remote sensing

MDPI

Article

3D Reconstruction of Remote Sensing Mountain Areas with TSDF-Based Neural Networks

Zipeng Qi [1,2,3], Zhengxia Zou [4,*], Hao Chen [1,2,3] and Zhenwei Shi [1,2,3]

[1] Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, China
[2] Beijing Key Laboratory of Digital Media, Beihang University, Beijing 100191, China
[3] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China
[4] Department of Guidance, Navigation and Control, School of Astronautics, Beihang University, Beijing 100191, China
* Correspondence: zhengxiazou@buaa.edu.cn

Abstract: The remote sensing 3D reconstruction of mountain areas has a wide range of applications in surveying, visualization, and game modeling. Different from indoor objects, outdoor mountain reconstruction faces additional challenges, including illumination changes, diversity of textures, and highly irregular surface geometry. Traditional neural network-based methods that lack discriminative features struggle to handle the above challenges, and thus tend to generate incomplete and inaccurate reconstructions. Truncated signed distance function (TSDF) is a commonly used parameterized representation of 3D structures, which is naturally convenient for neural network computation and computer storage. In this paper, we propose a novel deep learning method with TSDF-based representations for robust 3D reconstruction from images containing mountain terrains. The proposed method takes in a set of images captured around an outdoor mountain and produces high-quality TSDF representations of the mountain areas. To address the aforementioned challenges, such as lighting variations and texture diversity, we propose a view fusion strategy based on reweighted mechanisms (VRM) to better integrate multi-view 2D features of the same voxel. A feature enhancement (FE) module is designed for providing better discriminative geometry prior in the feature decoding process. We also propose a spatial–temporal aggregation (STA) module to reduce the ambiguity between temporal features and improve the accuracy of the reconstruction surfaces. A synthetic dataset for reconstructing images containing mountain terrains is built. Our method outperforms the previous state-of-the-art TSDF-based and depth-based reconstruction methods in terms of both 2D and 3D metrics. Furthermore, we collect real-world multi-view terrain images from Google Map. Qualitative results demonstrate the good generalization ability of the proposed method.

Keywords: TSDF-based 3D reconstruction; spatial–temporal information fusion; multi-head cross attention; optical remote sensing; aerial image

check for updates

Citation: Qi, Z.; Zou, Z.; Chen, H.; Shi, Z. 3D Reconstruction of Remote Sensing Mountain Areas with TSDF-Based Neural Networks. *Remote Sens.* **2022**, *14*, 4333. https://doi.org/10.3390/rs14174333

Academic Editor: Tamas Sziranyi

Received: 14 June 2022
Accepted: 26 August 2022
Published: 1 September 2022

1. Introduction

Reconstruction of 3D scenes has a wide range of applications in remote sensing [1–7], surveying [8], and game modeling. In computer vision and computer graphics, given a single image or a set of images, 3D scene reconstruction is the process of capturing the shape and appearance of real scenes. Outdoor mountains are an important ground feature in remote sensing applications. In the remote sensing field, most recent works of 3D scene construction concentrate on using airborne radar images. In this paper, we investigate the problem of reconstructing 3D mountains using optical images from remote sensing scenes. The low cost and easy availability of the reconstruction through optical images makes it easy to acquire and implement from both airborne and spaceborne platforms.

The representation of 3D scenes is a prerequisite for 3D reconstruction. There are mainly three 3D data representation approaches widely used for 3D reconstruction: point cloud, mesh, and truncated signed distance function (TSDF). Among these approaches, point cloud representation is an irregular data structure for 3D structure, storing sparse point-based data of objects. The limitation of the point cloud representation is that there is no connection between points and the lack of surface information of scenes or objects. Mesh representation divides the surface of an object into many triangular patches with a tight graph structure. TSDF represents the 3D structure of an object by evenly dividing the space into several voxels and storing the distance from each voxel to the surface. TSDF is represented as a regular grid structure, which can be naturally processed by 3D convolution. The above representations can be converted to each other easily. For example, point cloud and TSDF can be converted into mesh using the Poisson reconstruction algorithm [9] and marching cubes algorithm [10], respectively.

In recent years, TSDF-based neural networks [11–14] have drawn increasing attention and have greatly promoted the research progress of the 3D reconstruction researches. Recent TSDF-based networks mainly focus on indoor scene/object reconstruction problems. These methods usually adopt a coarse-to-fine strategy to integrate features from different resolutions, determine whether the voxel is within the truncated distance, and finally generate the TSDF value of each voxel within the truncated distance. When the TSDF value is obtained, the marching cubes algorithm [10] can be used to obtain the 3D mesh format output.

Compared to indoor objects/scenes [15,16], the reconstruction of outdoor mountain terrains we studied in this paper has more challenges. First, outdoor scenes may have intense illumination variations due to the change of sunlight angle and intensity. Second, mountains may have a highly complex texture and topology, compared to daily objects. Figure 1 shows three types of mountains commonly seen in outdoor scenes: (a) vegetation-covered mountains, (b) rocky mountains, and (c) snowy mountains. We found that recent TSDF-based networks [17–19] are difficult to model the truncated distance near complex surfaces of the mountain where voxels are either ignored or assigned an ambiguous multi-view feature representation. The weak spatial correspondence makes it hard to learn discriminative 2D features spatially. Some depth-based methods [20,21] are also used for outdoor scene reconstruction, but they may suffer from poor result consistency.

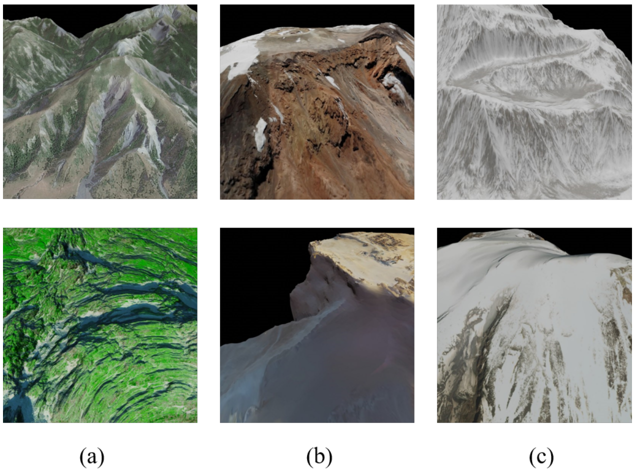

Figure 1. Different types of mountain areas: (**a**) vegetation-covered mountains, (**b**) rocky mountains, (**c**) snowy mountains.

In this paper, we propose a novel TSDF-based network for robust 3D reconstruction from remote sensing images containing mountain terrains. Our method takes in a set of images captured around an outdoor mountain and produces high-quality TSDF volumes of the mountain areas. A coarse-to-fine reconstruction pipeline is adopted, where we first fuse the 2D features extracted from different multi-view images and then fuse features from different voxel resolutions. During the multi-view feature fusion procession, we propose a view fusion strategy based on reweighted mechanisms (VRM) to better integrate 2D features on multiple views of the same voxel in 3D space. This module adaptive learns the weights for different views and makes the reconstruction focus on more important 2D image regions. We then design a spatial–temporal aggregation (SPA) module to fuse 3D features of the same location in time series based on the local neighborhood. The SPA module can reduce the ambiguity between temporal features and improve the smoothness of the reconstruction surfaces. In the feature decoding process, we notice that the large-scale depth value and complex local geometric structure of the mountains may significantly increase the difficulty of network training for conventional TSDF-based methods. To this end, we propose a feature enhancement (FE) module with probability distribution encoding [22], which can provide better discriminative geometry prior to the feature decoding process. Similar to the NeRF setting, we use a set of high-frequency sine and cosine functions to map continuous probability into a higher dimensional space to enable our decoder to more easily approximate a higher frequency signed distance function. Our motivations and observations can be summarized as follows: (1) The implicit representation for each 2D feature point can increase the discriminative ability of 2D features. The proposed VRM module can also reduce the influence of inaccurate 2D features caused by the camera distortion. (2) The geometry consistency can be better maintained by fusing temporal 3D features. We incorporate the cross-attention mechanism to eliminate the spatial instability in the feature fusion.

To verify the effectiveness of the proposed method, we build a challenging dataset that contains mountain images with different shapes, surface textures and rendered with different lighting conditions. The test data and the training data are very different in texture. For a fair comparison, we retrain the proposed model and open-source methods on this dataset. The results show that our method can better reconstruct the mountain geometry than other methods, even with limited training data. Furthermore, we also verify the proposed method on real-world multi-view terrain images from Google Map. The qualitative results show that the proposed method transfers well to real-world terrains, even those that have different styles of texture from the training data.

