remote sensing

MDPI

Article

RealPoint3D: Generating 3D Point Clouds from a Single Image of Complex Scenarios

Yan Xia [1,2], Cheng Wang [2,*], Yusheng Xu [1], Yu Zang [2], Weiquan Liu [2], Jonathan Li [3] and Uwe Stilla [1]

1 Photogrammetry and Remote Sensing, Technical University of Munich, 80333 Munich, Germany; xiayan@stu.xmu.edu.cn (Y.X.); yusheng.xu@tum.de (Y.X.); stilla@tum.de (U.S.)
2 Fujian Key Laboratory of Sensing and Computing, School of Informatics, Xiamen University, Xiamen 361005, China; zangyu7@xmu.edu.cn (Y.Z.); wqliu1026@163.com (W.L.)
3 Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada; junli@uwaterloo.ca
* Correspondence: cwang@xmu.edu.cn

Received: 2 October 2019; Accepted: 3 November 2019; Published: 13 November 2019

Abstract: Generating 3D point clouds from a single image has attracted full attention from researchers in the field of multimedia, remote sensing and computer vision. With the recent proliferation of deep learning, various deep models have been proposed for the 3D point cloud generation. However, they require objects to be captured with absolutely clean backgrounds and fixed viewpoints, which highly limits their application in the real environment. To guide 3D point cloud generation, we propose a novel network, RealPoint3D, to integrate prior 3D shape knowledge into the network. Taking additional 3D information, RealPoint3D can handle 3D object generation from a single real image captured from any viewpoint and complex background. Specifically, provided a query image, we retrieve the nearest shape model from a pre-prepared 3D model database. Then, the image, together with the retrieved shape model, is fed into RealPoint3D to generate a fine-grained 3D point cloud. We evaluated the proposed RealPoint3D on the ShapeNet dataset and ObjectNet3D dataset for the 3D point cloud generation. Experimental results and comparisons with state-of-the-art methods demonstrate that our framework achieves superior performance. Furthermore, our proposed framework works well for real images in complex backgrounds (the image has the remaining objects in addition to the reconstructed object, and the reconstructed object may be occluded or truncated) with various viewing angles.

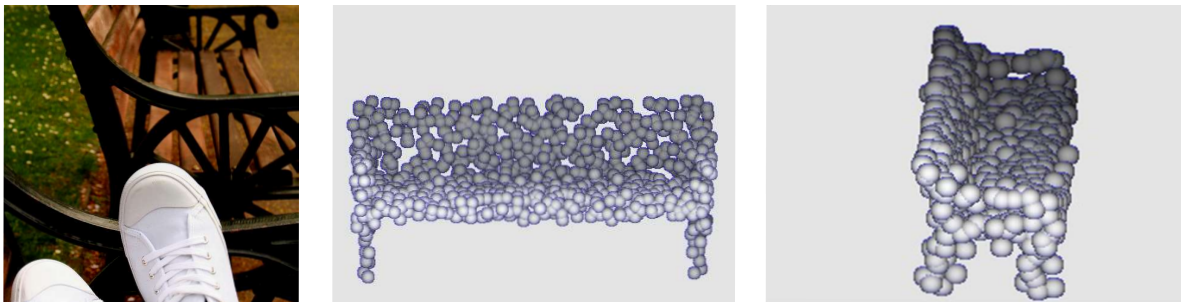
Keywords: point cloud generation; single image; indoor scenario; deep learning; 3D reconstruction

