3-D Model Extraction Network Based on RFM-Constrained Deformation Inference and Self-Similar Convolution for Satellite Stereo Images

Wen Chen, *Student Member, IEEE*, Hao Chen, *Member, IEEE*, and Shuting Yang

***Abstract*—Traditional three-dimensional (3-D) reconstruction methods for satellite stereo images (SSIs) are limited by observation angles and image resolution, resulting in poor reconstruction results and only a rough 3-D model of the extracted target. Meanwhile, deep-learning methods require a large number of training samples and restoring the complete 3-D structure of the target is challenging when it is quite different from the training sample. To address these problems, we propose a 3-D extraction method for SSIs based on self-similar convolution and a deformation inference network constrained by a rational function model (RFM). Inspired by the implicit relationship between 2-D image features and 3-D shapes, we construct a 2-D–3-D mapping relationship to mine the depth features of remote-sensing images by incorporating the RFM. The deformation result of each point in the point cloud is inferred by a graph convolution network to iteratively optimize the 3-D reconstruction effect of the visible surface. We construct the self-similar convolution module by utilizing the self-similarity characteristics existing in the target itself. The reconstruction results of the invisible surface are optimized while establishing the mesh vertex connection relationship. Experiments on multiple datasets show that the target reconstruction results of our method outperform those of other classical methods, and the relative accuracy of root-mean-square error for targets such as buildings, planes, and ships can reach up to 3 m or less. The accuracy of the earth mover's distance is better than 0.5.**

***Index Terms*—Deformation inference, satellite stereo images (SSIs), self-similar convolution, three-dimensional (3-D) model extraction.**

I. Introduction

WITH the ongoing development of satellite imaging capabilities, satellite platforms can acquire local remote-sensing images without requiring personnel to travel to the shooting location. Across civilian and defense industries, there is an increasing demand for three-dimensional (3-D) reconstructions based on satellite stereo images (SSIs), especially for 3-D structures of ground targets, which have become a focal point for research in recent years.

Traditional 3-D reconstruction for SSIs typically involves processing epipolar geometry constraints, stereo matching, coordinate solving, and determining strict geometric constraint relationships to obtain a digital surface model (DSM). The 3-D model of the target is mostly directly extracted from DSM [1]. Most of these traditional 3-D target extraction methods utilize target characteristics to constrain the DSM or remote-sensing images, usually based on a series of strict assumptions and rules, and the application of the scene and conditions are relatively harsh. However, the DSM data obtained by SSIs are limited by the observation angle and image resolution and, therefore, cannot capture information regarding the hidden surfaces of the target. Moreover, most target types are simple buildings, and it is difficult to reconstruct targets with more complex structures such as planes and ships.

With the development of artificial intelligence techniques, deep-learning networks utilizing computer vision can encode rich a priori information in 3-D graphical space to generate 3-D shapes of various object types, which can then be used to create 3-D point clouds or meshes for the target of interest directly through an end-to-end neural network, thereby bypassing the processes of stereo matching and coordinate solving involved in traditional photogrammetry. Particularly in recent years, research on single-view 3-D reconstruction in computer vision has attracted considerable attention. Most studies use methods based on 3-D convolutional neural networks (CNNs) or generative adversarial networks (GANs) to establish the relationship between the target image and the 3-D model of the target [e.g., occupancy networks (O-net) [2]]. On this basis, recovering 3-D geometry from many views via correspondence established by photo consistency has been well studied [3]. However, generating a 3-D model using fewer views is more challenging and less well-solved.

Although these deep-learning-based 3-D reconstruction methods have been well validated in the field of computer vision, there are problems when applying them to remote-sensing images, especially SSIs, which are summarized as follows.

1) Limited by the range of scenarios for computer vision applications, studies often focus on simple targets appearing in daily life and utilize mesh-connection relationships for complex ground targets such as buildings and ships commonly seen in SSIs, despite difficulties regarding mesh self-intersection in 3-D reconstructions.
2) Deep-learning networks depend on a large number of training samples to construct a correspondence between the image and the target. In the field of computer vision, with a large number of publicly available datasets, this

Manuscript received 15 January 2024; revised 6 May 2024 and 30 May 2024; accepted 15 June 2024. Date of publication 27 June 2024; date of current version 12 July 2024. *(Corresponding author:Hao Chen)*

The authors are with the Department of Information Engineering, Harbin Institute of Technology, Harbin 150000, China (e-mail: 1152878181@qq.com; hit_hao@hit.edu.cn; 21b905002@stu.hit.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2024.3419896

end-to-end reconstruction algorithm has achieved good results. But the cost of obtaining SSIs data is relatively high. Even if SSIs are obtained, the cost of obtaining the corresponding high-precision 3-D model of the target is still high. This makes it difficult to recover the 3-D structure of the target in testing when there are limited training samples available.

3) The reconstructed target is a normalized model that cannot directly use parameter information from the satellite image to describe the real-world size and scale of the target. Most computer vision networks are trained and tested on the target scale normalized datasets. This results in the reconstructed target being unable to be directly converted to real-world scales, requiring conversion based on the image scale. However, the resolution of SSIs is relatively low, which can easily lead to errors in the scale conversion process.

To address these problems, this article proposes a 3-D extraction method for SSIs based on self-similar convolution and a deformation inference network constrained by a rational function model (RFM). By combining deep-learning network feature mining and satellite remote-sensing detection mechanisms, the proposed method uses a deep-learning network to continuously optimize the 3-D model based on the rough results of the target extracted by the DSM in order to progressively construct a 3-D model of the target from an initial point cloud to a point-cloud deformation and then perform mesh optimization.