The main contributions of our paper are summarized as follows:

- We investigate the problem of outdoor 3D mountain reconstruction and propose a new TSDF-based reconstruction method. A challenging synthetic dataset is built for this problem.
- We propose a feature enhancement (FE) module, view fusion via reweighted mechanism (VRM), and a spatial–temporal aggregation (STA) module to effectively utilize features from different 2D views and improve feature discriminative capability on voxels. With the above design, we outperform other state-of-the-art TSDF-based methods on our task.

2. Background and Related Work

Here, we first give a brief introduction of the TSDF and then review some related topics, including depth estimation, TSDF-based reconstruction, and implicit neural representation.

2.1. Truncated Signed Distance Function (TSDF)

Truncated signed distance function (TSDF) [23] is a common method for representing surfaces in 3D reconstruction. In TSDF representation, the space occupied by the object or scene is evenly divided into several voxels. The value corresponding to each voxel represents the distance between the voxel and its nearest surface. The value is positive if the voxel is outside the surface, and negative if the voxel is inside the surface. Usually, the results of the TSDF can be converted into mesh structures using the marching cubes algorithm [10]. TSDF is an improvement based on the signed distance function (SDF), which truncates long distances and only considers the value of voxels near the surface of the object. Therefore, in the case of parallel computing of graphics cards with large memory, real-time reconstruction can be achieved by using TSDF. The truncation distance of each voxel corresponding to the current frame is written as follows:

$$tsdf = F(depth - P_z) \tag{1}$$

where the depth can be obtained by a depth camera, and the P_z represents the z coordinate value in the camera coordinate system. $F(x)$ is defined as follows:

$$F(x) = \begin{cases} min(1, \frac{x}{|u|}) & \text{x >0} \\ max(-1, \frac{x}{|u|}) & \text{x <0 ,} \end{cases} \tag{2}$$

where u represents the truncated distance. The weight of each voxel at the current frame can be calculated as $w \propto \cos\theta$, where θ is the angle between the projected ray and the surface normal vector, and the value of w is proportional to the value of $\cos\theta$. During the camera shooting process, the TSDF corresponding to the voxels is continuously updated with the following formula:

$$\begin{aligned} TSDF_{i+1} &= \frac{W_i \times TSDF_i + w_{i+1} \times tsdf_{i+1}}{W_i + w_{i+1}}, \\ W_{i+1} &= W_i + w_{i+1}. \end{aligned} \tag{3}$$

where i represents the ith view.

2.2. Depth Estimation

MVSNet [20] is one of the first to use neural networks to predict dense depth maps. MVSNet innovatively encodes camera parameters to construct cost volume and predicts depth through 3D CNNs. Later on, DPSNet [24] introduces the plane sweep algorithm where the depth prediction is transformed into a multi-class task and is learned in an end-to-end manner with neural networks. GPMVS [25] proposes a pose-kernel to measure the similarity between frames as prior information. PatchmatchNet [21] introduces cascaded patch match, which is fast, does not rely on 3D cost volume regularization, and has low memory requirements. NerfingMVS [26] proposes to use neural radiance fields (NeRF) for depth estimation and integrate learning-based depth priors into the optimization process. When the depth estimation is completed, the Poisson reconstruction [27] and Delaunay triangulation [28] are usually used to reconstruct 3D mesh results.

2.3. TSDF-Based Reconstruction

In recent years, many researchers choose to directly predict the voxel-to-surface distance for 3D reconstruction, which is also known as TSDF-based representation. Back in the 1980s, the marching cubes algorithm [10] was proposed to find the location of surfaces when obtaining TSDF results. Atlas [11] extracts 2D features of images and projects them into 3D space through corresponding camera parameters, where the 3D features are then passed through a 3D CNN to predict the TSDF value of each voxel. NeuralRecon [13] reconstructs the scene in real time by incrementally predicting the TSDF values with a GRU fusion module. Transformerfusion [14] utilizes the Transformer networks to model the relations between different views and fuse their 2D features adaptively. NerualFusion [29] performs the view feature fusion operation in a learned latent space that allows encoding additional information and improves the reconstruction by combining the view-aligned features. Most of the above-mentioned methods take a coarse-to-fine reconstruction strategy and have achieved good results. However, they are mostly designed for restricted indoor scenes and are difficult to apply to the outdoor mountain reconstruction problem studied in this paper.

2.4. Implicit Neural Representation

Implicit neural representation is a recently emerged research topic in 3D reconstruction and novel view synthesis. Implicit neural representation establishes a mapping between continuous pixel values and discrete pixel coordinates in 2D space or view coordinates in 3D space. NeRF [30] is a representative of this group of the method. NeRF-in-the-wild [31] utilizes NeRF to model wild scenes with uncontrolled images. Very recently, implicit neural representation was also introduced in tasks, such as super-resolution [32,33], image generation [34–36] and implicit 3D model generation [37–39]. A common idea of the above methods is to encode discrete coordinates with high-frequency positional encoding to improve the generative modeling capability on details. We also borrowed this idea in our proposed feature enhancement module.

3. Materials and Methods

3.1. Overview

Figure 2 shows an overview of the proposed TSDF-based method. The proposed method consists of (1) an encoder–decoder network integrated by a feature enhancement (FE) module to extract multi-scale discriminative 2D features from the input images, (2) a view fusion module based on a reweighted mechanism (VRM) to map multi-view 2D features of each scale to a 3D volume space, (3) a spatial–temporal aggregation (STA) module to enhance current voxel features by incorporating neighborhood features from the last input frames. We adopt a coarse-to-fine reconstruction pipeline and incrementally obtain detailed reconstruction results by utilizing image features from different resolutions. In each stage of the network, we verify whether the distance between each voxel and its nearest surface is smaller than the truncated distance. If yes, the voxel features are further processed by the next stage. Note that we employ memory banks to restore the latest voxel features for each stage.

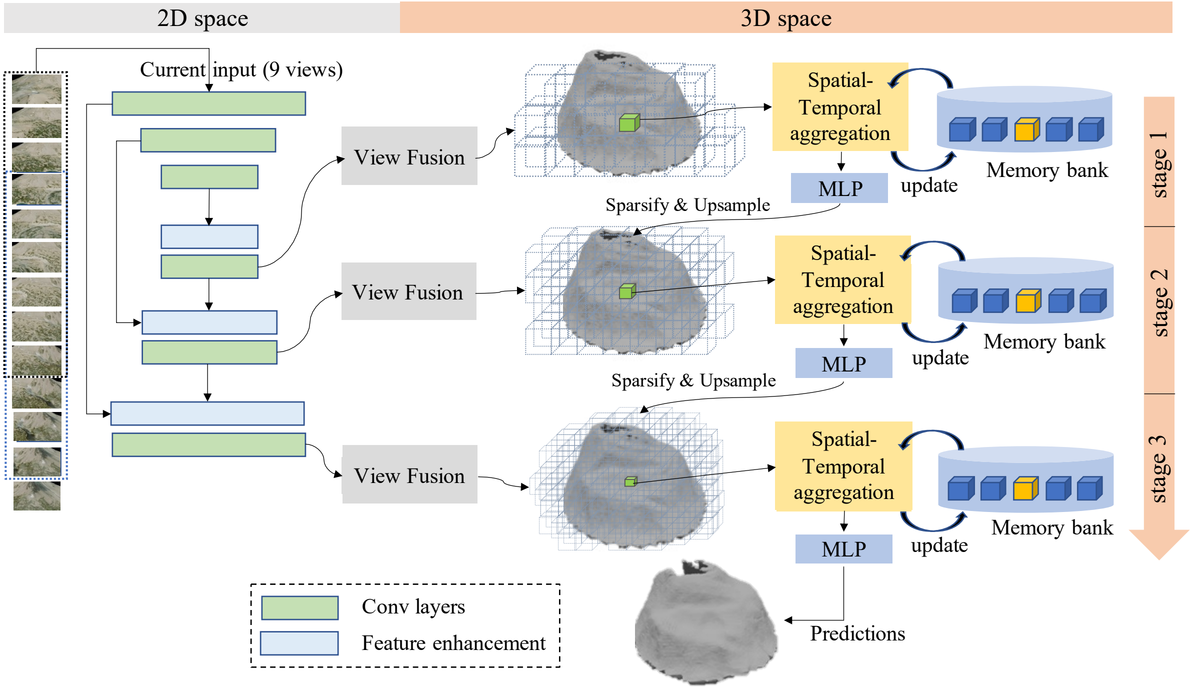

Figure 2. We adopt a coarse-to-fine pipeline to generate the reconstruction result. First, an encoder–decoder network extracts 2D features where the view fusion via reweighted mechanisms is proposed to generate more discriminative features. Then the view fusion module maps each level 2D feature to 3D volume through camera parameters. Furthermore, the spatial–temporal aggregation module eliminates the reconstruction ambiguity in timing. Finally, we update the memory bank with fused features and predict the TSDF value of each voxel.

