the datasets used in this study.

Dataset	Name	Image Size	Number of images	Initial Mesh	Image Acquisition
Tanks And Temples	Family	$1920 \times 1080$	153	OpenMVS	Handheld
	Francis	$1920 \times 1080$	302	OpenMVS	Handheld
	Horse	$1920 \times 1080$	151	OpenMVS	Handheld
	Panther	$1920 \times 1080$	314	OpenMVS	Handheld
ETH3D	delivery area	$6048 \times 4032$	44	OpenMVS	Handheld
	facade	$6048 \times 4032$	76	OpenMVS	Handheld
	relief	$6048 \times 4032$	31	OpenMVS	Handheld
	relief 2	$6048 \times 4032$	31	OpenMVS	Handheld
BlendedMVS	UAV_Scene1	$2048 \times 1536$	77	OpenMVS	Rendered
	UAV_Scene2	$2048 \times 1536$	125	OpenMVS	Rendered
	UAV_Scene3	$2048 \times 1536$	75	OpenMVS	Rendered
EPFL	Herz-Jesu-P8	$3072 \times 2048$	8	OpenMVS /COLMAP/ CMPMVS	Handheld
	Fountain-P11	$3072 \times 2048$	11	OpenMVS/ COLMAP/ CMPMVS	Handheld
CG Simulation Dataset	Joyful	$1920 \times 1080$	70	OpenMVS	Rendered
Personal Collection Dataset	House	$4592 \times 3056$	36	OpenMVS	UAV
	Woodcarving	$2016 \times 4032$	146	OpenMVS	Handheld



- (1) Tanks And Temples [22] is a benchmark for image-based 3D reconstruction. The image sequences come from video streams. We picked the Family, Francis, Horse, and Panther data for close-range scene evaluation.
- (2) ETH3D [23] is a benchmark for multiview stereo (MVS) evaluation. It provides ultra-high-resolution images registered to the 3D laser scan point clouds. We picked its facade, delivery\_area, relief, and relief\_2 data for close-range scene evaluation.
- (3) BlendedMVS [24] is a large-scale simulation MVS dataset. It provides ground-truth meshes and rendered images. We selected four outdoor scenes captured by UAVs, namely, UAV\_Scene1, UAV\_Scene2, and UAV\_Scene3, for UAV scene evaluation.
- (4) Custom simulation dataset. We picked computer graph (CG) mesh models Joyful (Kim et al., 2016) as ground-truth mesh and utilized the same lighting to render 70 images from fixed perspectives using Blender [25].
- (5) The EPFL dataset [26] provides two ground-truth meshes captured by LIDAR sensors, namely, Herz-Jesu-P8 and Fountain-P11, and provides the images registered with the meshes.
- (6) Personal Collection Dataset. We collected multiview images from the internet and natural scenes for qualitative evaluation.

Evaluation Metrics: Similar to [22,23], we use the shortest distance from a point to the surface to evaluate the precision of a mesh.  $I$  is the input mesh to be evaluated, and  $R$  is the reference mesh. For a vertex  $i \in I$ , its distance to the reference mesh is defined as  $d_{i \rightarrow R}$ . These distances can be aggregated to define the accuracy of the input mesh  $I$  for any distance threshold  $d$ :

$$P(d) = 100 * \frac{\sum_{i \in I} [d_{i \rightarrow R} < d]}{|I|} \quad (13)$$

where  $[\cdot]$  is the Iverson bracket.  $|I|$  is the number of vertices for mesh  $I$ .

Similarly, for a reference mesh vertex  $r \in R$ , its distance to the input mesh is defined as  $d_{r \rightarrow I}$ . The completeness of the input mesh for any distance threshold  $d$  is defined as:

$$C(d) = 100 * \frac{\sum_{r \in R} [d_{r \rightarrow I} < d]}{|R|} \quad (14)$$

Accuracy and completeness can be combined to calculate the F-score:

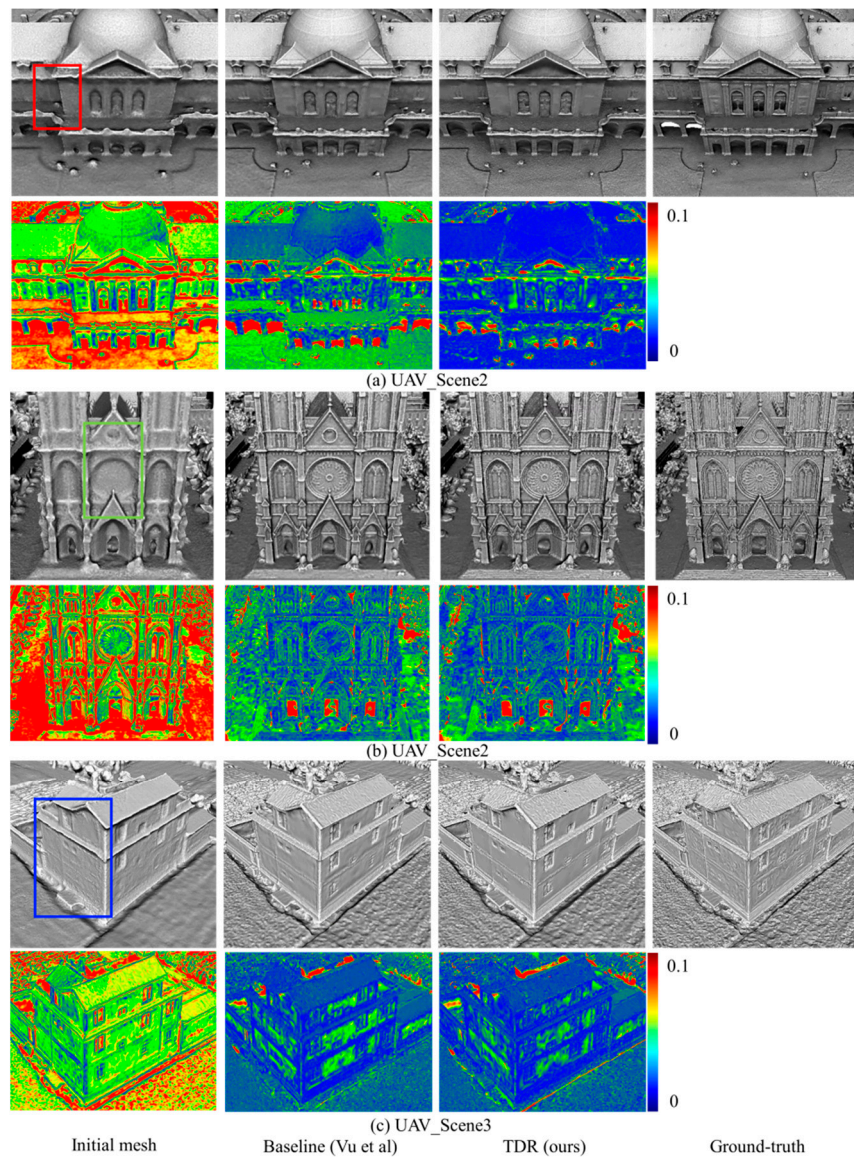
$$F(d) = \frac{2 * P(d) * C(d)}{P(d) + C(d)} \quad (15)$$

The F-score is the harmonic mean of the accuracy and completeness at threshold  $d$ . Threshold  $d$  varies according to the different scales of the dataset. Moreover, we also use the mean of  $d_{i \rightarrow R}$  of all vertices as the mean-accuracy metric and the mean of  $d_{r \rightarrow I}$  of all vertices as the mean-completeness metric.

### 3.2. Comparison with the Baseline Method

#### 3.2.1. Performance on the UAV Dataset

In urban scenes, the 3D reconstruction of structures (such as planes and edges) is the critical point. BlendedMVS provides urban meshes that are difficult to capture by a laser scanner fully. We selected the UAV\_Scene1-UAV\_Scene3 from BlendedMVS for experiment. Figure 7 shows the visual results. Compared with the baseline method [13], the mesh refined by our TDR approach is sharper in the edge region (see the red boxes in Figure 7a) because we utilize the bilateral normal filtering with an edge-preserving effect. In addition, our method obtains a better mesh detail than the baseline method (see the green box in Figure 7b) because our TDPG converges better on photo-consistency. Furthermore, in a limited number of iterations, the Laplace operator used in the baseline method does not remove the undulation in the initial mesh (see the blue box in Figure 7c). In contrast, our method makes the surface flatter due to the bilateral normal filtering with a better denoising ability.



**Figure 7.** The meshes result (odd rows) and the error ( $d_{i \rightarrow R}$ ) distributions of the meshes (even rows) on the BlendedMVS dataset.

Table 2 shows the quantitative results of the BlendedMVS dataset. Since the captured images are far from the objects, we set the cutoff distance  $d$  as 0.05 m. Table 2 shows that the baseline method and our TDR method both improve the precision of the initial mesh, but the improvement brought by our method is much more evident than that of the baseline method. The error ( $d_{i \rightarrow R}$ ) distribution of Figure 7 shows that compared with the baseline method, the accuracy improvement brought by our TDR method is reflected in the flat area. This shows that the proposed TDR method has a stronger ability to regularize the mesh than the baseline method.

**Table 2.** Quantitative evaluation of the BlenderMVS dataset. Acc. means accuracy and Compl. represents completeness.

		Initial Mesh	Baseline	TDR
UAV_ Scene1	Acc. [%]	28.11	77.00	81.67
	Compl. [%]	20.30	61.37	63.16
	F1 [%]	23.57	68.30	71.23
	Mean-Acc. [X10-2]	10.89	6.71	5.09

