remote sensing

Article

Textured Mesh Generation Using Multi-View and Multi-Source Supervision and Generative Adversarial Networks

Mingyun Wen , Jisun Park and Kyungeun Cho *

Department of Multimedia Engineering, Dongguk University-Seoul, 30, Pildong-ro 1-gil, Jung-gu,
Seoul 04620, Korea; wmy_dongguk@dongguk.edu (M.W.); jisun@dongguk.edu (J.P.)
* Correspondence: cke@dongguk.edu; Tel.: +82-02-2260-3834

Abstract: This study focuses on reconstructing accurate meshes with high-resolution textures from single images. The reconstruction process involves two networks: a mesh-reconstruction network and a texture-reconstruction network. The mesh-reconstruction network estimates a deformation map, which is used to deform a template mesh to the shape of the target object in the input image, and a low-resolution texture. We propose reconstructing a mesh with a high-resolution texture by enhancing the low-resolution texture through use of the super-resolution method. The architecture of the texture-reconstruction network is like that of a generative adversarial network comprising a generator and a discriminator. During the training of the texture-reconstruction network, the discriminator must focus on learning high-quality texture predictions and to ignore the difference between the generated mesh and the actual mesh. To achieve this objective, we used meshes reconstructed using the mesh-reconstruction network and textures generated through inverse rendering to generate pseudo-ground-truth images. We conducted experiments using the 3D-Future dataset, and the results prove that our proposed approach can be used to generate improved three-dimensional (3D) textured meshes compared to existing methods, both quantitatively and qualitatively. Additionally, through our proposed approach, the texture of the output image is significantly improved.

Keywords: single image textured mesh reconstruction; convolutional neural networks; generative adversarial network; super-resolution

check for updates

Citation: Wen, M.; Park, J.; Cho, K. Textured Mesh Generation Using Multi-View and Multi-Source Supervision and Generative Adversarial Networks. *Remote Sens.* **2021**, *13*, 4254. https://doi.org/10.3390/rs13214254

Academic Editor: Danfeng Hong

Received: 27 August 2021
Accepted: 20 October 2021
Published: 22 October 2021

1. Introduction

The generation of three-dimensional (3D) textured meshes from image inputs is a topic of significant interest. This is because of the widespread use of this approach in various applications, such as virtual reality (VR)-based and augmented reality (AR)-based games and platforms, which significantly rely on the rendering of 3D textured meshes. Extensive studies on the generation and reconstruction of 3D textured meshes from single images have been conducted, and significant progress has been made. However, most studies ignore the reconstruction of the corresponding texture of the generated mesh, which is essential for perceiving and understanding a 3D mesh. For example, without the corresponding textures, it is difficult to distinguish a 3D mesh of a horse from that of a zebra by only looking at their meshes. As it pertains to generated 3D meshes, the effective reconstruction of the corresponding textures remains a challenging task. This problem should be addressed to ensure the improved application of reconstructed meshes in practical contexts.

In some studies, such as those conducted by Sun et al. and Tulsiani et al. [1,2], the researchers proposed representing 3D textured meshes using voxels, where the voxels comprise the surface information and color information of the input image. Owing to the cubical increase in computational cost, as it pertains to increased voxels, voxel-based representations cannot be used to generate high-resolution 3D textured meshes. An implicit function-based approach for generating and representing textured meshes [3,4] has been proposed to address the problem of computational cost, as it pertains to increased voxels, and currently, it is a hot research topic in the field of 3D textured mesh generation. The implicit function-based method for reconstructing textured meshes has a limitation in that it cannot be directly used to obtain renderable textured meshes. The effective generation of textured meshes through this approach requires intensive computation. Additionally, representation methods involving voxels and implicit functions cannot be used to generate images with dense textures because the textures of such images are generated based on the prediction of the color values of points or voxels. In some studies, such as those conducted by Alldieck et al. [5] and Novotny et al. [6], the authors successfully generated textured meshes by deforming template meshes using UV maps that define the way in which textures are mapped to the template meshes. The limitation of the approach presented above is that it cannot be applied to the representation of objects that are not homeomorphic to those in the template meshes. However, this method is advantageous in that it can be used to directly generate renderable 3D textured meshes that are compatible with the related 3D-image rendering software.

Our study is similar to that conducted by Pavallo et al. [7] and Kanazawa et al. [8] in which a UV map is generated in advance by parameterizing the template mesh, which determines the way in which the texture of an image is mapped to the template mesh, after which a deformation map is used to deform the template mesh. Therefore, the problem of reconstructing 3D textured meshes can be addressed in the two-dimensional (2D) image space. The novelty of our methods concerns two aspects compared to the methods in [7,8]. First, Pavallo et al. and Kanazawa et al. used single-view supervision of the input and mask images, and manually annotated 3D key points. Contrarily, our proposed mesh-reconstruction network adopts a multi-view RGB, depth, and normal loss during the training process. The multi-view depth and normal images can provide more detailed 3D information to generate a deformation map for deforming the template mesh. Second, inspired by the super-resolution generative adversarial network (GAN) which can recover high-quality images from blurred 2D input, our texture-reconstruction network aims to recover the high-texture image from low-resolution images incorporating the 2D input and deformation map. The rendered 2D image is utilized for supervision, which enables our network to leverage the pre-trained networks to calculate perceptual loss, thereby enhancing the quality of the generated textures. Pavllo et al. also used a GAN to generate meshes and high-quality textures [7]. However, the deformation map and pseudo-ground-truth texture are utilized directly for supervision, which cannot utilize pre-trained network on other image datasets for loss calculation. In recent years, multimodal deep learning has been studied and achieved good results [9]; it incorporates data captured by different kind of devices, like CT and Lidar [10], to reconstruct accurate mesh. In our methods, even we utilized multi-source information (2D image, depth, and normal images), but our method is quite different from the multimodal methods. This is because the depth and normal images are only used for supervision during training, and they are not utilized during inference time. Therefore, our method has less requirements for input.

The contributions of this study are as follows:

- We propose a mesh-reconstruction method based on multi-view and multi-source supervision. We leverage the differentiable rendering technique to render multi-view RGB, depth, and normal images as the ground truth. This approach enables the network to achieve increasingly robust supervision during the training process, and thus, the network can be used to generate a highly accurate deformation map for deforming the template mesh.
- We propose a high-resolution texture-reconstruction method, which relies on a super-resolution-based GAN. We also propose the use of the super-resolution method to enhance low-resolution textures, obtained through our proposed mesh-reconstruction network. Specifically, to ensure that the network determines the correlation among the texture, the deformation map, and the input RGB image, we stacked the low-resolution texture and the deformation map obtained using the mesh-reconstruction network,

as well as the input image, channel-wise, as the input of the texture-reconstruction network. Thus, the network enhances the texture by considering the global texture information obtained from the low-resolution texture, the mesh topology obtained from the mesh deformation map, and the high-frequency texture information obtained from the input RGB image.

The remainder of this paper is organized as follows: In Section 2, we review recent studies on the generation and reconstruction of textured meshes using single-view images as the input. In Section 3, we describe our proposed methods. In Section 4, we present the experimental results and their evaluation. In Section 5, we discuss the results, and we provide directions for future research. We present the conclusions in Section 6.

2. Related Works

In this section, we analyze the existing studies on textured mesh generation using a single image. The related studies are categorized into the following two groups: mesh reconstruction and texture reconstruction.

2.1. Mesh Reconstruction

Various approaches, such as mesh deformation [11–15] and voxel-based representation [3,16,17], have been proposed for the generation of 3D meshes from 2D images. Zhu et al. generated a voxel-based mesh using an image feature extractor and a GAN [16]. However, the generated 3D model comprised relatively large voxel sizes, thereby decreasing its accuracy.

Extensive studies on the direct reconstruction of meshes have been conducted to enhance the detailed accuracy of output mesh surfaces. Pontes et al. used a multi-label classifier to select a 3D model that was very similar to the object in the input image, after which they fitted the selected 3D model to the object in the input image using free-form deformation (FFD) [11]. This approach requires a 3D model repository comprising multiple 3D models in the same category. Wu et al. predicted the normal, depth, and silhouette images from a 2D image, after which they reconstructed a 3D mesh of the image [12]. Because the network was trained through the single-view supervision approach, the prediction of invisible parts could fail. Similarly, Kato et al. trained a mesh-reconstruction network through the single-view supervision of silhouettes [13]. However, because silhouettes only provide the outline information of an object, they are insufficient as supervision approaches for the accurate prediction of detailed meshes.