1. Introduction

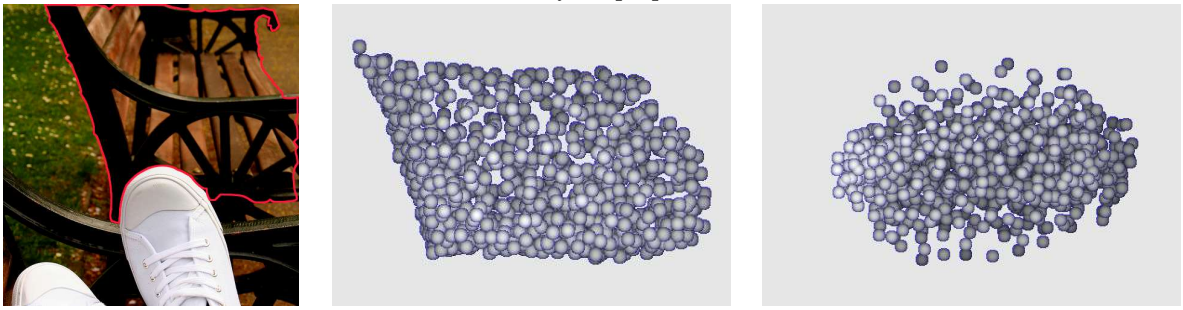
Generating 3D point clouds from a single image, which aims to provide solutions to tasks such as autonomous driving, virtual reality and robotic surgery, is a fundamental, intriguing area in remote sensing and computer vision. Humans can perceive the surrounding environment and objects in three dimensions without any effort. In fact, we are very good at inferring 3D structure from a single image of an object. Besides, in biology, monocular vision plays a vital role in many animals, especially for prey. For example, for a pigeon, the visual fields of its eyes only have about 10–15 degree overlap [1]. Thus, we want to give machines the ability to perceive the surroundings. Recently, with the development of deep learning, many learning-based methods have been proposed for depth estimation and 3D reconstruction. Primarily, 3D reconstruction from a single image considered as an ill-posed problem has achieved promising results by using the deep neural network [2–4]. Different from traditional geometric-based approaches [5,6], learning-based methods, with the superpower

representation of the deep neural network, build a complex mapping from image space to 3D object space. By using a single view image, because some parts are invisible in the image, the networks usually must guess the shape of the object. Before feeding the 3D data into the neural network, the data are usually transformed into volumetric grids or 2D images [7] rendered from different views. Then, the 2D or 3D convolution is easily applied to the regular data. Voxelization, a common way of 3D representation, has achieved great success in object classification, detection, and segmentation [3,8–11]. Especially, recently-proposed Octree Generating Networks (OGNs) [3] have achieved impressive performances in 3D object generation. However, there is also an obvious disadvantage to the voxel-based method: how to balance sampling resolution and net efficiency is a difficult problem.

To tackle this problem, directly generating a 3D point cloud for the object might be the right choice. Compared with 3D meshes, or volumetric grid representations, the point cloud representation has several advantages: (1) A point cloud is a simple, uniform structure that a network can quickly learn. (2) Because all the points are independent and no connectivity information must be updated, global geometric transformation and deformation can easily be applied to a point cloud. Combining the advantages of point cloud representation, the Point Set Generation Network (PSGN) [2] was proposed to generate 3D point clouds from a single image directly. However, this framework requires the object to be captured with an absolutely clean background, at a specific viewpoint, and a certain distance. The specific viewpoint means the object is entirely in the image. The absolute clean backgrounds mean the image has only this object without any things. When these requirements are not satisfied (e.g., complex background), the reconstruction performance drops dramatically. A typical failed case for PSGN is shown in Figure 1b, where the first column is the input image and the remaining columns are generated 3D point clouds displayed in two different views. In this case, although an object mask was provided, only part of the bench was created. Because part of the bench is occluded and some parts are truncated, this image is really challenging.



(**a**) Generation result by the proposed RealPoint3D.



(**b**) Generation result by Point Set Generation Network (PSGN) [2].

Figure 1. An example of 3D point cloud generation from a single real image by RealPoint3D and PSGN. (**a**) The proposed RealPoint3D works well here without any mask information. (**b**) PSGN fails to generate the point cloud even the object segmentation mask is provided.