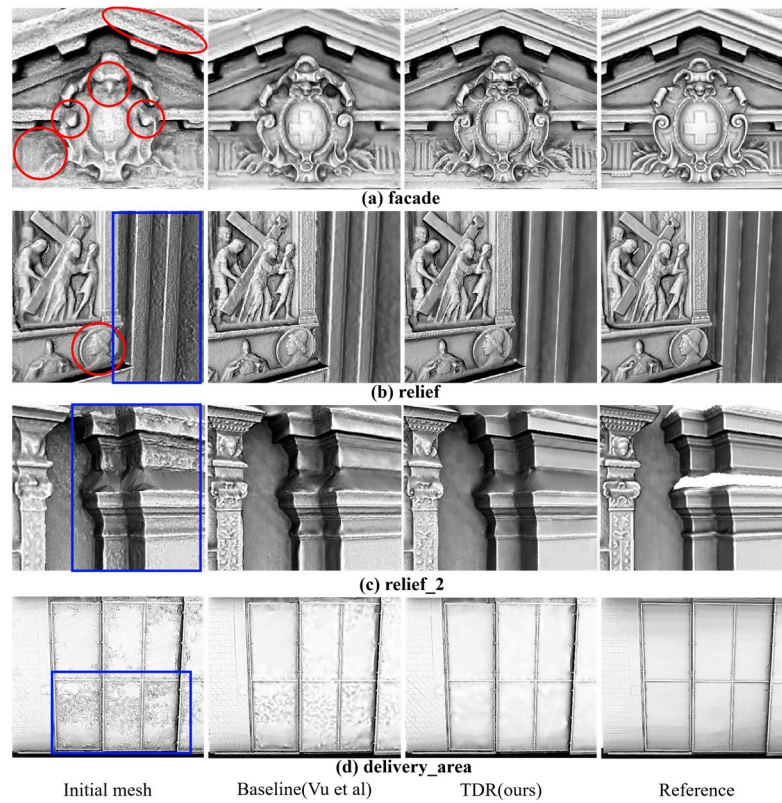
	Mean-Compl. [X10-2]	24.13	15.03	14.13
UAV_ Scene2	Acc. [%]	31.90	72.79	77.71
	Compl. [%]	29.80	90.20	90.77
	F1 [%]	30.81	80.57	83.73
	Mean-Acc. [X10-2]	12.80	8.00	6.70
	Mean-Compl. [X10-2]	42.40	28.30	27.30
UAV_ Scene3	Acc. [%]	29.53	83.98	85.79
	Compl. [%]	28.90	66.85	67.95
	F1 [%]	29.21	74.44	75.84
	Mean-Acc. [X10-2]	10.27	5.37	4.94
	Mean-Compl. [X10-2]	26.80	17.55	17.44

In summary, the TDR method can reconstruct sharp features and flat planes even under poor initial conditions, making our method very suitable for reconstructing urban scenes.

### 3.2.2. Performance on the Close-Range Dataset

For the close-range dataset, we test our method on the recently published close-range MVS datasets, i.e., ETH3D and Tanks and Temples.

ETH3D dataset: This dataset has ultrahigh-resolution images and provides the laser scan point cloud with the registered images. We use the Poisson surface reconstruction [27] to obtain the reference mesh. Figure 8 shows the results of our experiment. Comparing the baseline method, our TDR method which utilized TDPG has better mesh details (see the red ellipses in Figure 8). Then, Figure 8c,d show that the initial mesh has substantial noise in texture-less regions. The baseline method using the isotropic denoising method cannot effectively remove this noise and retain sharp edges, while our TDR methods can effectively achieve these effects (see the blue rectangles in Figure 8). The results of the accuracy evaluation are shown in Table 3. Since the captured images are close to objects, we set  $d$  as 0.005 m. Table 3 shows that the proposed TDR algorithm achieves the best results in terms of almost all quantitative metrics.



**Figure 8.** Visual comparison of results on the ETH3D dataset.

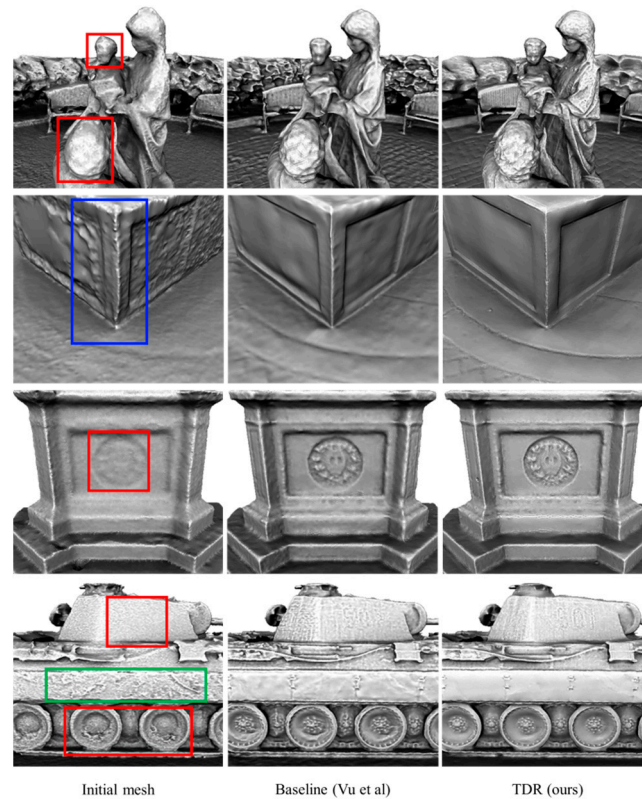
**Table 3.** Quantitative evaluation of results on the ETD3D dataset.

		Initial Mesh	Baseline	TDR
delivery_area	Acc. [%]	53.18	53.33	56.98
	Compl. [%]	37.65	41.85	42.95
	F1 [%]	44.09	46.89	48.98
	Mean-Acc. [X10-3]	7.89	7.80	7.24
	Mean-Compl. [X10-3]	51.24	50.90	50.84
facade	Acc. [%]	24.25	34.35	42.78
	Compl. [%]	27.07	38.22	45.15
	F1 [%]	25.58	36.18	43.93
	Mean-Acc. [X10-3]	34.89	30.67	26.76
	Mean-Compl. [X10-3]	15.35	13.91	13.22
relief	Acc. [%]	95.45	95.87	95.97
	Compl. [%]	94.09	95.63	96.79
	F1 [%]	94.77	95.75	96.38
	Mean-Acc. [X10-3]	1.79	16.25	13.51
	Mean-Compl. [X10-3]	2.15	1.76	13.17
relief_2	Acc. [%]	90.48	90.34	92.58
	Compl. [%]	86.72	89.74	90.72
	F1 [%]	88.56	90.04	91.64
	Mean-Acc. [X10-3]	1.94	2.16	1.93
	Mean-Compl. [X10-3]	2.66	2.19	2.13

Tanks and Temples dataset: This dataset does not provide image poses that are registered with the reference point cloud. Therefore, the results of this experiment are qualitatively evaluated. Compared with the baseline algorithm, results of the proposed TDR method have finer details (the red boxes in Figure 9), flatter planes (the green boxes in



Figure 9), and sharper edges (the blue boxes in Figure 9). This proves the effectiveness of the two improvements in this paper.