Groueix et al. [14] conducted a study on the deformation of meshes by combining multiple patches. The generated patches were combined, after which they were reconstructed to form a mesh. The loss was calculated using the Chamfer distance between the sampled 3D points from a ground-truth mesh and a reconstructed mesh. For the generated mesh, self-intersection can occur because the generated mesh is not watertight. The approaches used in the studies mentioned above achieved lower accuracy when the template meshes and topologies were different. Pan at el. attempted to improve the precision of generated 3D meshes by proposing a network that can change the mesh topology [15]. By learning the reconstruction error, the error-prediction-based network can determine the faces that should be excluded from the generated template meshes.

Voxel-based methods require significant computing resources, and they have complex structures. However, such methods have relatively few restrictions, as they pertain to generating the mesh topologies of target objects. In some studies, such as those conducted by Mescheder et al. and Xu et al. [3,17], the researchers aimed to re-construct smooth surfaces using implicit functions. In one network, the surface of an object is predicted through the calculation of the signed-distance functions (SDFs) after predicting the location in which a 3D point is projected onto a 2D image [17]. In another network with a similar structure, the boundary of an object is predicted by calculating the occupancy function [3]. Voxel-based methods are used to obtain the surface function values of a 3D grid, after which they are used to obtain a 3D mesh using the Marching cubes algorithm [18]. However, because meshes reconstructed through voxel-based approaches have gently expressed edges, they limit the enhancement and accuracy of generated meshes.

Therefore, the meshes reconstructed using the techniques mentioned above tend to demonstrate low accuracy. As a result, in this study, we propose a mesh generator that exploits the predicted RGB, depth, and normal images as a supervision approach for ensuring effective improvement in the precision of the reconstructed meshes.

2.2. Texture Reconstruction

The field of deep learning, methods for generating 3D textured meshes, depending on the form of texture, can be categorized as follows: voxel-based methods [1,2], implicit function-based methods [3,19], template-based methods [7,8,20–23], and point-cloud-based methods [2,24–26].

In voxel-based methods, meshes and textures are represented using voxels, where the voxels comprise the occupancy values that indicate the intersections between voxels and target meshes as well as the color values that define the colors of the target meshes. This approach usually requires networks that have 3D structures. Sun et al. used three decoders to generate color, blending weights, and a 3D-to-2D flow field. The 3D-to-2D flow field is used to sample colors from input images, after which the sampled and predicted colors are blended using the predicted blending weights [1]. Owing to the computational and memory restrictions, the resolutions of the reconstructed textures are limited to low resolutions.

Implicit function-based methods address the problem of memory and computational restrictions [4,19]. Such methods do not require discretization during the training process. Additionally, they do not have restrictive requirements during the generation of meshes, and they can be integrated with any mesh-reconstruction network to achieve end-to-end training. However, the inference process requires the use of voxels. Additionally, post-processing methods for generating textured meshes, such as the Marching cubes algorithm [18], are required.

In some studies, researchers generated colored 3D point clouds from single images [2,24–26]. Novel views, sometimes with depth, are usually estimated to generate colored point clouds. Tulsiani et al. generated a colored point cloud by estimating the virtual views and depth images [2]. However, colored point clouds do not show the structural information and textures of generated meshes. They require significant post-processing to generate textured meshes. Additionally, the reconstruction quality is usually poorer than that of the other methods.

All the methods mentioned above require additional post-processing to generate a standard textured mesh that can be applied in popular 3D entertainment creation tools, such as Unity3D. Template-based methods [7,8,20–23] solve this problem by incorporating a fixed UV map shared by the texture and deformation maps. Therefore, they can be used to reconstruct textured meshes through popular convolutional neural networks, which are lightweight and computationally efficient compared to voxel-based methods, which require a 3D network, for example, a 3D convolutional network. Such methods have demonstrated satisfactory performance in the generation of the textured meshes of objects that are homeomorphic to their corresponding template meshes. Essentially, template-based methods are used to train reconstruction networks using the reconstruction loss of reconstructed images, which are rendered using differentiable renderers [20,27] and original input images. Some studies, such as those conducted by Kanazawa et al. and Chen et al. [8,20], adopt silhouette-based reconstruction loss. Chen et al. trained their proposed network by calculating the silhouette and rendered RGB image losses [20]. Kanazawa et al. [8] added a 3D keypoint loss in their experiments on the Caltech-UCSB Birds (CUB) dataset.

This study is significantly like that conducted by Pavllo et al. [7]. However, contrary to their works, we adopt multi-view and multi-source supervision for loss calculation to ensure enhanced shape estimation. Additionally, our proposed texture-reconstruction network is based on a super-resolution method that incorporates mesh deformation maps and the appearances of objects in input images to enhance the quality of textures obtained from low-resolution images. The loss is calculated using the rendered images instead of the textured images. Therefore, the perceptual loss of the features extracted through the VGG-19 network [28] pretrained using the ImageNet dataset [29] can be utilized.

3. Proposed Methods for Generating Textured Meshes

Given a 2D image as the input, our proposed method generates a corresponding 3D textured mesh in two steps. First, a mesh-reconstruction network is used to predict a deformation map, which then deforms the existing template mesh into a target shape and a low-resolution texture image. The input, deformation map, and low-resolution texture image are then input into the texture-reconstruction network to generate a high-resolution textured mesh.

Figure 1 shows the workflow of the proposed method for reconstructing 3D textured meshes. After obtaining the deformation map, the target shape is generated using a deformation map for deforming a template mesh in the deformation module. In this study, we used a UV sphere as the template mesh. A textured mesh can be generated using a UV mapping module in which the texture is mapped to the 3D mesh using a pre-defined UV map.

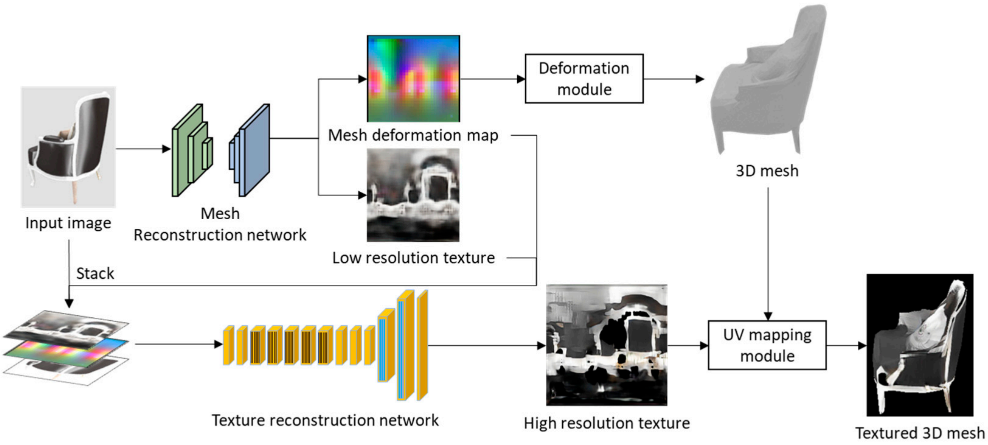

Figure 1. Textured mesh reconstruction workflow.

3.1. Mesh Reconstruction Network Based on Multi-View and Multi-Source Supervision

Given a single RGB image with a masked target object, an encoder is used to extract the object features and generate a feature map, after which a decoder is used to convert the feature map to a deformation map and a low-resolution texture. The low-resolution texture and deformation map can be mapped to the template mesh using a predefined UV map. Therefore, the problem of generating 3D textured meshes can be solved in 2D space. Figure 2 shows the training process of the mesh-reconstruction network.

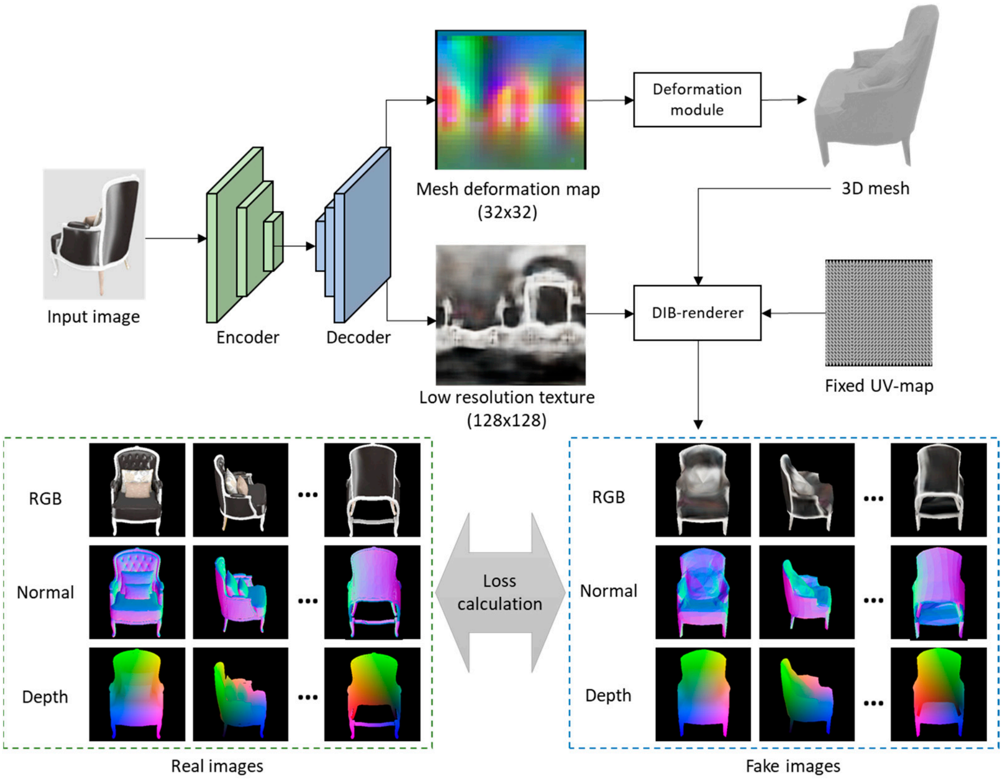

Figure 2. Training process of mesh-reconstruction network.