3.2. Feature Enhancement Module

Mountain texture is an important factor in terrain reconstruction. However, because the camera is far away from the mountain, in some views, there are a lot of similar textures in the captured images (see figures in Section 4.1), which increases the difficulty for the network to extract 2D features well. In order to solve the above problem, we design a feature enhancement module to obtain discriminative features. The value of the TSDF of the voxels corresponding to the same local surface should be similar, so we first predict an implicit aggregation attribute for each 2D feature, where the features of the same aggregation category belong to the same local surface.

Given a set of 2D features $F \in \mathbb{R}^{h\times w\times c}$ (c is the channel dimension, which is {24, 40, 80}, respectively, in the process from coarse to fine) at each reconstruction stage, we employ an local aggregation network (i.e., two convolutional layers) to obtain context-rich per-pixel features $F^c = \{f_i^c \in \mathbb{R}^n |, i \in \{1, 2, 3, \cdots, hw\}$ (see Figure 3). n is {16, 32, 64} respectively in different level features. We adopt a relatively low dimension n as the number of aggregation categories. For each feature vector f_i^c, we normalize its value with a Softmax function to obtain p_i since we expect similar texture features to generate similar probability distributions. To increase the distinguishability of aggregated attributes, and inspired by implicit representation and position embedding in the transformer, we multiply the probability distribution p_i by a random Gaussian and map the result by a sin and cos function. It was proved in [40] that it can increase the discrimination of the features. The position embedding can be written as follows:

$$F^{'} = FC(\text{cat}(\sin{(2\pi PB)}, \cos{(2\pi PB)})) \tag{4}$$

where $B \in \mathbb{R}^{n\times m}$ (m is the increased feature dimension which is {24, 24, 24} respectively in the process from coarse to fine) is the random Gaussian matrix, and $FC(\cdot)$ is a fully connected layer for dimensional transformation. $F^{'}$ the same as the original feature space. Finally, we combine $F^{'}$ with the original features with element-wise addition.

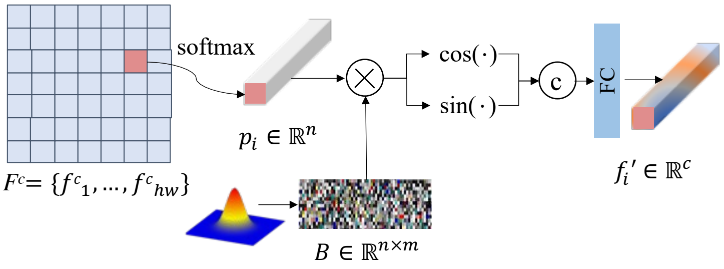
softmax
$p_i \in \mathbb{R}^n$
cos(·)
sin(·)
c
FC
$f_i' \in \mathbb{R}^c$
$F^c = \{f^c{}_1, \ldots, f^c{}_{hw}\}$
$B \in \mathbb{R}^{n\times m}$

Figure 3. Illustration of feature enhancement. A random Gaussian matrix is employed to enhance the feature discrimination capacity.

3.3. View Fusion via Reweighted Mechanisms

After extracting the 2D features for each view, we transfer the 2D features to 3D features for the subsequent process. We enclose all the images of one frame set and divide them into several voxels evenly with cubic-shaped fragment bounding volume (FBV). In the process of processing one frame set, we establish a mapping between each voxel and the 2D features by camera parameters (see Figure 4). Due to the overlapping contents between images of different views, one voxel often corresponds to multiple views. Previous methods [11,13] usually adopt simple feature averaging to integrate the multi-views 2D features. However, such an approach ignores the interaction and feature divergence between different views. The 3D information of the terrain is usually hidden in the content differences of different views. By simply averaging the features, it is difficult for the network to distinguish the feature differences between different views at the same location. In addition to this, the features at the boundary region of the view may contain incorrect information due to camera distortion. To this end, we introduce a view reweighted mechanism (VRM) for better integrating multi-view features. By considering the effects of camera distortion, we set a larger weight to center pixels, where the weights are inversely proportional to the distance between the feature location to the center of the feature map. The importance of the *i*th view feature is defined as follows:

$$\beta_i = 1/\sqrt{(h_i - h_c)^2 + (w_i - w_c)^2 + \epsilon}, \tag{5}$$

where ϵ is a positive number to prevent numeric overflow. The importance of each view is then mapped to $(0, 1)$ through a Softmax layer. Since β_i only considers the spatial location of the feature in a single view, the relationship between features at the same location across different views needs to be integrated. By following the configuration of the attention layer in the squeeze-and-excitation networks [41], the final fused feature $f_f^{'}$ are defined as follows:

$$f_f^{'} = \sum_{i=1}^{n} \gamma_i f_i^{'}, \tag{6}$$

where γ_i denotes the channel weight of the i_{th} view and is given by

$$\gamma_1, \gamma_2 \cdots \gamma_n = \mathrm{MLP}(\sum_{i=1}^{n} \beta_i f_i^{'}), \tag{7}$$

where n is the number of views and $f_i^{'}$ is the input 2D feature.

The motivation of the above spatial-weighted design is to mitigate the effects of inaccuracy features at the image boundary due to the camera distortion and distinguish the contribution of 2D features from different views to the voxel. As a result, the center features with realistic mountain textures are mainly utilized, and this design can improve the performance of fusing 2D features from different views. Note that spatial weight and channel weight are equally important.

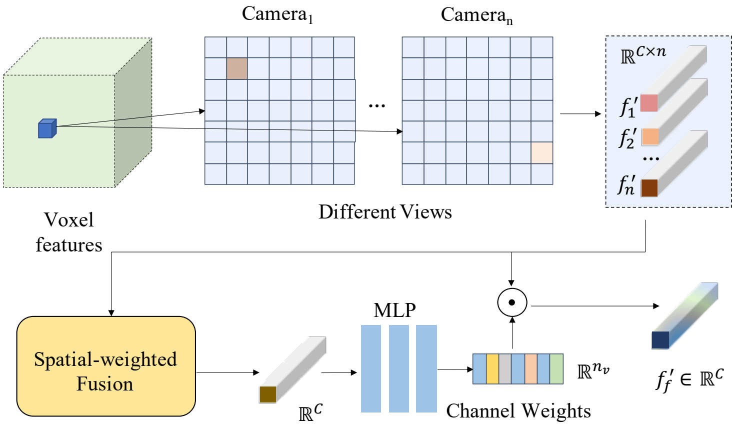

Figure 4. Illustration of the proposed view fusion method. Corresponding multi-view 2D features of each voxel are fused by (1) spatial reweighting and (2) channel reweighting.

3.4. Spatial-Temporal Aggregation Module

We divide the input images into different frame sets according to the sequence of shooting views. In order to better process 3D information, we set overlapping views in different frame sets (for details, please refer to the experimental section). Different frame sets may correspond to voxels with the same coordinates in 3D TSDF volume.

After the VRM module, the 3D feature corresponding to the same voxel are different in the fusion results of different frame-sets, which we define as temporal voxel features. The temporal voxel feature can also be understood as the same voxel that has different 3D features under different views as the camera moves. Merging different temporal voxel features benefits 3D representation learning. Very recently, NeuralRecon [13] proposed to fuse the features of the current frame with previous ones at the same locations. Note that the fusion is only performed at a single-voxel level, and we refer to this approach as point-by-point fusion. Due to the complex structure of the mountain surface, point-by-point fusion may suffer from error accumulation problem, particularly when the current and previous results both have noises.

The accumulation of errors caused by this point-to-point approach can make the predicted results inaccurate. Since the local surface of the mountain has a certain continuity in structure, the corresponding TSDF value will not change significantly. Therefore, we design a spatial–temporal aggregation module to integrate previous neighbor representations into the current voxel via a cross-attention mechanism (see Figure 5 for details).