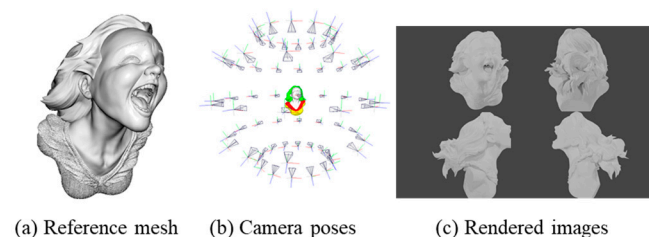


**Figure 9.** Visual comparison of results on the Tanks and Temples dataset.

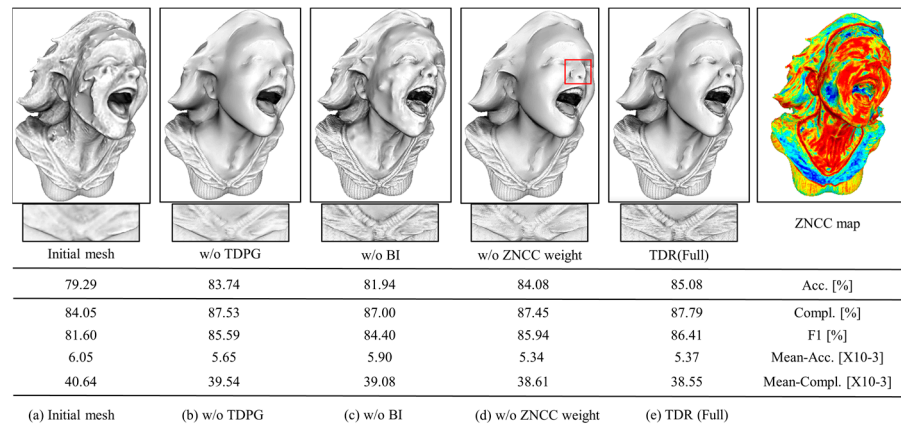
### 3.3. Discussion

#### 3.3.1. Ablation Experiment

We evaluated the effectiveness of the two improvements in TDR in the simulated CG dataset (see Figure 10), and ablation experiments were conducted. We tested four different configurations: (1) w/o TDPG: using PDPG and self-adaptive bilateral regularization. (2) w/o BI: using TDPG and Laplace regularization. (3) w/o ZNCC weighted: using TDPG and bilateral regularization without ZNCC weighted. (4) Full TDR: using TDPG and self-adaptive bilateral regularization. Figure 11 shows the results. There are giant pits in the face of the initial mesh due to a sizeable texture-less area in the rendered images (Figure 11). The meshes using bilateral regularization ((b), (c), (d)) do not show this error. Comparing (b) with (e) in Figure 11, we found that TDPG presents a more detailed and accurate result than PDPG. Comparing (d) with (e), the ZNCC weighted strategy succeeded in preserving mesh details (see the red box in Figure 11d).



**Figure 10.** The simulated CG dataset. The three columns are the mesh, cameras, and rendered images from left to right.

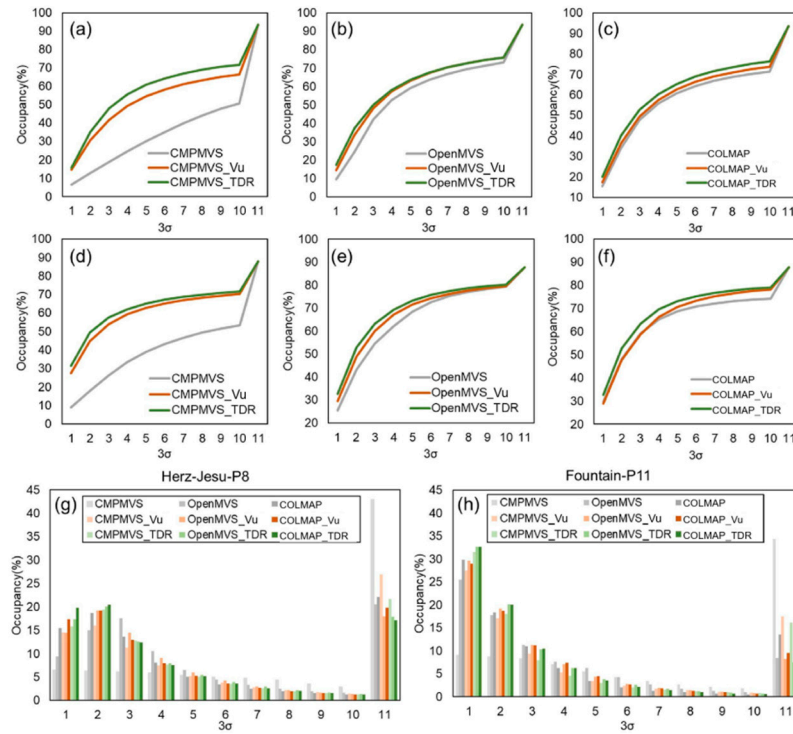


**Figure 11.** The visualization result and the accuracy metric of the Joyful data.

### 3.3.2. The Influence of Initial Meshes

To discuss the impact of initial meshes on the TDR algorithm, the widely used CPMVS mesh, OpenMVS mesh, and COLMAP mesh models were chosen as initial meshes for comparisons. We also added the baseline method [13] for comparison. The baseline algorithm generated the CPMVS\_Vu mesh, OpenMVS\_Vu mesh, and COLMAP\_Vu mesh. The TDR method generated the CPMVS\_TDR mesh, OpenMVS\_TDR mesh, and COLMAP\_TDR mesh.