The training process of the mesh-reconstruction network is typically based on calculating the reconstruction loss between the rendered images obtained from the predicted meshes and the ground-truth meshes. In this study, we propose calculating the loss using multi-view and multi-source supervision. In other words, we render different angles of textured meshes. We then use the RGB images rendered from the textured meshes, the depth images rendered based on the distance from the surface to the center of the meshes, and the normal images rendered based on the normal direction of the meshes. Therefore, our proposed approach allows for robust supervision during the training of the network.

We used the differentiable renderer (DIB-Renderer) [20], which is popular in deep learning-based 3D mesh reconstruction tools for rendering images. We used the vertex shader rendering function of the DIB-renderer to render the depth map and the normal map. To obtain the depth of each vertex, we set the origin as the center of the mesh and calculated the distance of each vertex to the origin of the mesh. The depths of each axis of a vertex represents as the R, G, and B values. This representation is advantageous in that, even when we render meshes using different camera angles, the color values of the same vertexes remain unchanged, thereby implicitly encouraging the network to generate semantically aligned shapes across different input images. In other words, for similar objects, such as chairs, the legs are represented using similar areas in the deformation map. The objective function is calculated using the mean squared error, as depicted in Equation (1). y_t, y_d, and y_n represent the rendered textured image, depth image, and normal image obtained using the original 3D model. The $\hat{y}_t$, $\hat{y}_d$, and $\hat{y}_n$ values are rendered from the predicted 3D textured mesh, as follows:

$$L(y,\ \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}[(y_t - \hat{y}_t) + (y_d - \hat{y}_d) + (y_n - \hat{y}_n)] \quad (1)$$

3.2. Texture Reconstruction Network Based on Super-Resolution

Texture reconstruction relies on a super-resolution method to generate high-resolution textured images. Given an input image, the mesh reconstruction network first generates a deformation map and a low-resolution texture image. Afterwards, the deformation map, the input image, and the low-resolution texture image are scaled to the same resolution and stacked channel-wise as the input of the texture-reconstruction network. The texture-reconstruction network outputs a high-resolution texture image. Figure 3 shows the process of texture reconstruction and the loss calculation method used during the training process.

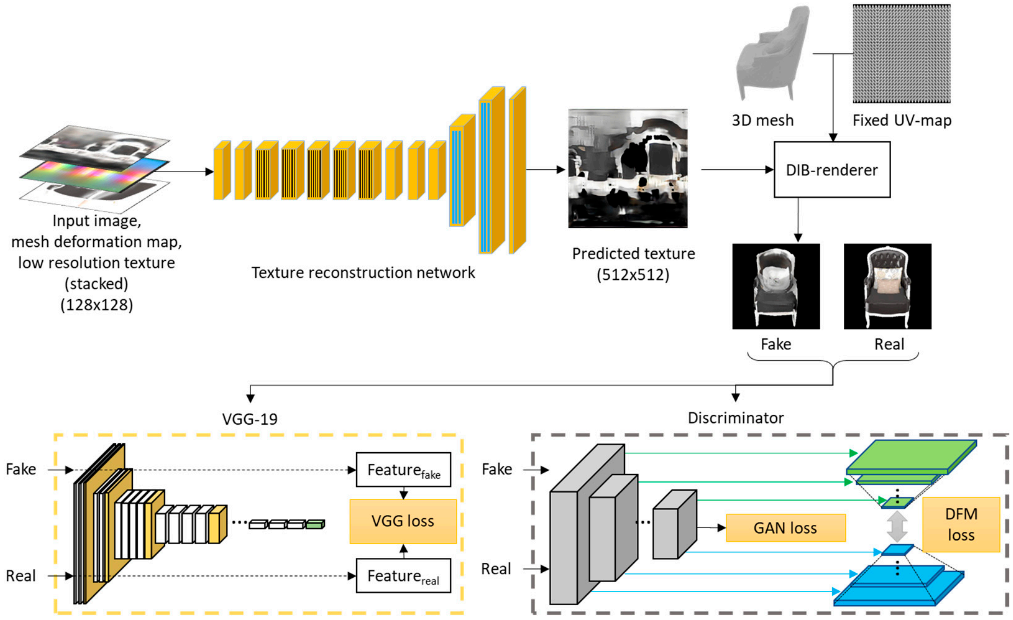

Figure 3. Training process of texture-reconstruction network.

To generate textures with high-frequency details, we adopt a GAN, which has demonstrated significant success in the generation of realistic images in the image-generation area. To train the generator of the GAN, we adopt three losses: the VGG-19 loss, which is denoted as L_{vgg}, the discriminator feature matching loss [30], which is denoted as L_{DFM} , and the GAN loss, which is denoted as L_{GAN}. The total training loss of the generator is described using Equation (2), as follows:

$$L = L_{vgg} + L_{GAN} + L_{DFM} \tag{2}$$

The VGG-19 loss was first proposed by Johnson et al. and Dosovitskiy et al. [31,32], and it is used to calculate the loss on feature spaces using the VGG-19 network, which is pretrained using the ImageNet dataset, and its use has demonstrated significant improvements in determining perceptual quality of generated images [31]. The features of the 2nd to the 5th layers were extracted, and the L2 distance was applied in the calculation of the VGG-19 loss. The loss is depicted in Equation (3), where i represents the index of the layer. $\hat{I}$ and I denote the image generated from the predicted texture and the predicted mesh, as well as the texture generated using the inverse renderer proposed by Pavllo et al. [7] and the predicted mesh, as follows:

$$L_{vgg} = \sum_{i=2}^{5} \|f^{i}_{vgg}(\hat{I}) - f^{i}_{vgg}(I)\|^{2}_{2} \tag{3}$$

A discriminator feature matching loss (DFM) is also incorporated to enable the generator to generate natural images, as shown in Equation (4). Features are extracted from multiple layers of the discriminator, f_D, after which they are calculated using the L1 distance. T represents the total number of layers, and M represents the elements of each layer, as follows:

$$L_{DFM} = \sum_{i=1}^{T} \frac{1}{M} \| f_D^i(\hat{I}) - f_D^i(I) \|_1 \tag{4}$$

Let x, and x_d represent the input and deformation images generated using the mesh-reconstruction network, respectively. The GAN loss can be depicted as shown in Equation (5), where f_G represents the generator of the GAN, respectively. as follows:

$$L_{GAN} = -log(f_D(f_G(x,\ x_d))) \tag{5}$$

4. Experimental Results and Evaluation

In this section, we present the experimental process and the achieved results in detail, after which we evaluate the reconstruction results for textured meshes. All the experiments were conducted on the 3D-Future dataset [33].

4.1. Dataset Processing for the Training and Evaluation of the Mesh-Reconstruction Network

The 3D-Future dataset comprises 9992 3D models with corresponding high-resolution textures. To generate the training and evaluation datasets for the textured mesh reconstruction task, we generated input and ground-truth images. To generate the input images, we randomly generated 25 views of each 3D model with textures using Blender [34]. The resolutions of the rendered images were 256 × 256 pixels. The training and testing set were divided following the method mentioned in [33].

In both the mesh-reconstruction and texture-reconstruction tasks, we leveraged the DIB-Renderer to render images of the reconstructed mesh. The rendered images were used to calculate the loss. To avoid the discrepancies resulting from the use of different renderers, the ground-truth images were rendered using the DIB-Renderer. The textured images, depth images, and normal images of the eight views for each 3D model were rendered using both the ground-truth and estimated meshes in the mesh-reconstruction task. In the texture reconstruction task, only textured images were rendered to calculate the loss.