Given a set of previous features

$$F^p = \{(f_1^p, c_1^p), (f_2^p, c_2^p), \dots (f_m^p, c_m^p)\} \tag{8}$$

and current features

$$F^s = \{(f_1^s, c_1^s), (f_2^s, c_2^s), \dots (f_n^s, c_n^s)\} \tag{9}$$

where f_i^p and f_j^s are voxel features, c_i^p and c_j^s are corresponding coordinates (the dimension of voxel features is {24, 48, 96} respectively, in different levels, and the corresponding is 3-dimension vector), we first select the top k features $f_j^p, j = \{1, ..., k\}$ closest to f_i^s in the volume space (top k). Then, we apply the multi-head cross-attention (MHCA) [42,43] to relate current voxel features and top k previous features as follows:

$$f_i^{new} = \text{MHCA}(q, K, V), \tag{10}$$

where $q = \text{MLP}(f_i^s), K/V = \{\text{MLP}(f_i^p) | j = \{1, ..., k+1\}\}$.

Since the TSDF values corresponding to the features in a neighborhood are relatively continuous, the new features that incorporate spatial neighborhood information can effectively eliminate the instability of the feature fusion. Finally, we fuse f_i^{new} with the closest previous feature (the voxel marked yellow in Figure 5) in a point-by-point way to further enhance the details. We update F^P of the memory bank with the fused feature in a dynamic fashion.

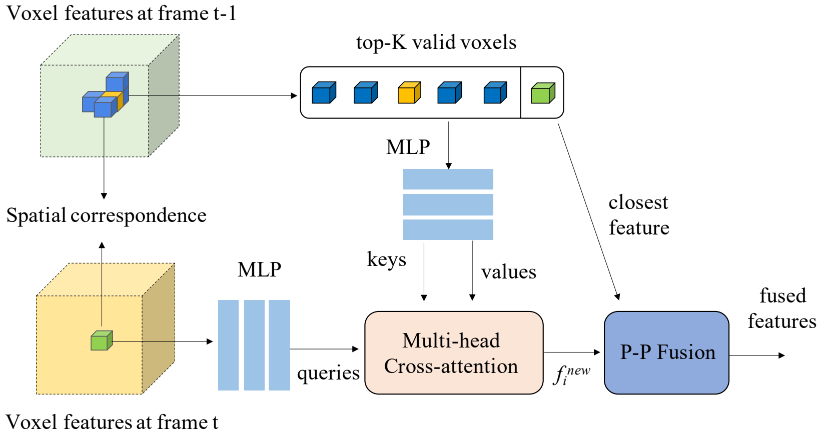

Figure 5. The multi-head cross-attention mechanism is employed to fuse the current feature with previous neighborhood ones. The resulting f_i^{new} is then fused with the closest feature (yellow voxel) in a point-by-point fusion way.

3.5. Implementation Details

Networks details. We use the pre-trained MnasNet [44] as the backbone for extracting 2D features. We use torchsparse [45] to implement 3D sparse convolution. We set the number of aggregation categories in the FE module for each stage as {16, 32, 64} respectively.

In SPA, we select 3 previous voxel features which are closets to f_j^s and send them to MHCA. The parameter details of the main modules are shown in the Table 1.

Loss function. We train the model with binary cross-entropy (BCE) loss and l_1 loss, where the former is used to tell where the voxel is within the truncation distance and the latter to regress the distance between the voxel and the surface. We follow [13] to apply log-transformation before the l_1 loss.

Each input frame-set is assigned nine views, and any two neighboring frame-sets have six overlapping frames. We employ the Adam optimizer with a 10^{-3} learning rate. Since we did not predict the depth map explicitly, we render the reconstructed mesh to the image plane and estimate the depth values [11].

To evaluate the performance of depth-based methods [24,25] in 3D metrics, we apply the standard TSDF fusion method proposed in [23] to reconstruct mesh results with depth results and compare them with ours.

Table 1. The input dimension and output dimension of main modules.

Modules	Input Demisons	OutPut Dimensions
Backbone	N_*views* × 512 × 512 × 3	Level 1:24 Level 2:40 Level 3:80
FE	The input dimensions of the three levels are correspondingly {80, 80, 80}	The output dimensions of the three levels are correspondingly {80, 80, 80}
Aggregation network	The input dimensions of Conv1 in three levels are {80,80,80}	The output dimensions of Conv1 in three levels are {40, 40, 40}
	The input dimensions of Conv2 in three levels are {40, 40, 40}	The output dimensions of Conv2 in three levels are {16, 32, 64}
VRM	The input dimensions of MLP in three levels are {24, 40, 80}	The output dimensions of MLP in three levels are {9, 9, 9}
STA	The dimension of Queries in three levels are {24, 48, 96}	The dimension of Queries in three levels are {128, 128, 128}
	The dimension of Keys in three levels are {24, 48, 96}	The dimension of Keys in three levels are {128,128,128}
	The dimension of Values in three levels are {24, 48, 96}	The dimension of Values in three levels are {128, 128, 128}
Last MLP	The input dimension of the last MLP in three levels are {24, 48, 96}	1

4. Results

4.1. Dataset

We build a diverse and challenging synthetic 3D dataset for 3D reconstruction using images containing mountain terrains. The dataset contains 3D models of 8 snowy mountains, 8 vegetation-covered mountains, and 8 rocky mountains. Some examples are shown in Figures 6 and 7.

We randomly select 4 models from each class terrain for training: 1 model for validation and 3 models for testing. As result, we render 33 video clips: 12 for training, 3 for validation, and 18 for testing. The training videos and the validation videos are obtained by shooting around the mountain at a random height with a simulated camera. The test videos are shot at two random heights for each mountain model. There is no overlap between our training and testing sets. The light intensity throughout video shooting is randomly set.

For each video, a keyframe is selected at an interval of 10 degrees. A total of 36 images are selected for one video clip, and corresponding depth images and camera parameters are also recorded. The size of each image is rendered at 512 × 512 pixels. For each video, we use the depth map and camera parameters to render the ground-truth TSDF. The size of the voxel is set to 1 m. The maximum depth is set to 120 m. The reason is that we normalize the length of all experimental models to around 120 m in the Blender coordinate system to conveniently validate our method (relative to the size of the model, this distance is very far for normal scenes). The TSDF truncation distance λ is set to 3 m.

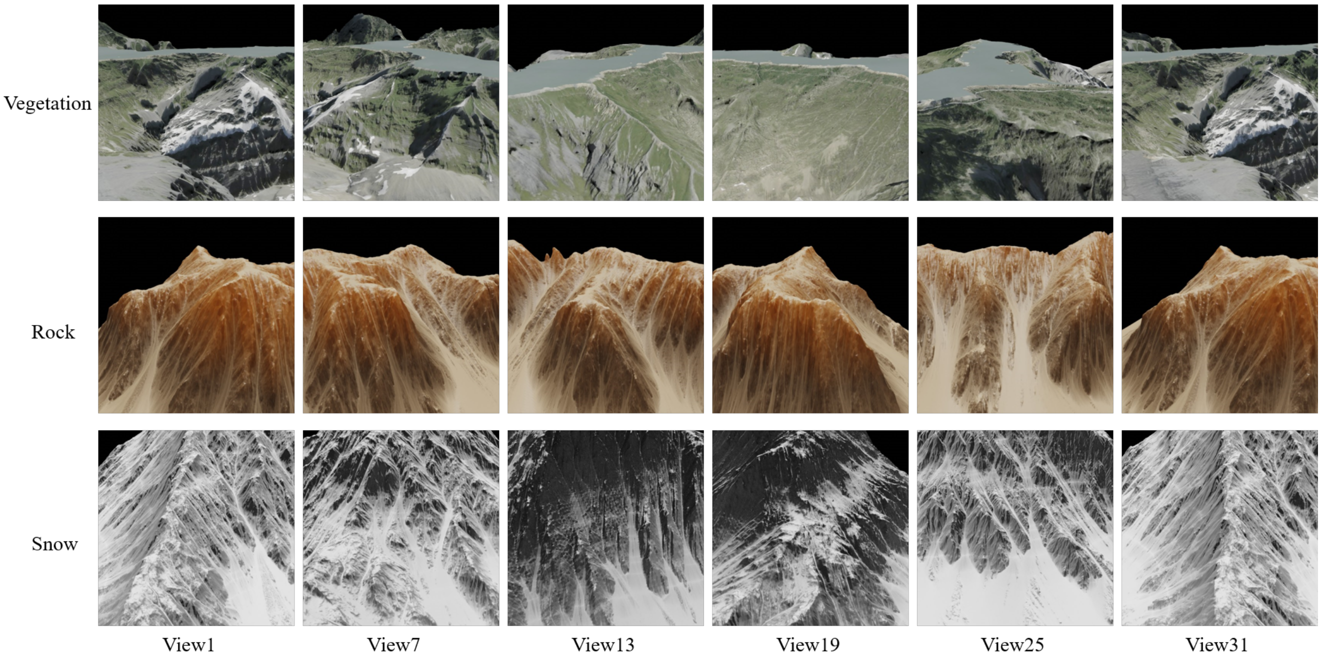

Figure 6. Input images of six camera views corresponding to each type of mountain terrain in the training set.