II. Related Work

Extracting the 3-D model of the target directly from DSM is an important means of traditional target 3-D reconstruction. Tripodi et al. [4] proposed a method for 3-D reconstructions of buildings based on SSI pairs using U-net to extract the contour polygons of a building while combining DSM and a digital elevation model to extract the building model correctly. Wang et al. [5] proposed a machine-learning-based method for automatic 3-D building reconstruction and vectorization that takes DSM and panchromatic images as inputs and combines a conditional GAN with a semantic segmentation network to extract the building target model while constructing the roof polygons. Qian et al. [6] proposed a method utilizing line segments of 3-D structures that abstracts the target in the remote-sensing image into a line segment, then groups the line segments into a small rectangle to realize a simplified 3-D spatial construction. Son and Soon-Yong [7] propose a 3-D reconstruction scheme from a single image with a deep monocular depth estimation network. By replacing the depth estimation loss function with the height estimation loss function, expand it to height estimation focused building from remote sensing images.

On the other hand, various data-driven CNN methods have been introduced into the field of 3-D reconstruction. Choy et al. [8] proposed a 3-D recurrent reconstruction neural network (3-D-R2N2) that learns a mapping from the target image to its underlying 3-D shapes from a large amount of synthesized data. Kar et al. [9] proposed an end-to-end learnt stereo machine (LSM) for multiview vision using feature projection and back projection along the observation rays to recover the 3-D shapes. Gwak et al. [10] constructed a ray-tracing pooling layer on top of a GAN to implement perspective projection and backpropagation to reconstruct the 3-D shapes using 2-D mask markers. Chapell et al. [11] experimented with neural radiance fields to generate 3-D meshes using our city-scale image dataset captured from drones and crewed aircraft flights in a circular orbit. However, it is difficult to apply to SSIs with fewer angles.

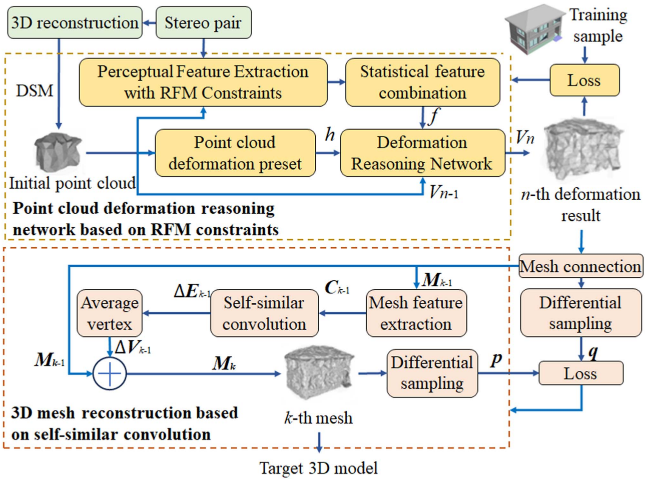


Fig. 1. Framework of the proposed method.

III. Methodology

Our method consists of two parts. The first part is to use the rough target model extracted from the DSM as the initial point cloud, then perform point-cloud deformation, and finally iteratively optimize the 3-D model of the visible surface. Inspired by the implicit relationship between 2-D image features and 3-D shapes, the RFM establishes a mapping relationship between point clouds and images, thereby extracting the remote-sensing image perception features corresponding to each vertex in the point cloud. As the 3-D model can be considered to be a graph structure, we construct a point-cloud deformation inference network on graph convolutional networks (GCNs). The position of the real point cloud is inferred using the extracted perceptual features so that the initial point cloud gradually approaches the real point cloud. The second part is to use the rough mesh reconstructed from the point-cloud surface as the initial mesh, perform self-similar mesh convolution, and optimize the mesh-connection relationship and the reconstructed results of the hidden surfaces of the target. Considering the self-similarity characteristics of the target itself, i.e., the existence of many similar structures, by combining MeshCNN with the idea of weight sharing, a self-similar convolution module is constructed to convolve the mesh and optimize and update the 3-D mesh, while ensuring the existence of loops and similar structures on hidden surfaces. The proposed method is shown in Fig. 1.

A. Point Cloud Deformation Inference Network Based on RFM Constraints

We first construct a point-cloud deformation inference network based on RFM constraints, which we use to deform the initial point cloud to a point cloud closer to the actual target. The proposed network structure is shown in the upper half of Fig. 1. First, the rough model extracted from DSM is used as the initial point cloud, where each vertex is preset with possible positions to move to. Then, the RFM is used to establish the mapping relationship between the point cloud and the image, ensuring that each point in the 3-D point cloud corresponds to the pixels of the 2-D image, thereby extracting the perceptual features of the 2-D image through a CNN. This feature has an implicit relationship with the corresponding 3-D vertex. Afterward, statistical features are obtained through feature concatenation, ensuring that the method can be applied to any number of views. Finally, by utilizing the similarity between the 3-D model and the graph structure, point clouds and their connections are considered as vertices and edges in the graph structure, thus constructing a point-cloud deformation inference network using GCN. By using inference networks and combining the extracted perceptual features, the optimal point-cloud deformation position is inferred as the initial point cloud gradually approaches the actual point cloud.

1) Initial Point Cloud: First, construct an initial point cloud for deformation and preset the possible positions of each vertex. Although there are various issues with how traditional 3-D reconstruction for SSIs produces a DSM, it can still describe the rough 3-D shape of the target. Therefore, we up-sample the target point cloud extracted from the DSM as the initial point cloud. The scale of up-sampling is influenced by the size of the target and the resolution of the remote-sensing image. In addition, to limit the range of point-cloud movement and better describe the deformation process of the point cloud, preset positions are proposed for each vertex of the initial point cloud after up-sampling, that is, the positions where vertices may move to are preset in 3-D space. Here, we sample k points on a sphere with a radius of the preset vertex movement distance (taking into account the target type and scale involved in this article, $k = 42$) as preset vertices, where the preset vertex movement distance is the average distance between all vertices and their nearest neighbors.