4.2. Experimental Results of Mesh Reconstruction

To verify that our proposed mesh-reconstruction method enhances performance compared to the existing methods, we compared the generation results to those of OccNet and Convmesh. Our mesh reconstruction method is similar to the methods proposed in [7,8], especially the one in [7]. Our mesh reconstruction method and [7] both adopts DIB-Renderer and use UV sphere as a template mesh for deforming into target shape, while [8] adopts Neural Mesh Renderer [35] and ICO-sphere. The proposed method of [7] is based on [8]. Therefore, we only compared with [7] because it is newer and more relevant to our mesh reconstruction method. Additionally, we trained a model using single-view and multi-source (SVMS) supervision, which was used to verify the effectiveness of multi-view supervision. The qualitative comparison results are shown in Figure 4. To replicate the results of OccNet, we used the pretrained model in Fu et al. [33], where the method for generating the dataset was similar to that employed in this study. The results are shown in Figure 4b. The model was trained using 3D supervision by comparing randomly sampled 3D points on the reconstructed and ground-truth meshes. Convmesh adopts a network structure that is like our proposed structure, and it is trained using the single-view supervision of the rendered RGB image, as described in the works of Pavllo et al. [7]. Therefore, our mesh reconstruction method can be treated as an enhanced version of it. The results are presented in Figure 4c. Figure 4d shows the results of our proposed methods, which were trained using single-view and multi-source supervision of rendered RGB images, depth images, and normal images. Figure 4e shows the reconstruction results obtained from our model, which was trained using multi-view and multi-source supervision. The resolution of the ground-truth image was 128 × 128 pixels for our proposed models and 256 × 256 pixels for Convmesh.