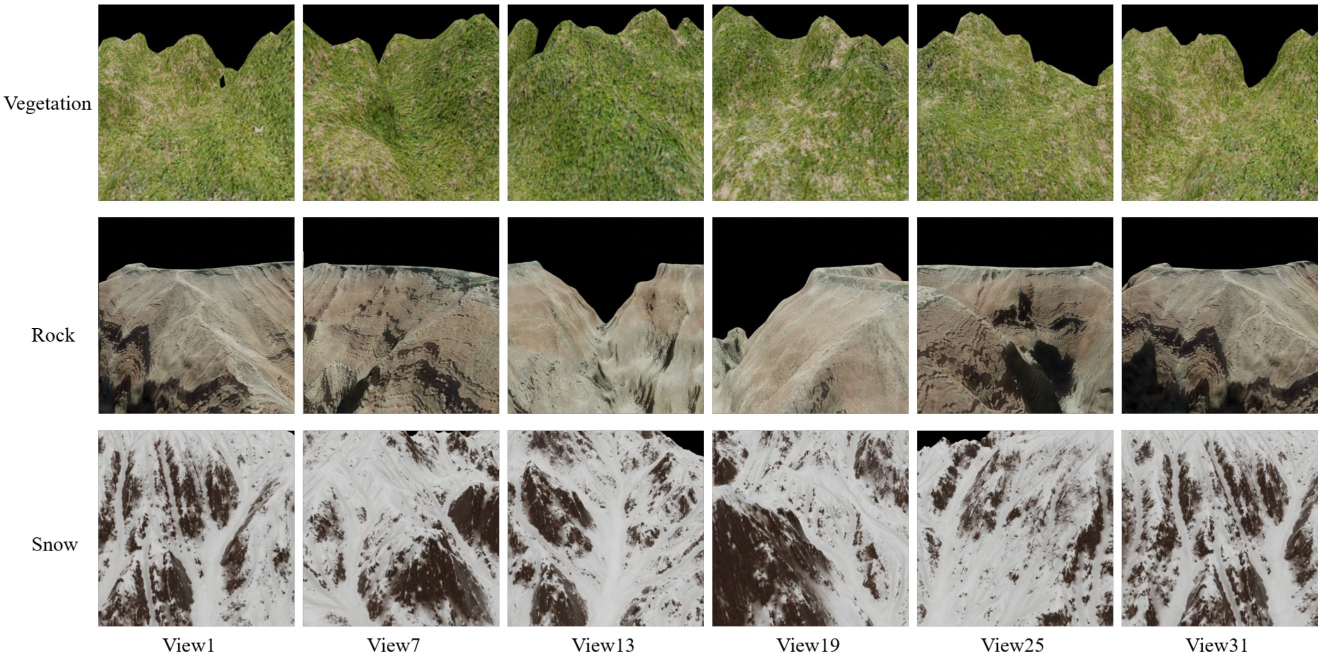

Figure 7. Input images of six camera views corresponding to each type of mountain terrain in the testing set.

4.2. Comparison Methods and Metrics

The following methods are used for comparison: (1) Traditional method, COLMAP [46], processes high-resolution images but suffers from high time consumption. (2) Depth-based methods: DPSNet [24] takes the plane swap approach to directly predict the depth map. The cost volume is constructed by a differentiable warping process, allowing end-to-end training. GPMVS [25] proposes a pose-kernel structure that encourages similar poses to have resembling latent spaces. This strategy makes the depth predicted by the model more complete. (3) TSDF-based methods: Atlas [11], an off-line method, directly regresses the TSDF value of each voxel and reconstructs the surface geometry. NeuralRecon [13] is a real-time method to reconstruct the object surfaces.

We follow [11,47], and use the 3D geometry metrics and the 2D depth metrics to compare all the above-mentioned methods. The details of each metrics are shown in Table 2. Since DPSNet and GPMVS are specialized for depth estimation, we only compare these two methods on 2D metrics. For a fair comparison, we implement all these methods on our dataset, using their open-source code.

Table 2. The evaluation metrics of 2D depth estimation and 3D reconstruction used in our experiment.

2D		3D	
Abs Rel	$\frac{1}{n}\sum\frac{\|d-d^*\|}{d^*}$	L1	$\text{mean}_{t^*<1}\|t-t^*\|$
Abs Diff	$\frac{1}{n}\sum\|d-d^*\|$	Acc	$\text{mean}_{p\in P}(\text{min}_{p^*\in P^*}\|\|p-p^*\|\|)$
Sq Rel	$\frac{1}{n}\sum\frac{\|d-d^*\|^2}{d^*}$	Comp	$\text{mean}_{p^*\in P^*}(\text{min}_{p\in P}\|\|p-p^*\|\|)$
RMSE	$\sqrt{\frac{1}{n}\sum\|d-d^*\|^2}$	Prec	$\text{mean}_{p\in P}(\text{min}_{p^*\in P^*}\|\|p-p^*\|\|)<0.05$
$\sigma<1.25^i$	$\frac{1}{n}\sum\max(\frac{d}{d^*},\frac{d^*}{d})<1.25^i$	Recall	$\text{mean}_{p^*\in P^*}(\text{min}_{p\in P}\|\|p-p^*\|\|)<0.05$
Comp	valid predictions	F-score	$(2\times\text{Recall}\times\text{Pres})\ /\ (\text{Prec}+\text{Recall})$

4.3. Overall Comparison on the Synthetic Dataset

Table 3 shows the quantitative evaluation results of different methods on the above 3D metrics. We can see our model achieves the best results compared with other neural network methods.

Table 3. The 3D reconstruction accuracy of different methods. The 1 voxel size and 1.5 voxel size represent the value of the corresponding threshold when calculating Prec and Recall. The method of calculating metric values are the same as Atlas [11] and NeuralRecon [13].

		1 Voxel Size			1.5 Voxel Size					
Method	**Layer**	**F-Score ↑**	**Prec↑**	**Recall↑**	**F-Score↑**	**Prec↑**	**Recall↑**	**Acc↓**	**Comp↓**	**Time↓ (ms)**
DPSNet	double	0.67	1.00	0.50	1.05	1.53	0.80	49.59	51.47	175
GPMVS	double	0.29	0.20	0.48	0.467	0.32	0.88	62.03	49.74	186
Altas	double	14.86	14.84	15.46	24.92	21.24	23.11	10.81	5.15	11.7
NeuralRecon	double	43.16	65.54	34.26	55.09	**84.25**	43.77	**1.10**	7.90	32.0
Ours	double	**52.16**	**65.87**	**44.26**	**64.63**	81.23	**55.00**	1.76	**4.53**	35.4
COLMAP	single	82.55	99.45	71.21	87.55	99.72	78.68	0.40	1.47	420
NeuralRecon	single	56.60	71.93	48.69	65.61	83.63	56.43	0.88	5.63	32.0
Ours	single	64.03	76.94	56.22	70.00	82.90	61.82	1.40	3.82	35.4

For the depth-based methods, the various shapes and textures of mountains make the depth-based methods difficult to learn useful geometric representations. There are also differences in the size of the mountains, so the estimated range of terrain depth also varies greatly, which makes the depth-based model unable to adapt well to different terrains. Even for the same terrain, similar textures may correspond to different depths, so texture features alone cannot generate reasonable results. The compared methods predict inconsistent depths, which makes it impossible to reconstruct the mesh results well. Our method outperforms previous methods for almost all 2D depth metrics, as shown in Table 4.

For TSDF-based neural networks, we focus more on the 3D metrics, where the F-score can better reflect the visual effects of the results. Since 3D information is often contained in the disparity of different views, the joint processing of features from different views is important. However, Atlas [11] extracts the features of each view separately, which results in inconsistent semantic information in the 3D space, and results in a low F-score. NeuralRecon [13] integrates the temporal features at a single-voxel level to improve robustness. However, due to its relatively insufficient 2D features and the possibility of error accumulation of voxel features, some voxels are deleted by mistake, resulting in a low recall rate and poor integrity. Compared with NeuralRecon, our proposed model improves the F-score and recall by 9.5 % and 11.3%, respectively. From the visualization results in Figure 8, we can see our method can have a more complete reconstruction of the mountain surface. Note that we do not show the colored results because our paper focuses on TSDF-based representations rather than textures. In addition, in order to compare with COLMAP, we use the same way as that in NerualRecon [13] to generate single-layer results. Although the results of COLMAP in Figure 8 are more complete, COLMAP produces the wrong topology, with some holes being filled incorrectly and over-smooth results. Table 3 shows that our method surpasses NerualRecon in almost all 3D metrics. Although, COLMAP can achieve a higher F-score but can be 10 times slower than ours.