2) Feature Extraction: After constructing the initial point cloud, the perceptual features of SSIs can be extracted by combining the mechanism of satellite stereo-imaging detection. The fundamental technique in stereo imaging is triangulation, which determines that when one side length and two observation angles are known, the observed target point can be calibrated as the third point of a triangle. In satellite stereo imaging, the positional information of ground target points can be calculated by knowing the homologous points and angles of different images (often encrypted in the rational polynomial coefficient). Therefore, based on the above-mentioned process, ground target points can be used to deduce the homologous points in different images. These homologous points are the dimensionality reduction mapping of ground target points from 3-D space to 2-D images, that is, there is an implicit relationship between 2-D image features and 3-D shapes. Therefore, we describe this relationship by constructing a point-cloud deformation inference network, which continuously deforms 3-D spatial points to ideal ground-truth points under the guidance of 2-D image perception features.

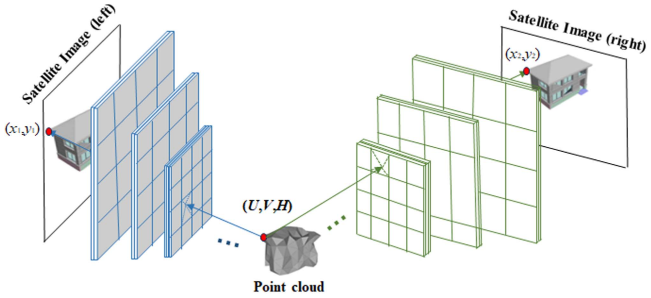


Fig. 2. Perceptual feature extraction based on a satellite-imaging mechanism.

First, the RFM is used to construct the corresponding relationship between ground target points and image points with the same name and then the perceptual features of remote-sensing images are extracted. The projection relationship between the vertex of the initial point cloud and the optical image is

$$x = \frac{\mathrm{Numl}(U,V,H)}{\mathrm{Denl}(U,V,H)}, y = \frac{\mathrm{Nums}(U,V,H)}{\mathrm{Dens}(U,V,H)} \tag{1}$$

where U, V, and H represent longitude, latitude, and altitude, respectively; x and y are image coordinates; and Numl(), Denl(), Nums(), and Dens() are third-order polynomials composed of rational polynomial parameters [12]. From this equation, the projection points of vertices (or preset vertices) in images from different perspectives can be obtained. Considering the neighborhood correlation of 2-D images, bilinear interpolation is used to gather the features of four adjacent feature blocks for each projection point, as shown in Fig. 2. This feature has an implicit relationship with the corresponding 3-D vertex and can be used for subsequent inference of the coordinates of 3-D vertices. We use a typical VGG-16 architecture to extract perceptual features, and the perceptual feature pool gathers features from the first few layers of VGG-16 (i.e., "conv1_2," "conv2-2," and "conv3_3"), which have high spatial resolution and can retain more detailed information.

Feature concatenation is widely used as a lossless way to combine multiple features, but the final total size of features will vary depending on the number of input images. Some researchers have proposed statistical features for multiview shape recognition [13]. Inspired by this, we have linked the statistical information of each vertex's extracted features in all views as statistical features for subsequent network inference. By concatenating the maximum, average, and standard values of the image features from different views, it ensures that the extracted image feature dimensions are consistent when the number of input views is inconsistent. This enables the network to learn from the feature correlations of multiple views while adapting to the input of multiview images.

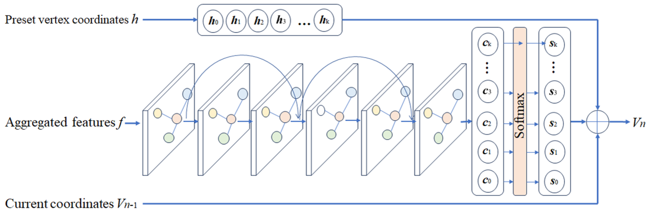


Fig. 3. Overview of proposed deformation inference network.

3) Deformation Inference Network: The next is to construct a point-cloud deformation inference network, which uses the collected statistical features to infer the optimal deformation of each vertex in combination with previous preset vertices.

Because a 3-D model can be described as a collection of point clouds and their connection relationships, which is similar to a graph structure, it can be represented accordingly by $\mathcal{M} = (\mathcal{V}, \mathcal{E})$. Therefore, we consider point clouds and connections as vertices and edges in the graph structure, which can be used to construct point-cloud deformation inference networks using GCNs. For this purpose, we first perform rough surface reconstruction on a point cloud to ensure a connection relationship between the point clouds.

Fig. 3 shows the deformation inference network. It takes the current vertex coordinates, aggregated features, and preset vertex coordinates as network inputs. It comprises six graph convolutional layers with residual connections, followed by a SoftMax layer. The final output is normalized to the weights of 43 vertices, including 42 preset vertices and one current vertex. These weights describe the confidence level that each vertex is a valid coordinate point of the 3-D model. Therefore, the new coordinates obtained by weighting and averaging all vertex coordinates using this weight can be considered as the vertex coordinates inferred by the deformation inference network.

Specifically, we first input the multiview perception feature P into the scoring network. This scoring network consists of six graph convolutional residual networks and ReLU, which is used to predict the scalar weight c_i of each vertex. Then all weights are fed into the SoftMax layer and normalized to the score s_i, where $\sum_{i=1}^{43} s_i = 1$. Then the vertex position is updated to the weighted sum of the original vertex and all preset vertices, i.e., $v = \sum_{i=1}^{43} s_i \times h_i$, where h_i is the position of each vertex.