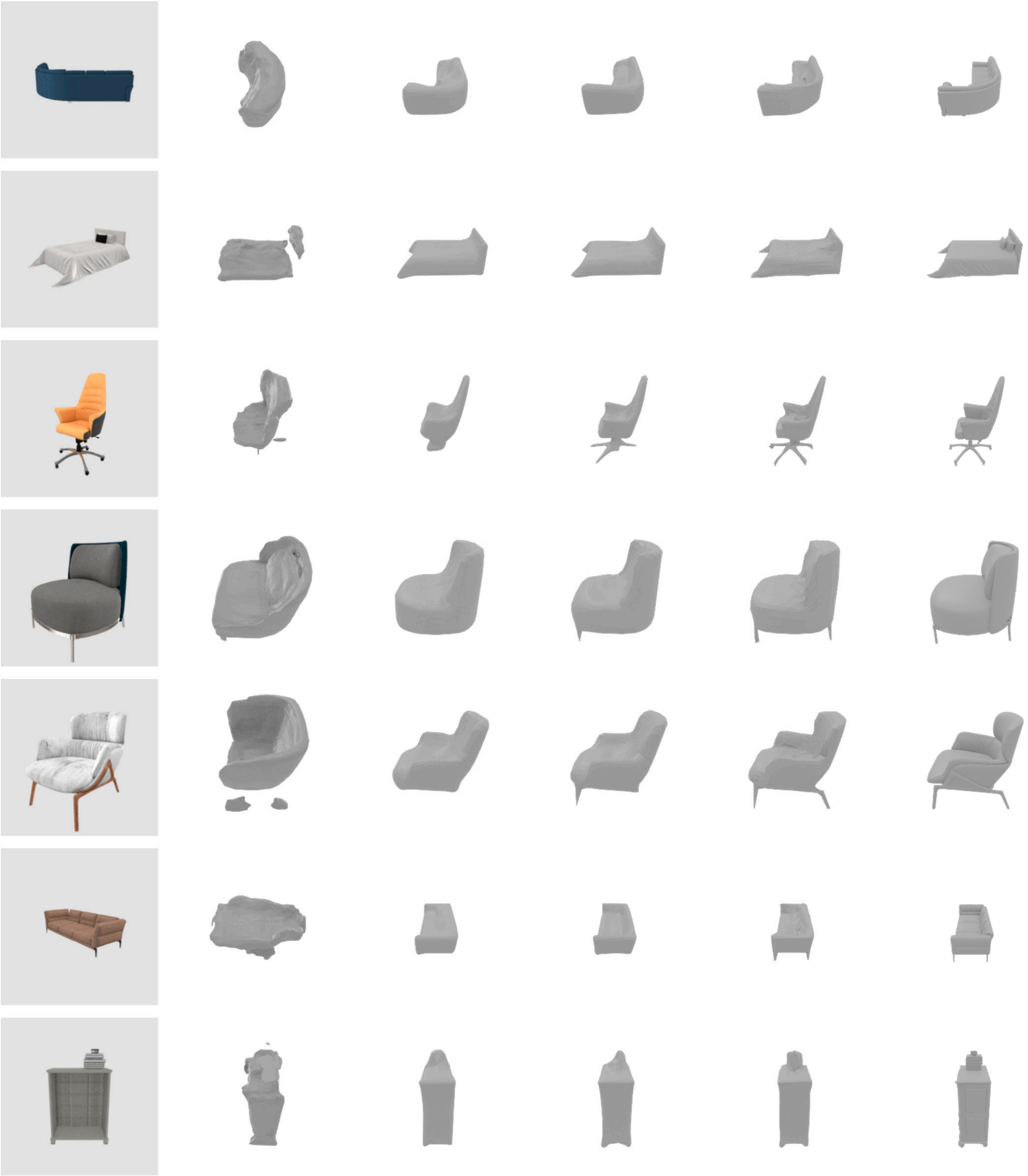

Figure 4. *Cont.*

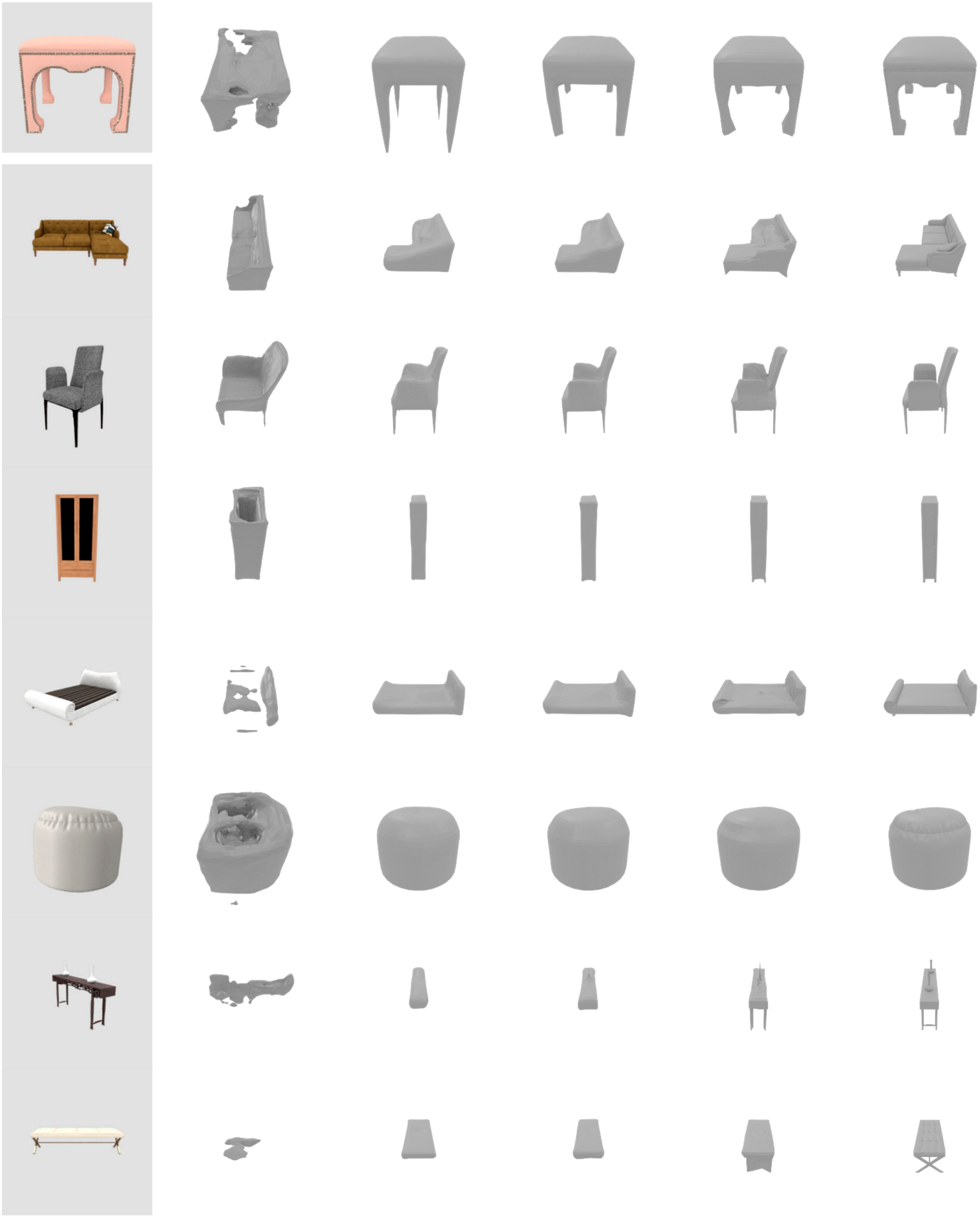

Figure 4. *Cont.*

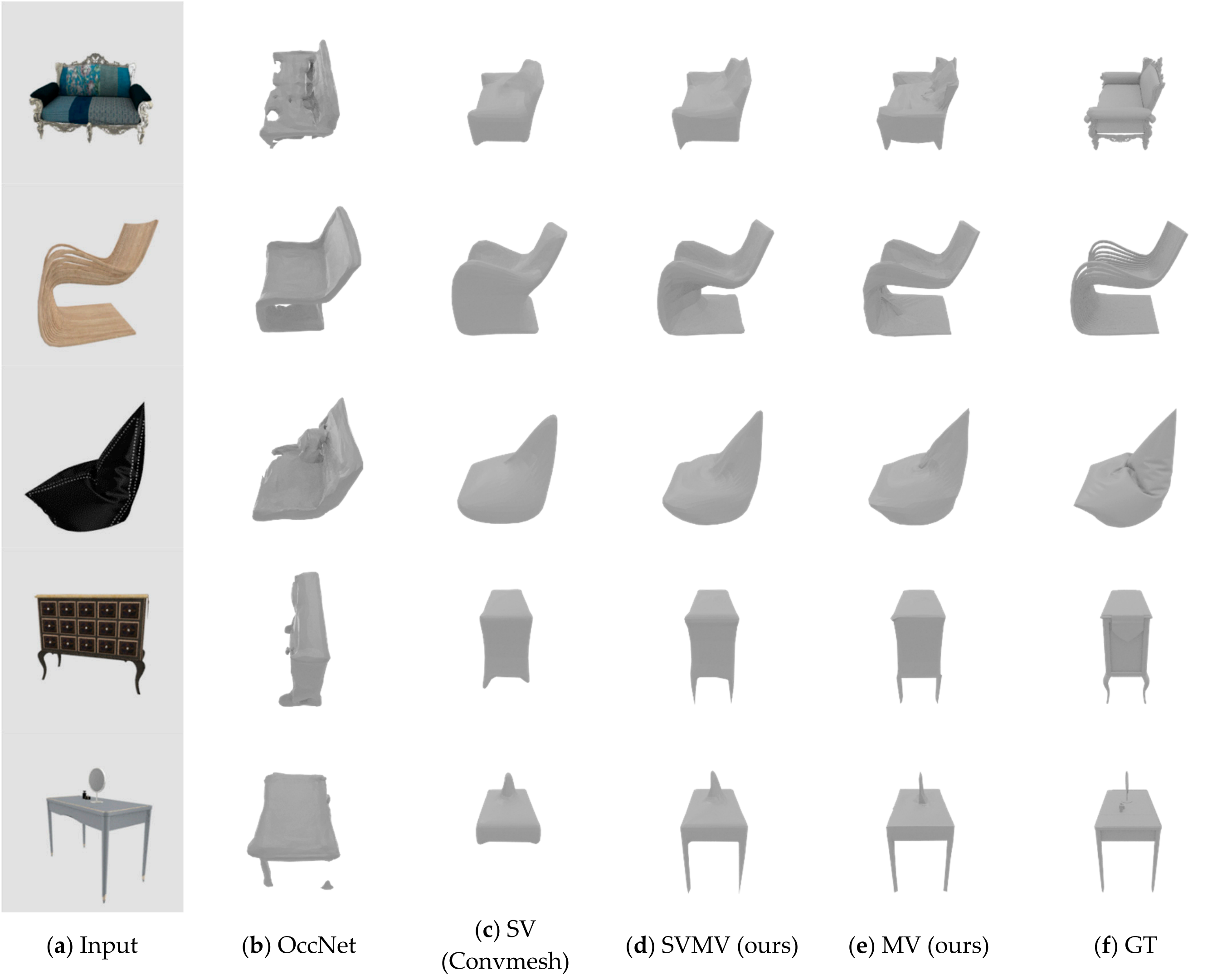

Figure 4. Mesh reconstruction results. Given the input (**a**) image, we compare the mesh generation results with those of our model trained using single view supervision (**d**) and multi-view supervision (**e**) to the results obtained using (**b**) OccNet and (**c**) SV(Convmesh). The ground truth mesh is shown in (**f**). All the mesh images were rendered using Blender.

As shown in Figure 4, OccNet failed to generate watertight meshes, such as the sofa (first row in Figure 4a), bed (second row in Figure 4a), and chair (fifth row in Figure 4a). These meshes are broken into several parts and are not very different from the ground-truth mesh. Meanwhile, the surfaces of the generated meshes are not flat. Convmesh was trained using only single-view supervision. As shown in Figure 4b, all the generated meshes have over-smooth surfaces. The Convmesh experimented on the CUB-200-2011 [36] and Pascal3D+ datasets [37]. The CUB-200-2011 dataset comprises images of 200 types of birds and the Pascal3D+ dataset comprises images of cars. Birds and cars have streamlined body shapes, and the gradients of their surfaces in local areas do not have significant differences. As a result, they demonstrate enhanced performance on the Pascal3D+ and CUB-200-2011 datasets. However, in the 3D-Future dataset, even the shapes in the same category, such as chairs, differ significantly, making the generation of accurate shapes more challenging. Thus, the model trained using only the single-view supervision of RGB images failed to recover the details of meshes. The reconstruction results from the model trained using the single-view and multi-source supervision approaches show improvements in the aspect of accuracy. However, the legs of the chairs, as shown in Figure 4, are not reconstructed completely compared to the ground-truth data. After adding multi-view supervision, the legs of the chairs were reconstructed, and the shapes of the reconstructed meshes were closer to the ground-truth meshes compared to other methods.

Quantitative evaluation was carried out in 3D space. The Chamfer distance (CD) and mean F-score were selected because they have been widely used in the evaluation of reconstructed 3D meshes. Then, 2048 3D points were uniformly sampled for both the ground truth and reconstructed meshes. These points were used to calculate the metrics. The quantitative comparison results are listed in Table 1. Table 1 shows the evaluation results by the categories of the input images and the best scores are shown in bold.

Table 1. The quantitative comparison results.