In this section, we use the EPFL dataset [26] for evaluation. This benchmark is designed for mesh evaluation. Therefore, we test our results using the evaluation metric proposed in this benchmark. First, the reference and the input mesh are projected into the same image, and the residual of the depth of each pixel is calculated. Then, the occupancy rates of the residuals from  $3\sigma$  ( $\sigma = 1.1$  mm) to 10-times  $3\sigma$  are counted. Finally, the occupancy rates of the residuals of all images are averaged to obtain the final occupancy distribution histogram and occupancy density map, as shown in Figure 12. In addition, the weighted average of the residual distribution histogram can be drawn to obtain the accuracy metric. The proportions of the part less than  $30\sigma$  are counted as the completeness metric [26], shown in Table 4.



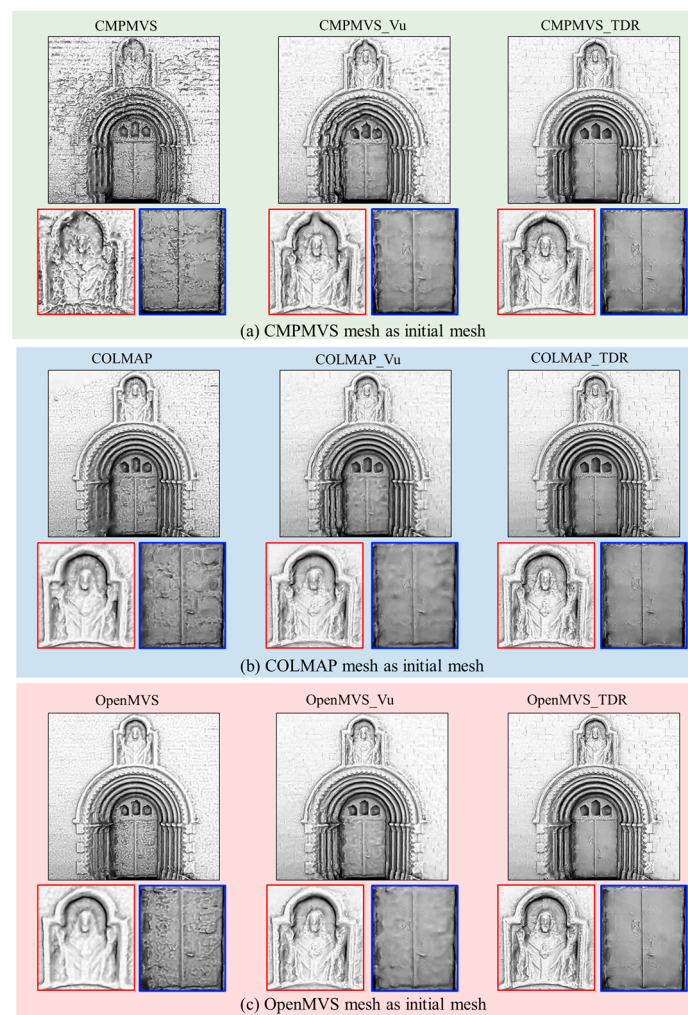
**Figure 12.** The residual occupancy density maps (a–f) and occupancy distribution histograms (g,h) of all meshes.

**Table 4.** The accuracy and completeness of all meshes.

	Herz-Jesu-P8			Fountain-P11		
	#faces [M]	Acc. [3σ]	Compl. [%]	#faces [M]	Acc. [3σ]	Compl. [%]
CMPMVS	2.76	6.25	50.57	2.47	5.08	53.42
CMPMVS_Vu	1.25	4.30	66.55	1.55	2.90	70.34
CMPMVS_TDR	1.25	3.77	71.83	1.55	2.64	71.70
OpenMVS	1.54	4.04	72.96	1.89	2.42	79.37
OpenMVS_Vu	1.26	3.59	75.62	1.52	2.15	79.52
OpenMVS_TDR	1.26	3.49	75.72	1.53	1.95	80.29
COLMAP	1.14	3.80	71.45	1.51	2.42	74.33
COLMAP_Vu	1.23	3.60	73.77	1.39	2.24	78.25
COLMAP_TDR	1.22	3.32	76.41	1.40	1.99	79.12

Table 4 and Figure 13 show that, regardless of the initial mesh, the precision improvement brought by the TDR algorithm is much more significant than that of the baseline algorithm. For both TDR and baseline methods, the accuracy and completeness of the refined meshes are high if the initial meshes are accurate, and vice versa. The reason is that mesh refinement is a nonconvex problem and has a certain dependence on the initial value when using the gradient descent method to solve it.