4) Loss Functions: We defined two types of losses to constrain the deformation process to ensure that vertices move to the correct position. In the following loss functions, p represents the vertex in the prediction mesh and q represents a vertex in the ground-truth mesh. The first is the chamber loss

$$l_c = \sum_p \min_q ||p - q||_2^2 + \sum_q \min_p ||p - q||_2^2. \tag{2}$$

The chamfer distance measures the distance from each predicted point to the actual point, which can evaluate the accuracy of each point's position in the results, thereby ensuring that the predicted point is close to its correct position.

The second is the homonymous epipolar line loss. To ensure that vertices meet the homonymous epipolar line relationship, the homonymous points of multiple sets of remote sensing images aligned with the kernels have the same coordinates in the vertical direction along the epipole. The epipolar line employed here are all oriented along the x-axis, meaning that points with the same name have the same coordinates on the y-axis. Therefore, we have added a loss of the homonymous epipole

$$l_b = \sum_p \left| \frac{\text{Nums}_l(p)}{\text{Dens}_l(p)} - \frac{\text{Nums}_r(p)}{\text{Dens}_r(p)} \right|. \tag{3}$$

The loss describes the difference in y-axis coordinates between homonymous points on different images corresponding to a 3-D point cloud.

Since the loss of the same core line only considers the distance in a single direction, the total loss is defined as the weighted sum of two losses: $L = l_c + \lambda l_b$, where $\lambda = 0.3$.

B. 3-D Mesh Reconstruction Based on Self-Similar Convolution

Although the above-mentioned network makes the extracted 3-D point-cloud location more accurate, there are still two issues in this process.

Issue 1: The above-mentioned network is only used to obtain the positions of the point cloud, which can move freely during deformation. Therefore, the connection relationship between the point clouds is prone to confusion, leading to problems such as mesh self-intersection.

Issue 2: The angle of 2-D remote-sensing images is limited, making it impossible to extract image features of invisible surfaces in remote-sensing images, resulting in inaccurate 3-D structures of hidden surfaces.

We consider using mesh convolution and targeted self-similar convolution to address the above-mentioned two problems. Specifically, first, using MeshCNN [14] makes it more convenient to directly perform the convolution of the mesh model without the format conversion. The MeshCNN model was first proposed for mesh segmentation and classification, and we built a self-similar convolution module based on it to predict mesh offset and achieve mesh optimization. Second, the shape surfaces of artificial targets such as buildings, planes, and ships that we use are usually piecewise smooth, with strong self-similarities at multiple scales. Similar 3-D details often appear repeatedly, and the errors of these targets in the reconstruction process are random and unrelated. Therefore, in mesh optimization, it is essential to retain similar details that may occur repeatedly and to eliminate errors. When the weight-sharing structure of CNN is applied to 2-D images, it can be understood as using the same convolutional kernel to extract similar regions of the image [15]. When we apply this structure to MeshCNN, we can ensure mesh reconstruction of loops and similar structures while suppressing interference from nonloop geometric shapes.

Therefore, we propose a 3-D mesh reconstruction based on self-similar convolution by combining MeshCNN and a weight-sharing structure. By convolving the rough mesh obtained in the previous section, we can predict the mesh offset, move the mesh connection closer to the real point cloud, and avoid mesh

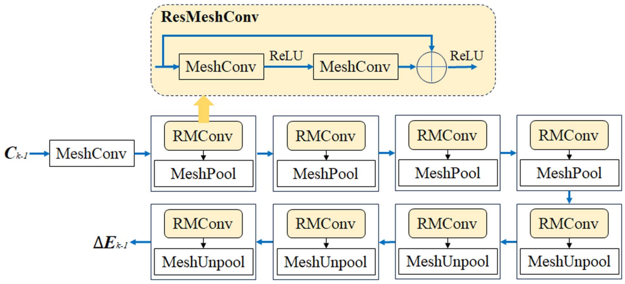


Fig. 4. Overview of the proposed self-similar convolution module.

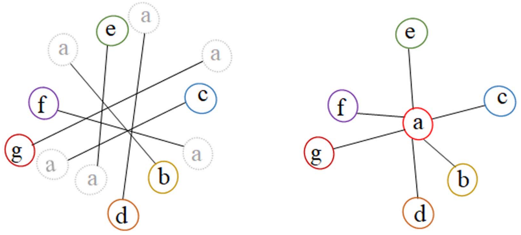


Fig. 5. Edge displacement predicted by the network (left) and the edge average (right).

disorder. Simultaneously, we optimize the hidden surfaces of the 3-D model to have more loops and similar structures.

The mesh reconstruction process is shown in the lower half of Fig. 1. We take the rough mesh M_{k-1} obtained in the previous section as input and extract the feature C_{k-1} of each mesh edge. We randomly initialize the weight W_{k-1}, predict the displacement ΔE_{k-1} of edges through self-similar convolution modules, and obtain the vertex displacement ΔV_{k-1}. This displacement moves M_{k-1} to M_k. Iteration converges because displacement deforms the mesh, shrinking the enveloping point cloud. During iteration, network weights are optimized to minimize losses.

1) Self-Similar Convolution: First, we use the connection relationship between point clouds obtained in the previous section as the initial mesh and extract mesh features. The input mesh feature is a 5-D vector for every edge: the dihedral angle, two inner angles, and, two edge-length ratios for each face [14].