Category/Method	CD ($\times 10^{-3}$)				Mean F-Score (%)			
	OccNet	SV (Convmesh)	Ours (SVMS)	Ours (MVMS)	OccNet	SV (Convmesh)	Ours (SVMS)	Ours (MVMS)
Children Cabinet	158.11	26.74	7.54	**6.28**	31.49	72.43	92.91	**94.90**
Nightstand	131.17	32.88	17.34	**13.82**	30.51	73.31	86.43	**89.15**
Bookcase	219.01	14.84	**10.25**	10.53	30.28	83.82	88.64	**90.80**
Wardrobe	167.27	17.94	7.17	**6.40**	31.67	82.97	94.06	**95.38**
Coffee Table	129.96	47.92	26.34	**17.82**	35.60	60.27	79.02	**85.76**
Corner/Side Table	149.49	107.50	64.38	**42.22**	33.52	55.41	75.65	**81.21**
Side Cabinet	262.03	17.32	9.28	**6.46**	24.61	80.82	91.64	**94.74**
Wine Cabinet	224.54	11.51	**6.52**	6.68	29.70	87.83	93.64	**94.23**
TV Stand	258.91	8.93	5.25	**3.70**	26.17	88.59	95.91	**97.30**
Drawer Chest	179.88	17.7577	8.45	**6.90**	31.43	80.29	92.69	**95.01**
Shelf	164.90	33.04	28.88	**12.53**	30.02	73.90	78.65	**86.36**
Round End Table	65.35	51.40	36.40	**17.38**	49.32	55.13	73.96	**80.90**
Double/Queen/King Bed	88.11	25.12	14.72	**13.67**	45.17	71.63	83.68	**86.71**
Bunk Bed	165.46	42.38	27.44	**25.56**	29.27	59.71	65.54	**68.69**
Bed Frame	129.69	56.34	**30.11**	66.19	44.38	65.18	**87.20**	71.93
Single Bed	99.20	18.08	12.23	**10.06**	44.45	78.25	85.97	**90.11**
Kid's Bed	160.13	29.81	22.02	**16.64**	33.46	66.44	76.27	**81.29**
Dining Chair	122.08	34.36	**14.35**	18.27	45.18	77.75	**89.13**	85.66
Lounge/Office Chair	122.59	41.81	19.42	**12.64**	40.94	66.37	81.31	**89.15**
Dressing Chair	182.86	45.69	30.28	**25.85**	29.50	58.51	70.59	**75.39**
Classical Chinese Chair	80.71	37.06	33.02	**22.11**	49.01	72.55	74.37	**78.69**
Barstool	96.35	79.20	38.93	**30.71**	41.42	63.62	**82.81**	79.90
Dressing Table	238.09	68.22	18.82	**8.74**	21.83	51.00	84.35	**93.53**
Dining Table	181.60	80.11	37.84	**18.97**	30.38	47.01	73.44	**89.57**
Desk	270.61	68.27	31.71	**12.43**	22.30	51.68	76.69	**88.99**
Three-Seat Sofa	198.80	10.91	7.86	**6.50**	31.85	84.34	89.81	**92.80**
Armchair	105.91	29.48	17.44	**12.44**	39.53	68.24	81.47	**89.15**
Loveseat Sofa	191.80	12.69	9.40	**7.58**	29.97	82.65	88.35	**91.76**
L-shaped Sofa	176.48	12.90	10.27	**9.60**	34.34	85.10	90.79	**93.23**
Lazy Sofa	120.35	19.18	11.44	**7.27**	35.99	81.42	89.44	**95.37**
Chaise Longue Sofa	157.61	14.07	10.46	**7.80**	37.25	79.47	87.07	**91.81**
Stool	125.78	29.02	19.46	**16.32**	33.90	78.58	88.81	**91.87**
Pendant Lamp	169.55	81.31	73.23	**60.29**	34.49	58.88	58.44	**64.75**
Ceiling Lamp	102.73	29.27	22.18	**20.18**	40.50	75.39	79.76	**82.55**
Mean	159.57	38.13	23.12	**18.11**	34.56	71.57	83.35	**87.50**

In both the Chamfer distance and mean F-score evaluations, our models (SVMS and MVMS) defeated OccNet and Convmesh in all categories. The results presented in Table 1 show that our proposed method (MVMS) achieves significant improvement compared to OccNet in both the Chamfer distance and the mean F-score, even when OccNet is trained using 3D supervision. However, our proposed models were trained using 2.5D (rendered RGB images, depth images, and normal images) supervision.

4.3. Dataset Generation for Texture Reconstruction

In the texture reconstruction experiment, we designed and generated the ground truth for calculating the loss during the training process. Although the 3D-Future dataset provides high-resolution textures because the mappings of the provided textures to the generated meshes are irregular, the network cannot use the provided texture for learning. To create textures that the network can learn to generate, a ground truth texture should be made, such that it can be used to render an RGB image similar to the RGB image rendered using the original 3D model and texture. To achieve this objective, first, high-resolution (1024 × 1024) images were rendered using a DIB-Renderer for each 3D model using the provided 3D model and texture. We then projected this high-resolution image onto the predefined UV map and masked out the visible part as the ground truth texture, as illustrated by [7]. Examples of the rendered ground-truth RGB images and their corresponding ground-truth texture images are presented in Figure 5.

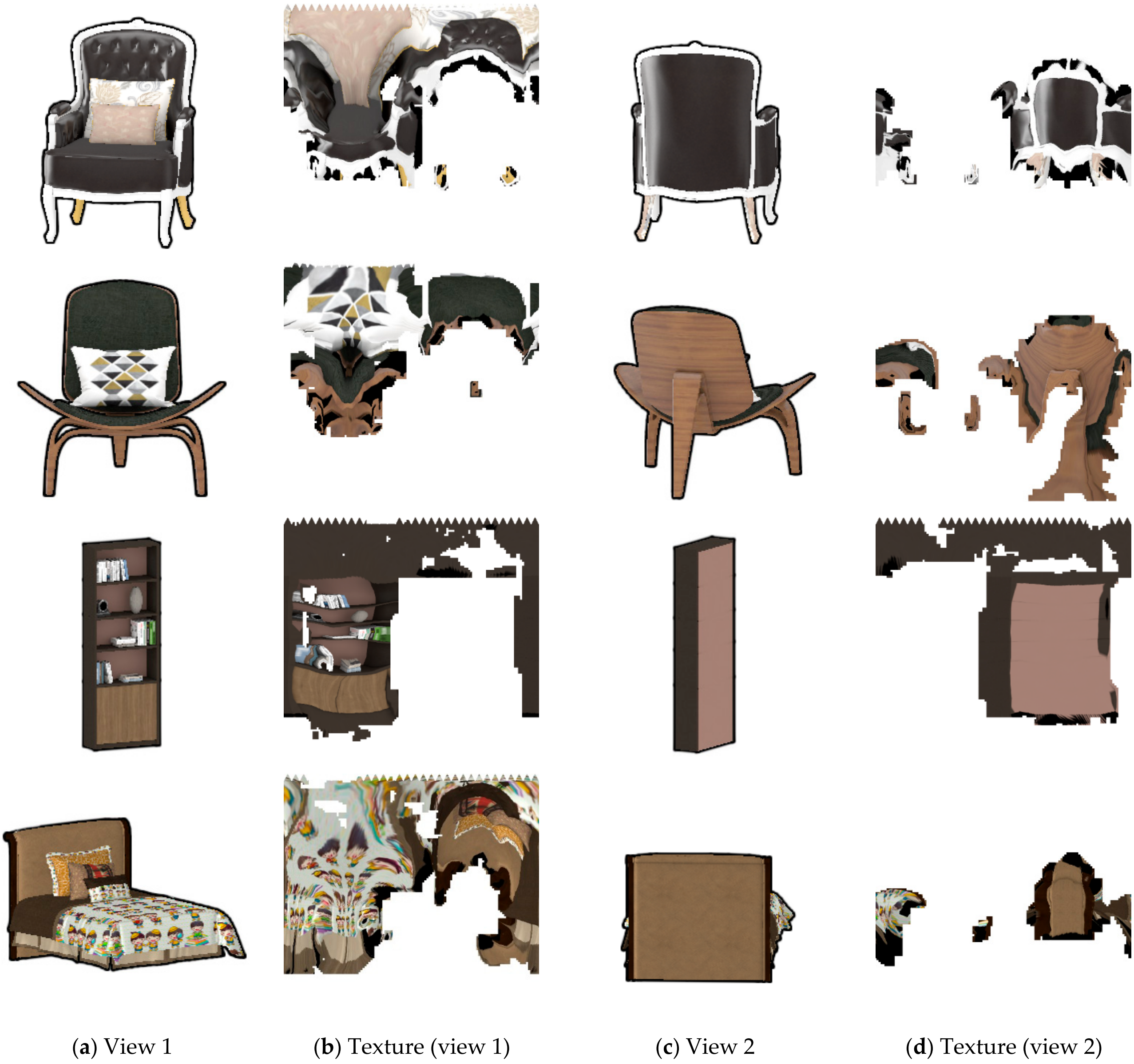
(**a**) View 1 (**b**) Texture (view 1) (**c**) View 2 (**d**) Texture (view 2)

Figure 5. Generated ground truth rendered image and corresponding texture image. The (**a**,**c**) columns show the rendered image, the (**b**,**d**) columns show the corresponding texture images that are visible in column (**a**,**c**). Image and texture of each row belong to one 3D model.

4.4. Experimental Results of Texture Generation

In the texture-reconstruction task, we adapted the generator of SRGAN [38] for our generator, and the discriminator of pix2pixHD [30] as our discriminator. The ground truth image was generated using the DIB-Renderer with generated textures, as shown in Figure 5b,d as well as the reconstructed meshes obtained from the mesh-reconstruction network. The reason to use reconstructed meshes is that the discrepancies between ground truth meshes and reconstructed meshes hinder the convergence of the training of GAN. If we use the rendered image from the ground-truth mesh, the discriminator will gradually learn to discriminate between real and fake images by identifying the mesh instead of the texture, making the generator unable to achieve convergence. We compared our reconstructed results with the results obtained from our proposed mesh-reconstruction network and Convmesh. Convmesh was trained using the generated textures presented in Figure 5b,d. The discriminator discriminates the real and fake images in the texture space. Figure 6 shows the texture reconstruction results.