Table 4. Depth estimation accuracy of different methods. We use the same method in NeuralRecon [13] to calculate the metric values.

Method	Abs Rel↓	Abs Diff↓	Sq Rel↓	RMSE↓	$\delta < 1.25^i$ ↑	Comp↑
GPMVS	0.90	44.74	60.92	50.39	0.17	84.04
DPSNet	0.44	24.80	14.77	27.09	0.29	**84.10**
NerualRecon	0.12	3.23	1.71	5.02	0.83	70.22
Ours	**0.04**	**2.16**	**0.64**	**3.81**	**0.97**	63.90

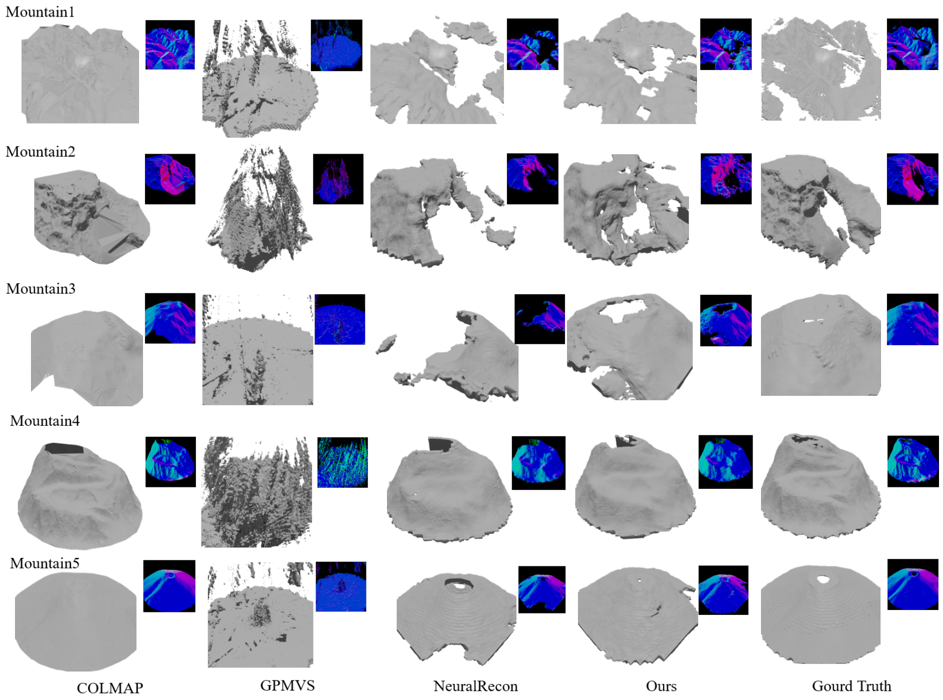

Figure 8. Qualitative results on our dataset. Compared to GPMVS [25] and NerualRecon [13], our method can produce much more complete reconstruction results. Notice that COLMAP generates single layer results and tends to give over-smooth results (some holes are wrongly filled).

Our results are TSDF-based forms that can be easily converted to mesh, point cloud and other formats. These are the explicit 3D data formats that can be simply imported into software, such as Unity, for further editing (modifying the terrain, combining multiple terrains, etc.). In this way, it will greatly reduce modeling time and effort.

4.4. Generalizing to Real Data

To further verify the adaptability of the proposed method to the real-world data, we randomly extract multi-view terrain images from Google Map. The real terrains include one vegetation-covered mountain, one rocky mountain, and one snowy mountain. The real data have different styles of textures from training data. The validation videos are also obtained by shooting around the real terrains at a random height.

There are two challenges on real terrains: (1) the texture of real terrains is very different from that of training mountains; and (2) the real terrains have a more complex geometrical structure. The above increases the difficulty of reconstructing and requires the model to be able to adapt to different types of textures.

Since it is difficult to obtain the ground truth of real terrains, we only show the qualitative results. In Figures 9–11, we show three groups of reconstruction results using our method. The small images on the left are the six views randomly shot by the camera, and the large image on the right is the reconstruction result. The results show that our method trained on synthetic data has good transferability to real terrains.

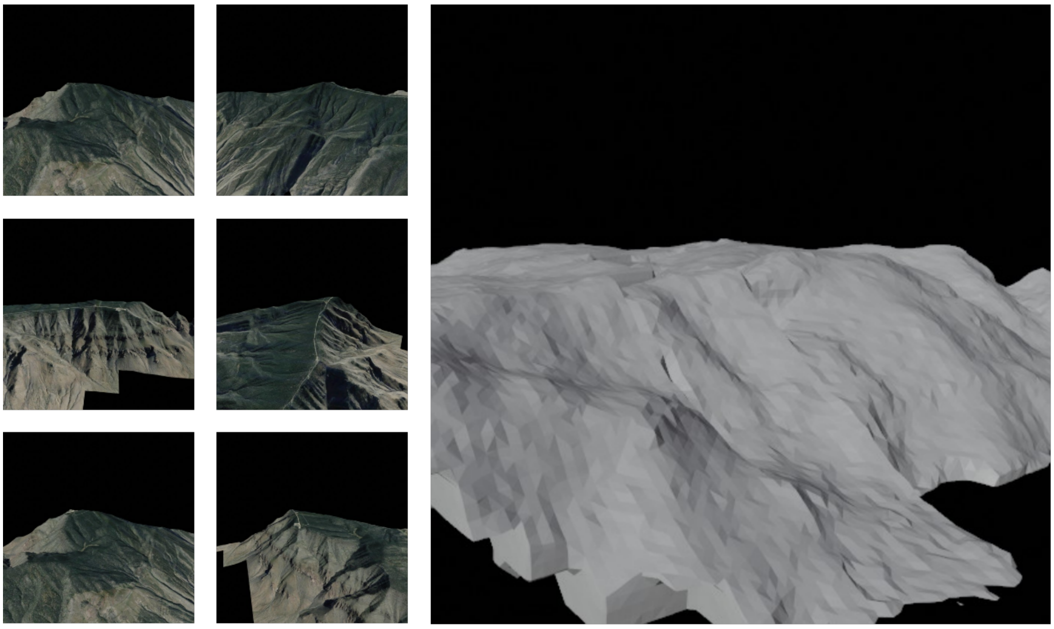

Figure 9. The real data of vegetation-covered mountain terrain. The small images on the left are the six views randomly shot by the camera, and the large image on the right is the reconstruction result.

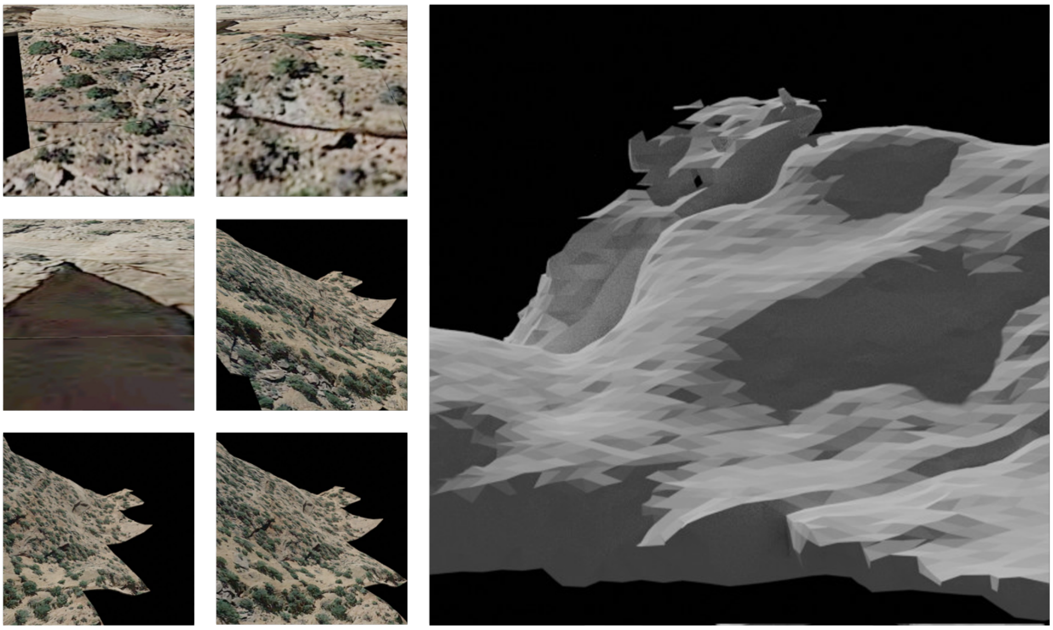

Figure 10. The real data of rocky mountain terrain. The small images on the left are the six views randomly shot by the camera, and the large image on the right is the reconstruction result.