Use a self-similar convolution module to regress and predict the final vertex positions of the mesh. The self-similar convolution module is shown in Fig. 4 and consists of multiple residual mesh convolutions. The self-similar convolution module operates on the mesh feature C_{k-1} extracted from the edge of the input mesh M_{k-1} using a randomly initialized convolution kernel. Due to the self-similar convolution module operating on mesh edges, it regressed a set of mesh edge displacement lists ΔE_{k-1}, which consist of vertex displacement pairs, where three coordinates of two endpoints represent each edge. This will result in the displacement of the same vertex position where multiple edges intersect (as shown on the left side of Fig. 5).

To address this issue, we average each vertex to obtain the final position of the vertices (as shown on the right side of Fig. 5). The average vertex position $\widehat{v}_i$ is calculated using the following formula:

$$\widehat{v}_i = \frac{1}{t}\sum_{j=1}^{t} e_j(v_i). \tag{4}$$

Among them, $e_j(v_i)$ is the vertex position of v_i in edge e_j and t is the number of edges e_j.

Update the mesh M_k using vertex displacement ΔV_{k-1}. Our network does not estimate the absolute vertex position V_k of the deformed mesh but instead estimates the relative displacement ΔV_{k-1} with the input mesh vertex V_{k-1} (i.e., $V_k = V_{k-1} + \Delta V_k$). This is mainly because, in neural networks, predicting residuals produces more favorable results [16], especially for position regression [17].

2) Loss Functions: Finally, calculate the loss function and drive the optimization of the network filter. Considering that chamfer loss can better describe the difference between the predicted results and the actual value, we first define chamfer loss as similar to the previous section

$$l_c = \sum_p \min_q ||p-q||_2^2 + \sum_q \min_p ||p-q||_2^2 \tag{5}$$

where the point cloud obtained from the self-similar convolution optimized mesh M_n is defined as p through differential sampling, and the input point cloud extracted from the input mesh M_0 is defined as q.

Mesh optimization using only chamfer distance may fall into local minima, so we introduce beam gap loss, where the beam gap is the beam from the mesh to the input point cloud. This loss describes the distance between the reconstructed mesh and the input point cloud, which can evaluate the accuracy of the mesh connection. Since the input point cloud is a sparse and discrete representation of a continuous surface, precise beam gap intersections cannot be calculated. Therefore, we propose an approximation method. For point p sampled from mesh M_n, a beam is projected in the direction of the mesh normalization. The intersection of the beam and the point cloud is the closest point in the cylinder with a radius of ε around the beam, denoted as $\text{BeamGap}_\varepsilon(p)$. Therefore, the beam gap loss is defined as

$$l_{bg} = \sum_p ||p - \text{BeamGap}_\varepsilon(p)||_2^2. \tag{6}$$

The total loss is defined as the sum of the two.

IV. Experiments

We conducted our experiments on a PC with an Intel Core i7-10870H CPU, 16 GB RAM, and Nvidia RTX 2080 GPU. We only focused on three types of targets: buildings, planes, and ships. We manually extracted target slices from network inputs and the DSM used as initialization input is obtained through [18]. The experiment evaluated the performance of the method using the earth mover's distance (EMD) [19] and root-mean-square error (RMSE) [20], with the following equations:

$$d_{\text{EMD}}(S_1, S_2) = \min_{\phi: S_1 \to S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2 \tag{7}$$

TABLE I
COMPARISON OF DIFFERENT 3-D RECONSTRUCTION METHODS

Method	3D Model (EMD/RMSE(m))				DSM (RMSE(m))
	Building in Harbin	Building in Yokosuka	Plane	Ship	Building
DSM	0.47/2.36	0.56/3.21	0.84/5.42	0.97/6.45	2.86
3D-R2N2	0.31/1.56	0.48/2.48	0.29/2.07	0.84/3.79	2.24
O-net	0.21/1.46	0.42/2.08	0.27/1.86	0.62/3.37	2.17
LSM	0.29/1.59	0.47/2.53	0.27/1.95	0.69/3.65	2.35
Ours	0.24/1.36	0.29/1.84	0.14/1.69	0.48/2.94	1.79

$$\mathrm{RMSE} = \sqrt{\sum_{i=1}^{N} \left(\Delta X_i^2 + \Delta Y_i^2 + \Delta Z_i^2\right)/N} \tag{8}$$

where N is the number of valid points in the ground truth. $(\Delta X, \Delta Y, \Delta Z)$ is the difference between the test point coordinates and the corresponding nearest neighbor 3-D reconstructed point coordinates. RMSE and EMD respectively describe the target's spatial position and shape accuracy. The satellite data does not include ground control points, so the RMSE used for performance evaluation reflects relative accuracy.

The experimental dataset consists of two groups. The SSIs of dataset A come from SuperView-1 with a spatial resolution of 0.5 m, including four regions in Harbin, China; Japan; the United States; and Qatar. The ground-truth value of the 3-D model was obtained from unmanned aerial vehicle tilt photogrammetry, Google Earth 3-D, Internet datasets, and other sources. Dataset B is a public dataset from the 2019 IGRASS Data Fusion Contest and contains World-View satellite data with a resolution of 0.35 m and DSM ground-truth values obtained by LiDAR.

A. Comparison to State-of-the-Art Methods

To fully demonstrate the effectiveness of the proposed method, we selected some recent methods similar to our method for comparison experiments, including 3-D-R2N2, O-net, and LSM. We set the parameters of all comparison methods according to the suggestions in their original papers.

1) Dataset A: The accuracy of dataset A is shown in the 3-D model column of Table I and some experimental results are shown in Fig. 6. The color bar on the right side of the picture represents the altitude, measured in meters. The first two lines in Fig. 6 show the experimental results for buildings in the Harbin area of China. Each method uses target 3-D models and satellite stereo pairs obtained at different times in the same area as the test data as training samples. By comparing with the true value, we see that, although various methods can restore the 3-D structure, the proposed method produces a 3-D model closer to the true value in visual effect and smoother at the roof of the building.