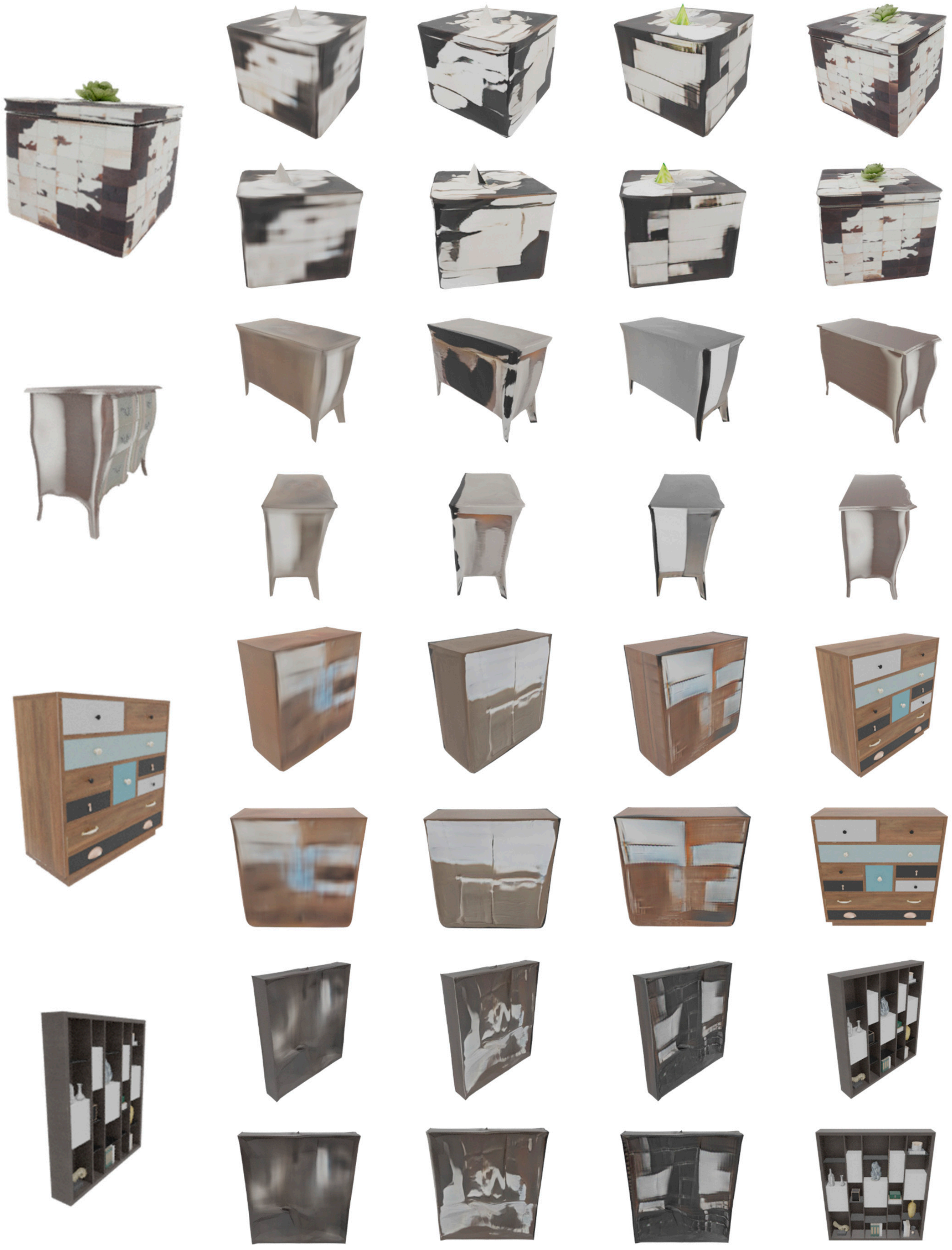

Figure 6. *Cont.*

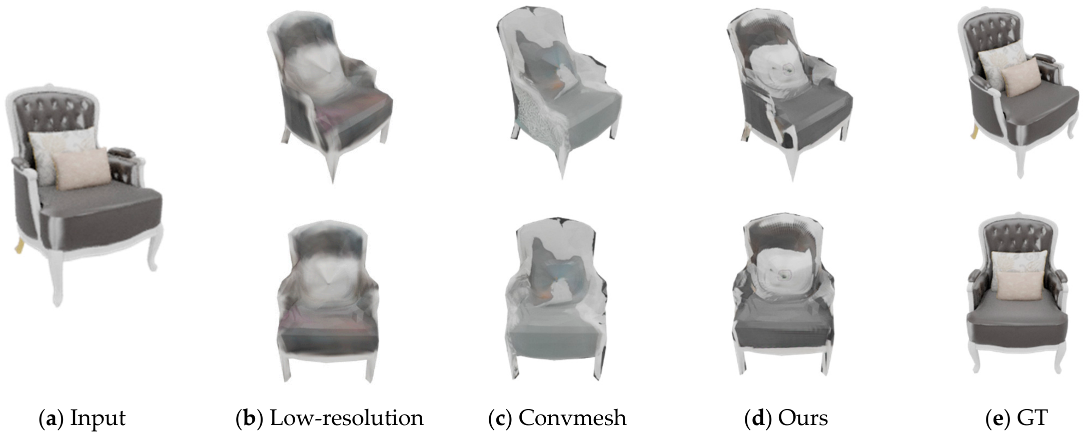

Figure 6. The texture reconstruction results. Given the input image (**a**), the reconstructed texture of mesh reconstruction (**b**), Convmesh (**c**), and our proposed texture reconstruction network (**d**) are illustrated from left to right. The rendered images with ground-truth textured meshes are presented in (**e**). Among them, image of (**e**) was rendered using low-resolution texture (128 × 128). Two views of the reconstructed results were provided for each input image.

The textured images generated from the mesh-reconstruction network are very blurry, and they cannot be used to describe the details of the appearances of the objects, as shown in Figure 6b. The textures obtained from Convmesh demonstrate high frequency details. However, the rendered image is not consistent with the original input image. For example, the results of wardrobe images shown in Figure 6 using Convmesh failed to reflect the appearance of the input image. In contrast, our results demonstrated high-frequency details and improved accuracy.

To evaluate the texture reconstruction results, the Fréchet inception distance (FID) [39] was computed. The FID score was evaluated on the rendered 2D images. It shows that its evaluation score can reflect the judgments of humans effectively [40]. The images used to compute the FID score were rendered using reconstructed textured meshes and ground-truth textured meshes. The quantitative evaluation results are listed in Table 2 and the best score is shown in bold.

Table 2. The quantitative comparison results of texture reconstruction.

Method	FID
Mesh reconstruction (ours) [1]	85.39
Convmesh	77.28
Texture super-resolution (ours)	**58.35**

[1] The texture output from mesh-reconstruction network.

Table 2 proves that our proposed texture reconstruction method can be used to generate more accurate textured meshes. Our proposed texture-reconstruction achieved FID scores of 58.35, which are better than those of Convmesh.

5. Discussion

In the mesh-reconstruction experiments, OccNet failed to predict continuous meshes in many cases (bed and chair), as shown in Figure 4. This may be due to their mesh-reconstruction method, which utilizes voxel-based representation. Like point clouds, it cannot be used to represent the relationship between vertices. Convmesh failed to generate sharp and accurate meshes. This was a result of training the networks using single-view supervision of RGB images. Evidently, single RGB images cannot provide enough supervision to enable the network to learn accurate shapes of objects. Thus, in our proposed method, multi-view and multi-source supervision approaches were utilized, and they demonstrated significant improvements in performance.

In the texture-reconstruction experiments, the texture images generated using the mesh-reconstruction networks were very blurry. However, they give a hint of the global texture appearance. Thus, when the low-resolution texture was input to a super-resolution network and trained using perceptual loss, the texture quality became enhanced in terms of the FID score and high-frequency details. Convmesh failed to capture the global appearance. One reason may be because it lacks global appearance hints, such as low-resolution textures. Simultaneously, it must capture the high-frequency details of objects to reconstruct a texture image. This may be an overwhelming task for a single network. Additionally, its discriminator was trained using texture images only and the perceptual loss, calculated on the feature space of the VGG-19 network trained using ImageNet dataset, could not be leveraged. Therefore, our proposed texture-reconstruction network demonstrates significant enhancements in FID score evaluation compared to Convmesh. We also noticed that in some examples (1–6 rows of Figure 6) the low-resolution results have less error than our high-resolution results, this may be because the texture reconstruction network recognized some pixels as noise and removed them.