For generating the invisible parts, we propose exploring prior shape information (i.e., knowledge common to humans) to help the 3D generation network. Specifically, because humans have built a

huge 3D object database in their brains, they can easily imagine a 3D shape even if only a small part is visible, and the other parts are occluded. Thus, provided with a small piece of information about an object in a 2D image, a human can easily find a similar 3D model in this database. Inspired by this, we propose a prior-knowledge-guided 3D generation framework to reconstruct the 3D point cloud from a single real image. To achieve the prior knowledge, an on-line object retrieval stage is added before the generation framework. Here, the object retrieval process is based on a pre-prepared database, which stores some conventional 3D object models together with their corresponding image features. For each model, the image features are extracted from a number of 2D images rendered from different viewpoints. During the retrieval stage, the nearest 3D shape is searched by comparing the features extracted from the query image and these stored in the database. Compared with the 3D generation, which requires dense image features, image retrieval, because it relies mainly on some local, sparse distinguished features, is very robust to occlusion, change of viewpoints and complex backgrounds [12]. By adding the retrieved 3D model, the proposed framework can handle the 3D generation from a single image captured in a real scenario. To simplify the expression, we refer to the proposed network as "RealPoint3D", where "RealPoint" consists of two meanings: On the one hand, it is designed especially for images captured from a real environment. On the other hand, the output of our network is a practical, complete 3D point cloud of the object, including both the visible and invisible parts.

We performed the proposed RealPoint3D on images of five categories of objects from ShapeNet dataset [13] and ObjectNet3D dataset [14]. Experimental results show the state-of-the-art performance on the synthetic rendered and real images for our proposed method. In summary, the main contributions of this paper are as follows:

- We design a novel and end-to-end network RealPoint3D, which combines a 2D image and 3D point cloud for 3D object reconstruction. By using prior knowledge, RealPoint3D reconstructs objects from an image with a complex background and change of viewpoints.
- Extensive experiments on images of five common categories of objects were conducted and the results demonstrate the effectiveness of the RealPoint3D framework.

2. Related Work

2.1. 3D Reconstruction from a Single Image

To address 3D structure recovery from a single projection, which, theoretically, is an ill-posed problem, many attempts, such as using massive SFM and SLAM [15,16] methods, were made. However, they usually need many textures on objects and specular reflections. In addition, if the viewpoints are separated by a large baseline, it makes the feature correspondences [17] very problematic because of the local appearance changes and self-occlusion. ShapeFromX, where X can be texture, specularity, shadow, etc. [18–20], also requires prior knowledge on natural images.

Boosted by the large-scale dataset of 3D CAD models [13], deep learning based generative methods have been widely employed in 3D reconstruction. Generally, these methods can be categorized into voxel-based and point-cloud-based methods. 3D-GAN [9], which embeds generation in generative adversarial nets, outperforms other unsupervised learning methods by a large margin. Choy et al. [10] applied the 3D recurrent neural network (3D-R2N2) to Long Short-Term Memory (LSTM) to infer 3D models by taking several images rendered from different views. Using the octree representation, Octree Generating Networks (OGN) [3] first proposed a generative method for large scene 3D reconstruction by relieving the burden of storage and computation. Different from voxel-based methods, PSGN [2] generates point clouds from a single image directly. Pixel2Mesh [4] produces a 3D shape in a triangular mesh from a single RGB image. However, a clean background and a fixed viewpoint are necessary conditions for good reconstruction results.

2.2. Shape Prior Guided 3D Reconstruction

Different from a natural scene, we have sufficient prior shape knowledge for artificial objects, such as chairs, vehicles, etc. Employing this information, 3D reconstruction becomes much easier. Su et al. [5] and Huang et al. [6] reconstructed the depth of objects from images collected from websites by exploiting a collection of aligned 3D models of related shapes of objects. Mahabadi et al. [21] proposed a novel prior shape formulation to split an object into multiple convex parts and then formulate the reconstruction as a volumetric multi-label segmentation problem.