In practical applications, it is difficult to directly obtain 3-D training samples of the target area. Therefore, we used the aforementioned data from Harbin, China, as training samples and conducted tests on samples from a certain region in Japan. The experimental results are shown in lines 3–5 of Fig. 6. From the experimental results, we see that, for buildings with simple structures (as shown in line 3 of Fig. 6), most methods can still restore the overall model of the target without corresponding regional training samples. However, for more complex buildings (such as lines 4 and 5 of Fig. 6), it is not possible to achieve good restoration, especially for the attic at the top of the building. This is because most current computer-vision reconstruction methods rely on a large number of similar training samples and use networks to construct the corresponding relationship between images and targets. Therefore, it is difficult to restore the 3-D structure of the target when there is a lack of training samples in the test area. Nevertheless, our reconstruction method can avoid dependence on the base training data of the test area and achieve good reconstruction results, even if there are differences between the training and testing data (as shown in the attic on the top of the building in lines 4 and 5 of Fig. 6). This is because, compared to other methods of establishing a one-to-one correspondence between images and targets, our proposed reconstruction network learns the direction and position of each vertex's movement while utilizing the self-similarity of the target itself to restore the hidden surface. Since most artificial buildings have regular geometric shapes, usually consisting of right-angled rectangles and exhibit a combination of multiple regular geometric shapes [21], the proposed method can be applied to various common artificial buildings.

The experimental results in line 6 of Fig. 6 show the reconstruction results for the aircraft target. We see that the proposed method can accurately restore the structure of the aircraft, including the extension angle and size of the wings. Meanwhile, compared to the other methods, the shape of the aircraft is more regular without significant bending. Line 7 in Fig. 6 shows the results of ship target reconstruction. Owing to the limited spatial resolution of remote-sensing images and the complexity of the ship, the details of the upper-layer plywood reconstructed by each method cannot be clearly identified. Nevertheless, compared to other methods, the proposed method can accurately highlight the position of the ship tower, which provides support for the application of 3-D models in subsequent recognition, segmentation, and other fields.

2) Dataset B: Considering the lack of a one-to-one correspondence between satellite data and the target 3-D model in the existing public dataset, we used a stereo-matching public dataset containing the target DSM ground truth to verify performance for the elevation dimension of the target reconstruction. We reprojected the reconstructed 3-D model as a DSM to compare with the true value, and the results are shown in Fig. 7. The color bar on the right side of the picture represents the altitude, measured in meters. The accuracy of the target reconstructed elevation dimension is verified using the RMSE metric (which contains only the z dimension), as shown in the DSM column of Table I.

From the results of the above figures and tables, when compared with other methods, our method yields more complete 3-D model results for aircraft, ships, buildings, and other targets; produces accurate spatial positions and shapes; scales consistently with real targets; and better restores the geometric structure of the target itself, while achieving high accuracy in different indicators.