However, our proposed methods demonstrate weaknesses in the reconstruction of objects with complex topologies, especially when they pertain to hollowed-out structures, although studies already exist, such as those conducted by Pan et al. and Nie et al. [15,40], which can modify the topologies of generated meshes. The methods proposed in these studies only aim to resolve problems associated with mesh reconstruction. Additionally, the modification of topologies makes the predefined UV maps lose the connection between the generated meshes and textures. Thus, modifying the topologies of template mesh is not suitable for reconstruction textured meshes. Reconstructing textures with high quality for arbitrary topology objects remains a challenge.

6. Conclusions

In this study, we proposed a method for reconstructing a high-resolution textured mesh from a single image. Our proposed method reconstructs textured meshes by deforming a template mesh associated with a fixed UV map. Therefore, the task of textured-mesh reconstruction is transformed to the task of estimating a deformation map and a texture image. After applying the deformation map and texture, the template mesh is expected to be the reconstructed textured mesh of the object in the input image. The proposed method comprises two main networks: a mesh-reconstruction network that outputs a deformation map, which is used to deform the template mesh and generate a low-resolution texture image, and a texture-reconstruction network, which is used to generate high-resolution textured images given a low-resolution image, deformation map, and input image. Comparisons between our proposed approaches and existing methods were conducted both quantitatively and qualitatively. The results prove that our proposed approach achieves an improved evaluation score, and it predicts clearer and highly consistent textured meshes.

Author Contributions: Methodology, M.W.; software, M.W.; validation, M.W.; writing—original draft preparation, M.W.; writing—review and editing, M.W., J.P., K.C.; visualization, M.W.; funding acquisition, K.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (2018R1A2B2007934).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://tianchi.aliyun.com/specials/promotion/alibaba-3d-future [31] (accessed on 22 October 2021).

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

References

1. Sun, Y.; Liu, Z.; Wang, Y.; Sarma, S.E. Im2avatar: Colorful 3d reconstruction from a single image. *arXiv* **2018**, arXiv:1804.06375.
2. Tulsiani, S.; Zhou, T.; Efros, A.A.; Malik, J. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2626–2634.
3. 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; pp. 4460–4470.
4. Oechsle, M.; Mescheder, L.; Niemeyer, M.; Strauss, T.; Geiger, A. Texture fields: Learning texture representations in function space. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 4531–4540.
5. Alldieck, T.; Pons-Moll, G.; Theobalt, C.; Magnor, M. Tex2shape: Detailed full human body geometry from a single image. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 2293–2303.
6. Novotny, D.; Shapovalov, R.; Vedaldi, A. Canonical 3D Deformer Maps: Unifying Parametric and Non-Parametric Methods for Dense Weakly-Supervised Category Reconstruction. In *Advances in Neural Information Processing Systems, Proceedings of the 34th Conference on Neural Information Processing Systems, Online Event, 6–12 December 2020*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates, Inc.: Vancouver, BC, Canada, 2020; Volume 33, pp. 20901–20912.
7. Pavllo, D.; Spinks, G.; Hofmann, T.; Moens, M.F.; Lucchi, A. Convolutional Generation of Textured 3D Meshes. In *Advances in Neural Information Processing Systems, Proceedings of the 34th Conference on Neural Information Processing Systems, Online Event, 6–12 December 2020*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates, Inc.: Vancouver, BC, Canada, 2020; Volume 33, pp. 870–882.
8. Kanazawa, A.; Tulsiani, S.; Efros, A.A.; Malik, J. Learning category-specific mesh reconstruction from image collections. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 371–386.
9. Hong, D.; Gao, L.; Yokoya, N.; Yao, J.; Chanussot, J.; Du, Q.; Zhang, B. More diverse means better: Multimodal deep learningmeets remote-sensing imagery classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 4340–4354. [CrossRef]
10. Zhou, X.; Gan, Y.; Xiong, J.; Zhang, D.; Zhao, Q.; Xia, Z. A method for tooth model reconstruction based on integration of multimodal images. *J. Healthc. Eng.* **2018**, *2018*, 1–8. [CrossRef] [PubMed]
11. Pontes, J.K.; Kong, C.; Sridharan, S.; Lucey, S.; Eriksson, A.; Fookes, C. Image2mesh: A Learning Framework for Single Image 3d Reconstruction. In *Asian Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 365–381.
12. Wu, J.; Wang, Y.; Xue, T.; Sun, X.; Freeman, W.T.; Tenenbaum, J. MarrNet: 3D Shape Reconstruction via 2.5D Sketches. In *Advances in Neural Information Processing Systems, Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Long Beach, CA, USA, 2017; Volume 30, pp. 540–550.
13. Kato, H.; Ushiku, Y.; Harada, T. Neural 3d mesh renderer. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3907–3916.
14. Groueix, T.; Fisher, M.; Kim, V.G.; Russell, B.C.; Aubry, M. A papier-mâché approach to learning 3d surface generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 216–224.
15. Pan, J.; Han, X.; Chen, W.; Tang, J.; Jia, K. Deep mesh reconstruction from single rgb images via topology modification networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 9964–9973.
16. Zhu, J.; Xie, J.; Fang, Y. Learning adversarial 3d model generation with 2d image enhancer. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 7615–7622.
17. Xu, Q.; Wang, W.; Ceylan, D.; Mech, R.; Neumann, U. DISN: Deep Implicit Surface Network for High-quality Single-view 3D Reconstruction. In *Advances in Neural Information Processing Systems, Proceedings of the 33st Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Vancouver, BC, Canada, 2019; Volume 32, pp. 492–502.
18. Lorensen, W.E.; Cline, H.E. Marching cubes: A high resolution 3D surface construction algorithm. *ACM Siggraph Comput. Graph.* **1987**, *21*, 163–169. [CrossRef]
19. Saito, S.; Huang, Z.; Natsume, R.; Morishima, S.; Kanazawa, A.; Li, H. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 2304–2314.
20. Chen, W.; Ling, H.; Gao, J.; Smith, E.; Lehtinen, J.; Jacobson, A.; Fidler, S. Learning to predict 3d objects with an interpolation-based differentiable renderer. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 9609–9619.
21. Henderson, P.; Tsiminaki, V.; Lampert, C.H. Leveraging 2d data to learn textured 3d mesh generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 7498–7507.

22. Deng, J.; Cheng, S.; Xue, N.; Zhou, Y.; Zafeiriou, S. Uv-gan: Adversarial facial UV map completion for pose-invariant face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7093–7102.
23. Saito, S.; Wei, L.; Hu, L.; Nagano, K.; Li, H. Photorealistic facial texture inference using deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5144–5153.
24. Park, E.; Yang, J.; Yumer, E.; Ceylan, D.; Berg, A.C. Transformation-grounded image generation network for novel 3d view synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3500–3509.
25. Rematas, K.; Nguyen, C.H.; Ritschel, T.; Fritz, M.; Tuytelaars, T. Novel views of objects from a single image. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1576–1590. [CrossRef] [PubMed]
26. Zhou, T.; Tulsiani, S.; Sun, W.; Malik, J.; Efros, A.A. View Synthesis by Appearance Flow. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 286–301.
27. Liu, S.; Li, T.; Chen, W.; Li, H. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 15–20 June 2019; pp. 7708–7717.
28. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
29. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90.
30. Wang, T.C.; Liu, M.Y.; Zhu, J.Y.; Tao, A.; Kautz, J.; Catanzaro, B. High-resolution image synthesis and semantic manipulation with conditional gans. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8798–8807.
31. Johnson, J.; Alahi, A.; Fei-Fei, L. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 694–711.
32. Dosovitskiy, A.; Brox, T. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems, Proceedings of the 30st Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016*; Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R., Eds.; Curran Associates, Inc.: Barcelona, Spain, 2016; Volume 29, pp. 658–666.
33. Fu, H.; Jia, R.; Gao, L.; Gong, M.; Zhao, B.; Maybank, S.; Tao, D. 3D-FUTURE: 3D Furniture shape with TextURE. *arXiv* **2020**, arXiv:2009.09633.
34. The Blender Foundation. Available online: https://www.blender.org/ (accessed on 5 August 2021).
35. Welinder, P.; Branson, S.; Mita, T.; Wah, C.; Schroff, F.; Belongie, S.; Perona, P. *Caltech-UCSD Birds 200*; Technical Report CNS-TR-2010-001; California Institute of Technology: Pasadena, CA, USA, 2010.
36. Xiang, Y.; Mottaghi, R.; Savarese, S. Beyond pascal: A benchmark for 3d object detection in the wild. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Steamboat Springs, CO, USA, 24–26 March 2014; pp. 75–82.
37. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.
38. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems, Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Long Beach, CA, USA, 2017; Volume 30, pp. 6629–6640.
39. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 586–595.
40. Nie, Y.; Han, X.; Guo, S.; Zheng, Y.; Chang, J.; Zhang, J.J. Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 55–64.