**Figure 13.** Visualization of the results of all methods on Herz-Jesu-P8.

Figure 13 shows the visual results for Herz-Jesu-P8. Even when handling a very poor initial mesh, TDR still reconstructs a good result (Figure 13a). All initial meshes have considerable noise at the door region (the blue boxes in Figure 13). The TDR method outperforms the baseline method in mesh denoising ability. What is more commendable is that the proposed algorithm with a strong regularization term well retains the details in the human sculpture (the red boxes in Figure 13). This is attributed to the self-adaptive weighted denoising strategy, which effectively combines the photo-consistency gradient and the mesh denoising gradient, with the denoising gradient mainly applied to the noise areas.

In other words, due to the stronger denoising ability, the TDR method can handle a worse initial mesh model.

### 3.3.3. Running Times Evaluation

In this section, we discuss the running time of our 3D reconstruction system and TDR algorithm on datasets EPFL and Tanks and Temples.

Our 3D reconstruction system comprises SFM, MVS, mesh reconstruction and mesh refinement steps. The SFM step uses COLMAP (CUDA version) with default parameters. The MVS and mesh reconstruction step uses OpenMVS with default parameters. The proposed TDR algorithm is used in the mesh refinement stage with no special performance optimization. Table 5 shows the processing times of the 3D reconstruction system. The SFM, MVS, mesh reconstruction, and mesh refinement time ratios are about 16%, 25%,

4%, and 55%, respectively. Although the mesh refinement step evolves the initial mesh to a high quality with fine details, it cost the most time in the 3D reconstruction system.

**Table 5.** The processing times of the 3D reconstruction system.

	Herz-Jesu-P8		Fountain-P11		Family		Francis		Horse		Panther	
	Time (s)	Ratio (%)	Time (s)	Ratio (%)	Time (s)	Ratio (%)	Time (s)	Ratio (%)	Time (s)	Ratio (%)	Time (s)	Ratio (%)
SFM	-	-	-	-	472	12	925	19	276	12	1388	22
MVS	120	35	196	38	1009	25	1313	27	575	24	1545	24
Mesh reconstruction	60	18	86	17	222	6	131	3	91	4	331	5
Mesh refinement	159	47	233	45	2335	58	2455	51	1427	60	3149	49

Table 6 shows the time consumption of critical steps in the TDR and the baseline method. The two improvements of our method correspond to the computation of  $\partial M$  and  $g_{regularization}$ , respectively. On the one hand, the PDPG method can use the integral image [28] to accelerate calculations, while the TDPG method cannot. On the other hand, PDPG calculates the partial differential once per pixel, while TDPG calculates the 25 times partial differential for the  $5 \times 5$  image patch. Therefore, the running time of  $\partial M$  is 8–13 times that of PDPG in the experimental data. For the regularization item  $g_{regularization}$ , the baseline method takes less than 1 s on all experimental data. Our TDR method utilizes the bilateral normal filter method, thus increasing the time consumption, but the time ratio does not exceed 2% of the total time. Overall, the running time of our method is 1.5–2 times more than the baseline method due to a time increase for TDPG.

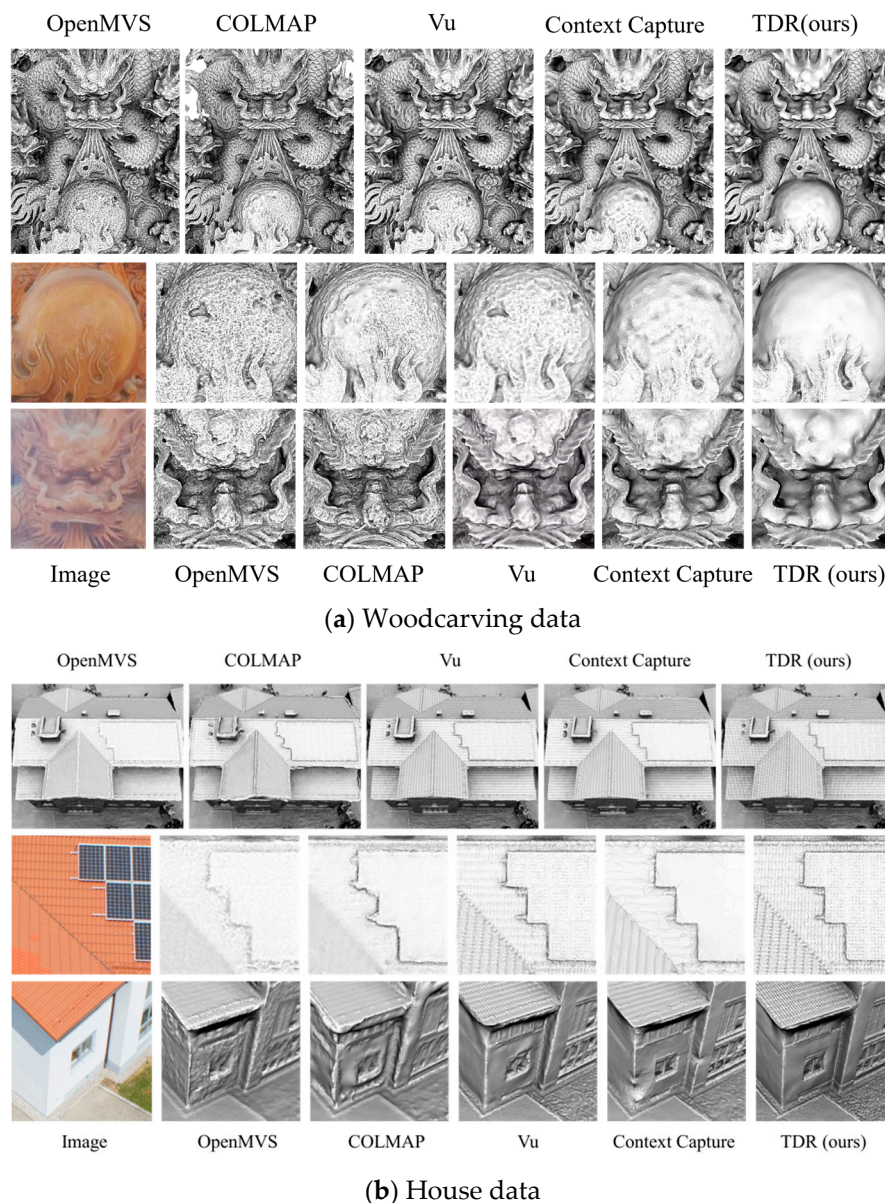
**Table 6.** The time consumption of critical steps in the TDR and the baseline method.