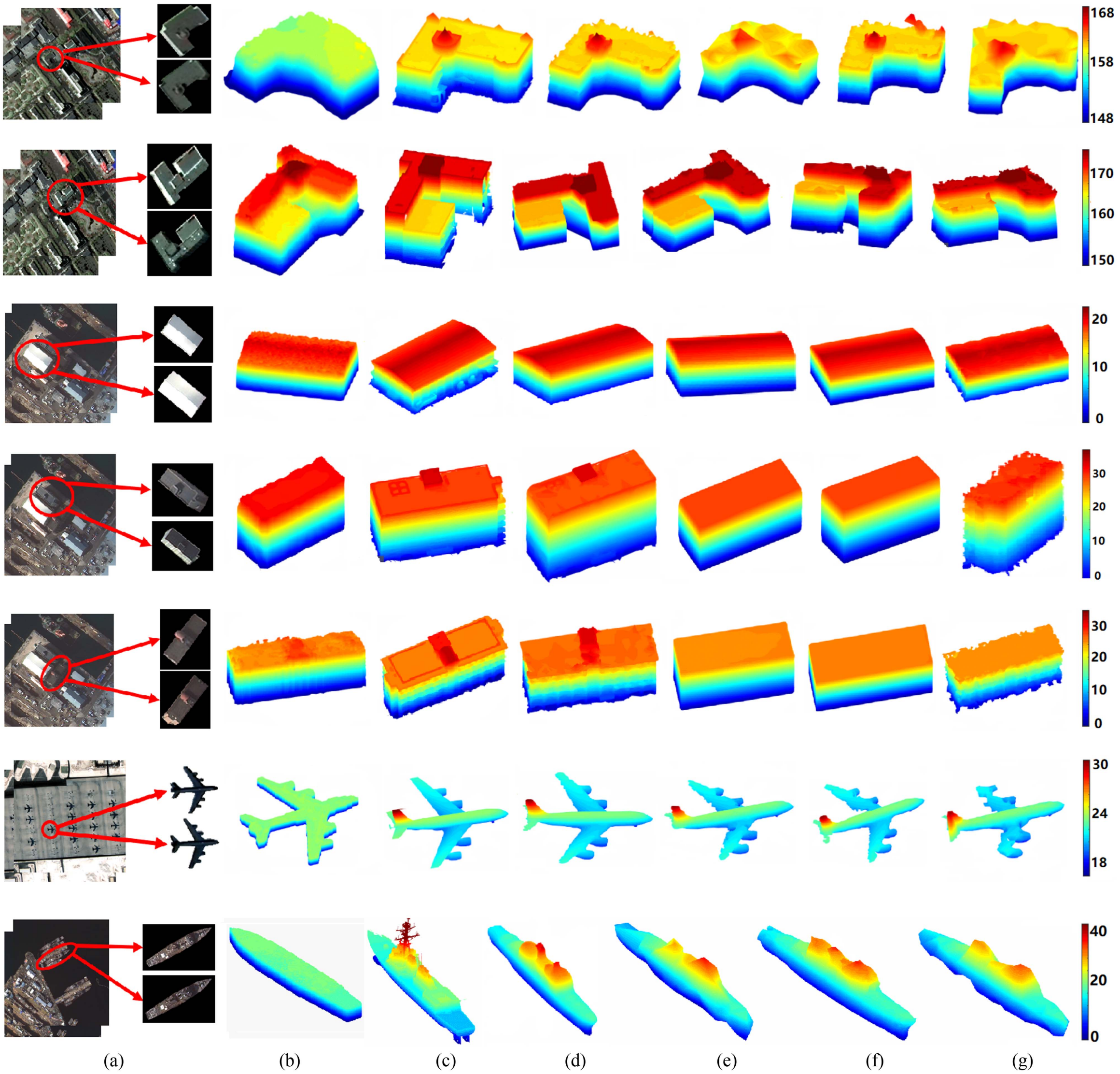


Fig. 6. Three-dimensional model results of our method and other methods. (a) Stereo pair. (b) DSM. (c) Ground truth. (d) Proposed method. (e) 3-D-R2N2. (f) O-net. (g) LSM.

B. Ablation Study

Considering that our proposed method involves multiple vital steps and each step's adjustment will impact performance, we conducted ablation experiments to compare the effects of different structures or parameters on the method. We used building targets from dataset A.

1) Different Losses: We evaluated the method using different losses to test the effectiveness of the loss function. Chamfer loss l_c can comprehensively describe the reconstruction effect and, when missing, it causes the network to be unable to converge. Therefore, chamfer loss will be included in both steps of the experiment.

By comparing the accuracy of Table II, we see that we can improve the network performance by addressing the increased homonymous epipolar loss caused by deformation inference and the increased beam gap loss caused by mesh reconstruction.

TABLE II
COMPARISON OF RESULTS WITH DIFFERENT LOSSES

Deformation inference		Mesh reconstruction		EMD	RMSE
l_c	l_b	l_c	l_{bg}		
✓		✓		0.49	1.97
✓	✓	✓		0.41	1.74
✓		✓	✓	0.37	1.79
✓	✓	✓	✓	0.27	1.57

2) Different Iterations: The two steps can be iterated to achieve higher accuracy. We compare the impact of different iterations of the two steps on reconstruction accuracy. When the number of iterations in one link changes, the number of iterations in the other link is fixed.

From Table III, the performance of our model continues to improve as the number of iterations increases. Among them, deformation inference is approximately saturated at three times, while mesh reconstruction requires approximately five times.

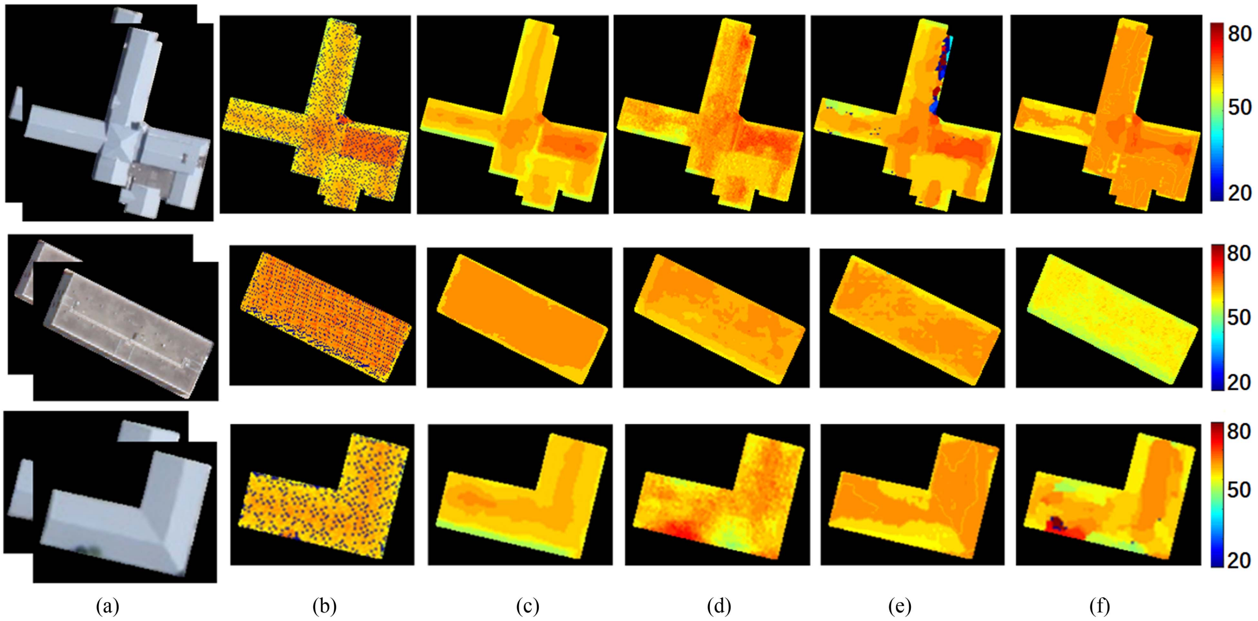


Fig. 7. DSM results from the 3-D model projection of our method and other methods. (a) Stereo pair. (b) Ground truth. (c) Proposed method. (d) 3-D-R2N2. (e) O-net. (f) LSM.