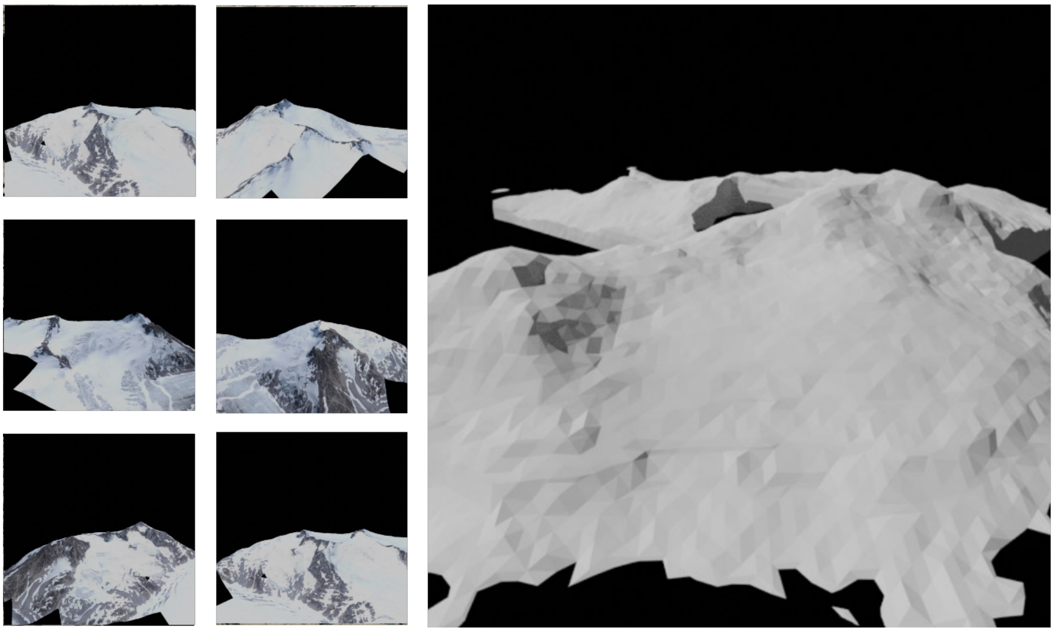

Figure 11. The real data of snowy mountain terrain. The small images on the left are the six views randomly shot by the camera, and the large image on the right is the reconstruction result.

5. Discussion

In this section, we analyze carefully the benefits and drawbacks of the proposed strategy. To demonstrate the effectiveness of our method, we incrementally add the proposed modules (FE, VRM, STA) to the baseline in Section 5.1. In Section 5.2, we focus on the limitations of our model.

5.1. Ablation Study

To verify the effectiveness of our designed modules, we conduct ablation studies on the proposed three key components, i.e., feature enhancement (FE), view reweighted mechanism (VRM), and spatial–temporal aggregation (STA). Starting from a baseline using a normal 2D encoder–decoder and normal view fusion-based 3D feature generator, we incrementally add the three components.

From Table 5, we can observe that STA, VRM, and FE can bring improvement of the F-score of our model by 8.7%, 3.7%, and 2.2%, respectively. The above results show that the modules we designed for image reconstruction can improve the performance of the network and achieve better reconstruction results. We can also observe that our STA induces further improvement of F1-score against NeurlRecon by 3.7%. The quantitative results indicate that the aggregation with neighborhood voxel features can benefit the 3D reconstruction. We also visualize the reconstruction results of our models in Figure 12. We can observe that the reconstruction quality improves when incrementally adding the three modules. When the STA module is added, the top area of the terrain becomes flat and the whole is smoother. This is because the local spatial features can reduce the ambiguity between temporal features.

The qualitative comparison of our ablation experiment is shown in Figure 12. In Figure 12d, we can see that the VRM module can better convert 2D features into 3D features. As a result, the results become more complete. The result of Figure 12e also removes some discontinuous regions, which shows that the FE module can improve the network to extract more distinguishable features from 2D images, which is more conducive to predicting an accurate TSDF for each voxel. In conclusion, both quantitative and qualitative experiments show that our proposed module is beneficial for the task of terrain reconstruction.

Table 5 also shows the memory usage in testing process. The memory computation is reduced with our proposed modules due to the powerful ability of added modules to neglect the computation of a lot of voxels that are not near the mountain surface.

Table 5. Ablation study on the proposed three modules, i.e., feature enhancement (FE), view reweighted mechanism (VRM), and spatial–temporal aggregation (STA).

Method	FE	VRM	STA	F-Score	Prec	Recall	Memory Consumption
baseline	×	×	×	50.09	77.55	39.44	2.2 GB
NeuralRecon	×	×	×	55.09	84.25	43.77	<4 GB
ours	√	×	×	60.68	67.22	57.74	3.7 GB
ours	×	√	×	62.23	75.72	55.35	3.6 GB
ours	×	×	√	58.79	76.49	48.93	4.4 GB
ours	√	√	×	62.72	77.44	55.37	2.1 GB
ours	×	√	√	62.45	75.70	54.71	2.2 GB
ours	√	×	√	63.12	70.31	58.86	2.1 GB
ours	√	√	√	64.63	81.23	55.00	2.2 GB

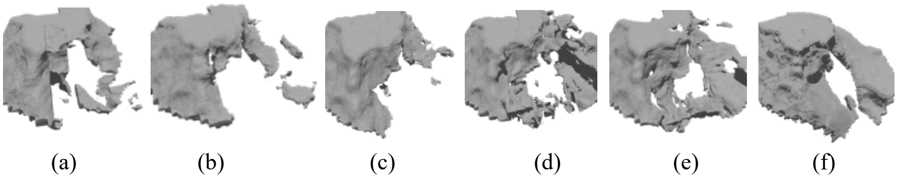

Figure 12. The visualization result of ablation experiments. (**a**) Our baseline; (**b**) NeuralRecon (**c**) STA; (**d**) STA+VRM; (**e**) STA+VRM+FE; (**f**) ground truth.

5.2. Limitations and Future

As we discussed above, our designed module can improve the integrity and accuracy of image reconstruction that contains mountain terrains. However, the textures of the images in our current dataset are blurry, so applying textures to the reconstructed model is not the focus of our work. Our further work is to reconstruct multi-view high-definition images that can paste clear textures on the reconstructed terrain. In the future, we will collect more types of mountains to expand our dataset. Our model can be more fully trained to adapt to different input images, which in turn can achieve better reconstruction of the real terrain in Google Map. Furthermore, we are also interested in researching that reconstructing images captured from widely varying distances, which is a very interesting but challenging problem. The key is to extract effective texture features from multi-resolution images, which is very important and critical in many tasks.

6. Conclusions

In this paper, we propose a novel TSDF-based method for 3D reconstruction from images containing mountain terrains. Different from existing methods that only fuse temporal features of the same voxel, we introduce neighbor information to smooth the current voxel representation via cross-attention. We further propose feature enhancement and reweighted mechanisms to enhance the discriminative capacity of 2D features. Extensive experiments on our proposed dataset verify the effectiveness of our method. Our method outperforms several state-of-the-art methods in terms of both 3D and 2D metrics. The visual evaluation also illustrates the completeness and refinement of our reconstruction. In addition, the results on real terrains show that our method has a good ability to adapt to real styles of textures. Our method can effectively learn the geometry of the mountain from a small number of videos.

Author Contributions: Conceptualization, Z.Z.; methodology, Z.Q., Z.Z. and Z.S.; validation, Z.Q.; formal analysis, Z.Q. and Z.Z.; writing—original draft preparation, Z.Q.; writing—review and editing, Z.Q., H.C. and Z.Z.; visualization, Z.Q. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported in part by the National Natural Science Foundation of China under the Grant 62125102.

Data Availability Statement: Not applicable.

Acknowledgments: Thanks to all editors and commenters.