	Herz-Jesu-P8		Fountain-P11		Family		Francis		Horse		Panther	
	Vu	TDR	Vu	TDR	Vu	TDR	Vu	TDR	Vu	TDR	Vu	TDR
#Vertices (K)	754	754	944	944	895	895	672	672	527	527	1897	1897
#Images pixels (M)	50	50	69	69	317	317	626	626	313	313	651	651
Ray tracing (s)	30	32	43	45	561	389	624	524	416	291	850	810
Compute $\partial M$ (s)	7	85	10	129	231	1391	285	1271	184	759	198	1626
Compute $g_{photo}$ (s)	17	16	26	23	371	257	277	245	211	151	339	316
Compute $g_{regularization}$ (s)	0	3	0	4	0	7	0	5	0	5	0	10
Others (s)	22	24	30	32	435	290	504	410	331	221	433	387
Total (s)	76	159	109	233	1597	2335	1690	2455	1142	1427	1819	3149

### 3.4. Comparison with Open Source and Commercial Software

In this section, we compare our method with representative open source software (OpenMVS and COLMAP), commercial software (Context Capture [29]), and the baseline method [13]. Figure 14 visually compares the reconstructed 3D models on the Personal Collection Dataset. OpenMVS can reconstruct the overall shapes, although some fine details are lost (the roof in the House data), and noise is covered on the smooth surface (all enlarged parts in Figure 14). The COLMAP mesh is similar to the OpenMVS mesh, with slightly more details (the roof in the House data) but fails in texture-less regions (the white wall in the House data). The Context Capture mesh also fails in texture-less regions (the white wall in the House data), and it is too smooth, which causes details loss (roof in the

House data). The method of [13] enhances the details of the initial mesh to a certain extent (the roof in the House data), but the mesh noise is still not eliminated, and the edges are not sharpened (the white wall in the House data). In contrast, due to the strong denoising ability brought by the bilateral normal filter, our mesh result on the Woodcarving data is the smoothest, and the house edges are the sharpest. At the same time, the TDPG recovers the most details.



**Figure 14.** Visual comparison on the Personal Collection Dataset.

#### 4. Conclusions

This study proposed a new mesh refinement approach coupling total differential photometric mesh refinement and self-adapted mesh denoising. On the one hand, traditional PDPG in variational mesh refinement is replaced by TDPG. TDPG considers the neighboring pixels and increases the area affected by the gradient, which results in more effective convergence of the photo-consistency, thus increasing the details and accuracy of the mesh. On the other hand, the self-adaptive denoising strategy provides a framework for image-guided mesh denoising. The intensity of the denoising gradient can be

adaptively adjusted according to the multiview ZNCC metric, which facilitates the removal of significant errors in the initial mesh and preserving mesh details. Experiments on different scenes and comparisons with open-source and commercial software were conducted. The refined meshes are evaluated in terms of both accuracy and completeness. Results showed that our method outperformed current variational mesh refinement methods and is comparable and even better than commercial software, and the mesh refined by our method is the most detailed, accurate, and regular. In the future, we plan to run our method on GPU and explore the fusion of subpixel sampling and photometric stereo technology with mesh refinement.

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

**Funding:** This work was supported in part by the Ningbo Key Research and Development Project (No. 20201ZDYF020236) and Hubei Key Research and Development Project (NO. 2022BAA035).

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** The Tanks And Temples Dataset can be obtained from <https://www.tanksandtemples.org/> (accessed on 20 December 2022). The ETH3D Dataset can be obtained from <https://www.eth3d.net/> (accessed on 20 December 2022). The EPFL Dataset can be obtained from <https://icwww.epfl.ch/multiview/denseMVS.html> (accessed on 20 December 2022). The BlendedMVS dataset can be obtained from <https://github.com/YoYo000/BlendedMVS> (accessed on 20 December 2022). The CG Simulation and Personal Collection datasets are available from the corresponding author upon reasonable request.