TABLE III
COMPARISON OF RESULTS WITH DIFFERENT ITERATIONS

Iterations	1	2	3	4	5	6
Deformation inference (EMD)	0.42	0.39	0.27	0.27	0.28	0.27
Deformation inference (RMSE)	2.57	1.96	1.57	1.59	1.55	1.58
Mesh reconstruction (EMD)	0.37	0.35	0.33	0.29	0.27	0.28
Mesh reconstruction (RMSE)	1.85	1.81	1.81	1.75	1.57	1.58

V. Conclusion

In this article, we proposed a 3-D model extraction method for SSIs based on self-similar convolution and deformation inference networks constrained by an RFM. Through the proposed two steps, the target model extracted from a digital surface model was subjected to point-cloud deformation and mesh reconstruction to achieve progressive optimization of the 3-D model. Experiments on multiple datasets have shown that our method can obtain more accurate 3-D models in terms of spatial position and shape compared to other methods and can also achieve better accuracy for targets that differ from the training samples.

References

[1] Dawen Yu, S. Ji, S. Wei, and K. Khoshelham, "3D building instance extraction from high-resolution remote sensing images and DSM with an end-to-end deep neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 4406019.

[2] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4460–4470.

[3] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J.-B. Huang, "Deepmvs: Learning multi-view stereopsis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2821–2830.

[4] S. Tripodi et al., "Operational pipeline for large-scale 3D reconstruction of buildings from satellite images," in *Proc. IGARSS 2020-2020 IEEE Int. Geosci. Remote Sens. Symp.*, 2020, pp. 445–448.

[5] Y. Wang, S. Zorzi, and K. Bittner, "Machine-learned 3d building vectorization from satellite imagery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1072–1081.

[6] Y. Qian, S Ramalingam, and J. H. Elder, "LS3D: Single-view gestalt 3D surface reconstruction from Manhattan line segments," in *Proc. 14th Asian Conf. Comput. Vis. Comput. Vis.*, 2019, pp. 399–416.

[7] C. Son and P. Soon-Yong, "3D map reconstruction from single satellite image using a deep monocular depth network," in *Proc. 13th Int. Conf. Ubiquitous Future Netw.*, 2022, pp. 5–7.

[8] C. B. Choy et al., "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," in *Proc. 14th Eur. Conf. Comput. Vis.*, Proceedings, Part VIII 14, 2016, pp. 628–644.

[9] A. Kar, C. Häne, and J. Malik, "Learning a multi-view stereo machine," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 365–376.

[10] J. Y. Gwak, C. B. Choy, M. Chandraker, A. Garg, and S. Savarese, "Weakly supervised 3d reconstruction with adversarial constraint," in *Proc. Int. Conf. 3D Vis.*, 2017, pp. 263–272.

[11] D. C. Chapell et al., "NeRF-based 3D reconstruction and orthographic novel view synthesis experiments using City-scale aerial images," in *Proc. IEEE Appl. Imagery Pattern Recognit. Workshop*, 2023, pp. 1–7.

[12] S. Yavari, M. J. V. Zoej, A. Mohammadzadeh, and M. Mokhtarzade, "Particle swarm optimization of RFM for georeferencing of satellite images," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 1, pp. 135–139, Jan. 2012.

[13] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 945–953.

[14] R. Hanocka et al., "Meshcnn: A network with an edge," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–12, 2019.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[16] O. Shamir, "Are resnets provably better than linear predictors," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 507–516.

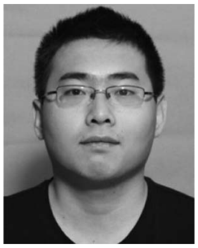
[17] R. Hanocka et al., "Alignet: Partial-shape agnostic alignment via unsupervised learning," *ACM Trans. Graph.*, vol. 38, no. 1, pp. 1–14, 2018.

[18] S. Yang, H. Chen, and W. Chen, "Generalized stereo matching method based on iterative optimization of hierarchical graph structure consistency cost for urban 3D reconstruction," *Remote Sens.*, vol. 15, no. 9, 2023, Art. no. 2369.

[19] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, 2000, Art. no. 99.

[20] Z. Hu, T. Li, Y. Yang, X. Liu, H. Zheng, and D. Liang, "Super-resolution PET image reconstruction with sparse representation," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf.*, 2017, pp. 1–3.

[21] L. Ding, H. Tang, Y. Liu, Y. Shi, X. X. Zhu, and L. Bruzzone, "Adversarial shape learning for building extraction in VHR remote sensing images," *IEEE Trans. Image Process.*, vol. 31, pp. 678–690, 2021.



Wen Chen (Student Member, IEEE) received the B.S. degree in information and communication engineering from the Harbin Institute of Technology at Weihai, Weihai, China, in 2016, and the M.S. degree in information and communication engineering in 2019 from the Harbin Institute of Technology, Harbin, China, where he is currently working toward the Ph.D. degree in information and communication engineering from the School of Electronics and Information Engineering.

His research interests include 3-D reconstruction and object detection.

Hao Chen (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in information and communication engineering from Harbin Institute of Technology, Harbin, China, in 2001, 2003, and 2008, respectively.

Since 2004, he has been with the School of Electronics and Information Engineering, Harbin Institute of Technology, where he is currently a Professor. His main research interests include remote sensing image processing and image compression.

Shuting Yang received the M.Sc. degree in electrical circuits and systems from Jilin University, Changchun, China, in 2021. She is currently working toward the Ph.D. degree in information and communication engineering with Harbin Institute of Technology, Harbin, China.

She is currently studying with the School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin, China. Her research interests include remote sensing data processing and deep learning algorithm research.