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

References

1. Li, H.; Chen, S.; Wang, Z.; Li, W. Fusion of LiDAR data and orthoimage for automatic building reconstruction. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2010, Honolulu, HI, USA, 25–30 July 2010.
2. Xiong, B.; Jiang, W.; Li, D.; Qi, M. Voxel Grid-Based Fast Registration of Terrestrial Point Cloud. *Remote Sens.* **2021**, *13*, 1905. [CrossRef]
3. Schiavulli, D.; Nunziata, F.; Migliaccio, M.; Frappart, F.; Ramilien, G.; Darrozes, J. Reconstruction of the Radar Image From Actual DDMs Collected by TechDemoSat-1 GNSS-R Mission. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, 9, 4700–4708. [CrossRef]
4. Aghababaee, H.; Ferraioli, G.; Schirinzi, G.; Pascazio, V. Corrections to "Regularization of SAR Tomography for 3-D Height Reconstruction in Urban Areas" [Feb 19 648–659]. *Sel. Top. Appl. Earth Obs. Remote Sens. IEEE J.* **2019**, *12*, 1063. [CrossRef]
5. Wang, M.; Wei, S.; Shi, J.; Wu, Y.; Tian, B. CSR-Net: A Novel Complex-valued Network for Fast and Precise 3-D Microwave Sparse Reconstruction. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 4476–4492. [CrossRef]
6. Yang, Y.C.; Lu, C.Y.; Huang, S.J.; Yang, T.Z.; Chang, Y.C.; Ho, C.R. On the Reconstruction of Missing Sea Surface Temperature Data from Himawari-8 in Adjacent Waters of Taiwan Using DINEOF Conducted with 25-h Data. *Remote Sens.* **2022**, *14*, 2818. [CrossRef]
7. Zhang, E.; Fu, Y.; Wang, J.; Liu, L.; Yu, K.; Peng, J. MSAC-Net: 3D Multi-Scale Attention Convolutional Network for Multi-Spectral Imagery Pansharpening. *Remote Sens.* **2022**, *14*, 2761. [CrossRef]
8. Williams, S.; Parker, L.T.; Howard, A.M. Terrain Reconstruction of Glacial Surfaces: Robotic Surveying Techniques. *Robot. Autom. Mag. IEEE* **2012**, *19*, 59–71. [CrossRef]
9. Kazhdan, M.; Bolitho, M.; Hoppe, H. Poisson surface reconstruction. In Proceedings of the Fourth Eurographics Symposium on Geometry Processing, Cagliari Sardinia, Italy, 26–28 June 2006; Volume 7.
10. Lorensen, W.E.; Cline, H.E. Marching cubes: A high resolution 3D surface construction algorithm. *ACM Siggraph Comput. Graph.* **1987**, *21*, 163–169. [CrossRef]

11. Murez, Z.; van As, T.; Bartolozzi, J.; Sinha, A.; Badrinarayanan, V.; Rabinovich, A. Atlas: End-to-end 3d scene reconstruction from posed images. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Part VII 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 414–431.
12. Choe, J.; Im, S.; Rameau, F.; Kang, M.; Kweon, I.S. Volumefusion: Deep depth fusion for 3d scene reconstruction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11 October 2021; pp. 16086–16095.
13. Sun, J.; Xie, Y.; Chen, L.; Zhou, X.; Bao, H. NeuralRecon: Real-Time Coherent 3D Reconstruction from Monocular Video. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15598–15607.
14. Bozic, A.; Palafox, P.; Thies, J.; Dai, A.; Nießner, M. Transformerfusion: Monocular rgb scene reconstruction using transformers. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 1403–1414.
15. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H. ShapeNet: An Information-Rich 3D Model Repository. *arXiv* **2015**, arXiv:1512.03012.
16. Li, W.; Saeedi, S.; McCormac, J.; Clark, R.; Tzoumanikas, D.; Ye, Q.; Huang, Y.; Tang, R.; Leutenegger, S. Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. *arXiv* **2018**, arXiv:1809.00716.
17. Grinvald, M.; Tombari, F.; Siegwart, R.; Nieto, J. TSDF++: A Multi-Object Formulation for Dynamic Object Tracking and Reconstruction. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021.
18. Kim, H.; Lee, B. Probabilistic TSDF Fusion Using Bayesian Deep Learning for Dense 3D Reconstruction with a Single RGB Camera. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2020.
19. Hardouin, G.; Morbidi, F.; Moras, J.; Marzat, J.; Mouaddib, E.M. Surface-driven Next-Best-View planning for exploration of large-scale 3D environments. *IFAC-PapersOnLine* **2020**, *53*, 15501–15507. [CrossRef]
20. Yao, Y.; Luo, Z.; Li, S.; Fang, T.; Quan, L. Mvsnet: Depth inference for unstructured multi-view stereo. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 767–783.
21. Wang, F.; Galliani, S.; Vogel, C.; Speciale, P.; Pollefeys, M. PatchmatchNet: Learned Multi-View Patchmatch Stereo. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 14194–14203.
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
23. Newcombe, R.A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A.J.; Kohi, P.; Shotton, J.; Hodges, S.; Fitzgibbon, A. Kinectfusion: Real-time dense surface mapping and tracking. In Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality, Basel, Switzerland, 26–29 October 2011; pp. 127–136.
24. Im, S.; Jeon, H.G.; Lin, S.; Kweon, I.S. Dpsnet: End-to-end deep plane sweep stereo. *arXiv* **2019**, arXiv:1905.00538.
25. Hou, Y.; Kannala, J.; Solin, A. Multi-view stereo by temporal nonparametric fusion. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 2651–2660.
26. Wei, Y.; Liu, S.; Rao, Y.; Zhao, W.; Lu, J.; Zhou, J. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11 October 2021; pp. 5610–5619.
27. Kazhdan, M.; Hoppe, H. Screened poisson surface reconstruction. *ACM Trans. Graph. (ToG)* **2013**, *32*, 1–13. [CrossRef]
28. Labatut, P.; Pons, J.P.; Keriven, R. Robust and efficient surface reconstruction from range data. In *Proceedings of the Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2009; Volume 28, pp. 2275–2290.
29. Weder, S.; Schonberger, J.L.; Pollefeys, M.; Oswald, M.R. NeuralFusion: Online Depth Fusion in Latent Space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 3162–3172.
30. Mildenhall, B.; Srinivasan, P.P.; Tancik, M.; Barron, J.T.; Ramamoorthi, R.; Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 405–421.
31. Martin-Brualla, R.; Radwan, N.; Sajjadi, M.S.; Barron, J.T.; Dosovitskiy, A.; Duckworth, D. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 7210–7219.
32. Chen, Y.; Liu, S.; Wang, X. Learning continuous image representation with local implicit image function. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8628–8638.
33. Xu, X.; Wang, Z.; Shi, H. UltraSR: Spatial Encoding is a Missing Key for Implicit Image Function-based Arbitrary-Scale Super-Resolution. *arXiv* **2021**, arXiv:2103.12716.
34. Skorokhodov, I.; Ignatyev, S.; Elhoseiny, M. Adversarial generation of continuous images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10753–10764.
35. Anokhin, I.; Demochkin, K.; Khakhulin, T.; Sterkin, G.; Korzhenkov, D. Image Generators with Conditionally-Independent Pixel Synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021.
36. Dupont, E.; Teh, Y.W.; Doucet, A. Generative Models as Distributions of Functions. *arXiv* **2021**, arXiv:2102.04776.

37. Chen, Z.; Zhang, H. Learning Implicit Fields for Generative Shape Modeling. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
38. Mescheder, L.; Oechsle, M.; Niemeyer, M.; Nowozin, S.; Geiger, A. Occupancy Networks: Learning 3D Reconstruction in Function Space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
39. Park, J.; Florence, P.; Straub, J.; Newcombe, R.; Lovegrove, S. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
40. Tancik, M.; Srinivasan, P.P.; Mildenhall, B.; Fridovich-Keil, S.; Raghavan, N.; Singhal, U.; Ramamoorthi, R.; Barron, J.T.; Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. *arXiv* **2020**, arXiv:2006.10739.
41. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
42. Wen, Z.; Lin, W.; Wang, T.; Xu, G. Distract Your Attention: Multi-head Cross Attention Network for Facial Expression Recognition. *arXiv* **2021**, arXiv:2109.07270.
43. Xu, X.; Hao, J. U-Former: Improving Monaural Speech Enhancement with Multi-head Self and Cross Attention. *arXiv* **2022**, arXiv:2205.08681.
44. Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; Le, Q.V. Mnasnet: Platform-aware neural architecture search for mobile. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2820–2828.
45. Tang, H.; Liu, Z.; Zhao, S.; Lin, Y.; Lin, J.; Wang, H.; Han, S. Searching efficient 3d architectures with sparse point-voxel convolution. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 685–702.
46. 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.
47. Eigen, D.; Puhrsch, C.; Fergus, R. Depth map prediction from a single image using a multi-scale deep network. *arXiv* **2014**, arXiv:1406.2283.