**Acknowledgments:** The authors are grateful to the providers of the Temples and Tanks dataset, the Strecha dataset, and the three computer graph mesh models (Joyful, Armadillo, and Happy\_vrrip). We would also like to thank researchers who published open-source code or programs used to generate the CPMVS meshes, OpenMVS meshes, and COLMAP meshes in our experiments.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Ju, Y.; Shi, B.; Jian, M.; Qi, L.; Dong, J.; Lam, K.-M. NormAttention-PSN: A High-frequency Region Enhanced Photometric Stereo Network with Normalized Attention. *Int. J. Comput. Vis.* **2022**, *130*, 3014–3034.
2. Yang, J.; Ding, B.; He, Z.; Pan, G.; Cao, Y.; Cao, Y.; Zheng, Q. ReDDLE-Net: Reflectance Decomposition for Directional Light Estimation. *Photonics* **2022**, *9*, 656.
3. Ju, Y.; Jian, M.; Guo, S.; Wang, Y.; Zhou, H.; Dong, J. Incorporating lambertian priors into surface normals measurement. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–13. <https://doi.org/10.1109/tim.2021.3096282>.
4. Ju, Y.; Peng, Y.; Jian, M.; Gao, F.; Dong, J. Learning conditional photometric stereo with high-resolution features. *Comput. Vis. Media* **2022**, *8*, 105–118.
5. Liu, Y.; Ju, Y.; Jian, M.; Gao, F.; Rao, Y.; Hu, Y.; Dong, J. A deep-shallow and global-local multi-feature fusion network for photometric stereo. *Image Vis. Comput.* **2022**, *118*, 104368.
6. Romanoni, A.; Matteucci, M. Facetwise Mesh Refinement for Multi-View Stereo. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, Milan, Italy, 10–15 January 2021; pp. 6794–6801.
7. Li, Z.; Wang, K.; Zuo, W.; Meng, D.; Zhang, L. Detail-Preserving and Content-Aware Variational Multi-View Stereo Reconstruction. *Ieee Trans. Image Process.* **2016**, *25*, 864–877. <https://doi.org/10.1109/tip.2015.2507400>.
8. Romanoni, A.; Matteucci, M. Mesh-based camera pairs selection and occlusion-aware masking for mesh refinement. *Pattern Recognit. Lett.* **2019**, *125*, 364–372. <https://doi.org/10.1016/j.patrec.2019.05.006>.
9. 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 2017 IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3839–3847.
10. 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.
11. Zheng, Y.; Fu, H.; Au, O.K.; Tai, C.L. Bilateral normal filtering for mesh denoising. *IEEE Trans. Vis. Comput. Graph* **2011**, *17*, 1521–1530. <https://doi.org/10.1109/TVCG.2010.264>.
12. Pons, J.-P.; Keriven, R.; Faugeras, O. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *Int. J. Comput. Vis.* **2007**, *72*, 179–193. <https://doi.org/10.1007/s11263-006-8671-5>.

13. Vu, H.H.; Labatut, P.; Pons, J.P.; Keriven, R. High accuracy and visibility-consistent dense multiview stereo. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 889–901. <https://doi.org/10.1109/TPAMI.2011.172>.
14. S. Li, S. Y. Siu, T. Fang, L. Quan, Efficient multi-view surface refinement with adaptive resolution control, in: European Conference on Computer Vision, Springer, Amsterdam, The Netherlands, 11–14 October 2016, pp. 349–364.
15. Kobbelt, L.; Campagna, S.; Vorsatz, J.; Seidel, H.-P. Interactive multi-resolution modeling on arbitrary meshes. In Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, Orlando, FL, USA, 19–24 July 1998; pp. 105–114.
16. Zhang, H.; Wu, C.; Zhang, J.; Deng, J. Variational mesh denoising using total variation and piecewise constant function space. *IEEE Trans. Vis. Comput. Graph.* **2015**, *21*, 873–886.
17. Byrd, R.H.; Lu, P.H.; Nocedal, J.; Zhu, C.Y. A Limited Memory Algorithm for Bound Constrained Optimization. *Siam J. Sci. Comput.* **1995**, *16*, 1190–1208. <https://doi.org/10.1137/0916069>.
18. Zhu, C.Y.; Byrd, R.H.; Lu, P.H.; Nocedal, J. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *Acm Trans. Math. Softw.* **1997**, *23*, 550–560. <https://doi.org/10.1145/279232.279236>.
19. Cernea, D. OpenMVS: Open Multiple View Stereovision. Available online: <https://github.com/cdcseacave/openMVS/> (accessed on 20 December 2022).
20. Schonberger, J.L.; Frahm, J.-M. Structure-from-motion revisited. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4104–4113.
21. Jancosek, M.; Pajdla, T. Multi-View Reconstruction Preserving Weakly-Supported Surfaces. In Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011.
22. Knapitsch, A.; Park, J.; Zhou, Q.-Y.; Koltun, V. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Trans. Graph. ToG* **2017**, *36*, 1–13.
23. Schps, T.; Schnberger, J.L.; Galliani, S.; Sattler, T.; Schindler, K.; Pollefeys, M.; Geiger, A. A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
24. 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, WA, USA, 13–19 June 2020; pp. 1790–1799.
25. Blender. Version v2.93.4 (Software). Available online: <https://www.blender.org/> (accessed on 20 December 2022).
26. Strecha, C.; Von Hansen, W.; Van Gool, L.; Fua, P.; Thoennessen, U. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 24–26 June 2008; pp. 1–8.
27. Kazhdan, M.; Chuang, M.; Rusinkiewicz, S.; Hoppe, H. Poisson surface reconstruction with envelope constraints. In *Computer Graphics Forum*; Wiley Online Library, Hoboken, New Jersey, USA, 2020; Volume 39, pp. 173–182.
28. Facciolo, G.; Limare, N.; Meinhardt-Llopis, E. Integral images for block matching. *Image Processing On Line* **2014**, *4*, 344–369.
29. ContextCapture, Version v4.4.9.516 (software). 2020. Available online: <https://www.bentley.com/en/products/brands/contextcapture> (accessed on 20 December 2022).

**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.