2.3. Deep Learning on Point Clouds

Recently, deep learning on point clouds has attracted the attention of more and more researchers. Voxelization, which transfers an unordered point cloud into regular grids, has been intuitively applied to 3D convolution. Maturana et al. [11] and Qi et al. [22] are pioneers in using voxel-based methods for object detection and classification. However, those methods can be applied only to a relatively small resolution with a sparse volume. PointNet [23] is an innovative architecture that can directly extract, from raw point cloud data, features, which can be used for classification and segmentation. PointNet++ [24], which can be seen as an extension of PointNet, employs multiple layers of different resolutions to increase the receptive fields of the network. Recently, there are also a wide variety of methods improved from PointNet by adding post-process and using similar multiscale strategy [25]. FoldingNet [26] proposes a novel deep auto-encoder to address unsupervised learning challenges in point clouds. SO-Net [27] achieves hierarchical feature extraction [28] in point clouds with a significantly faster training speed. Besides point clouds, many frameworks, such as the one proposed by Bronstein et al. [29], spectral graph CNN [30] and Geodesic CNN (GCNN) [31], have been proposed for mesh representation.

3. Methodology

3.1. Overview of the Workflow

Different from generative-based methods [2], in the proposed RealPoint3D approach, prior shape knowledge is taken into consideration for 3D object reconstruction. This information benefits 3D generation as follows: (1) Prior shape aids the network to eliminate the negative influence from complex backgrounds. (2) Shape information also guides a network to recover the 3D points of invisible parts which cannot be seen in a 2D image. For easy understanding, the flowchart of RealPoint3D is illustrated in Figure 2. The framework is divided into two major steps: (1) To retrieve the nearest 3D shape from a pre-prepared database, deep features are extracted from a query image. To robustly handle the change of viewpoint, multiple images from different perspectives are rendered for each model. (2) To generate the point cloud of the object, the RealPoint3D network uses the retrieval 3D model and 2D image as input. Detailed information for each step is introduced in the following sub-sections.

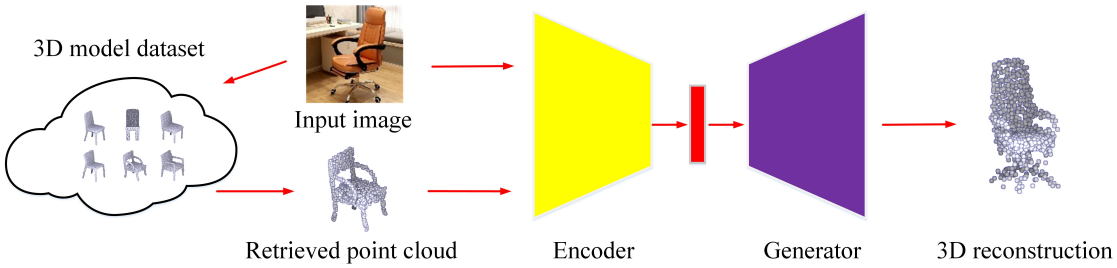


Figure 2. The flowchart of proposed RealPoint3D framework. First, a nearest 3D model is retrieved from a collected 3D model database based 2D image. Then, the retrieved 3D model together with the 2D image are fed into the network for 3D reconstruction.

3.2. Nearest Shape Retrieval

The nearest object retrieval process is summarized in Figure 3. A database, including some commonly used 3D models, is pre-prepared in advance. According to Su et al. [32], multiple images are rendered from different viewpoints for each model, e.g., eight directions. The rendering of the 3D object includes sampling lighting parameters and sampling camera parameters. For the lighting condition, N point lights are added and the environmental light is used. N is uniformly sampled from 1 to 10. The position plight is uniformly sampled on a sphere of radius 14.14, between latitude 0° and 60°; the energy is $E \sim N(4,3)$; and the color is fixed to be white. All lighting parameters are sampled i.i.d. In addition, to better simulate real-world scenarios, background textures are randomly added to the rendered images. The size of the rendered images is 128 × 128 pixels and they are textured with different texture levels.

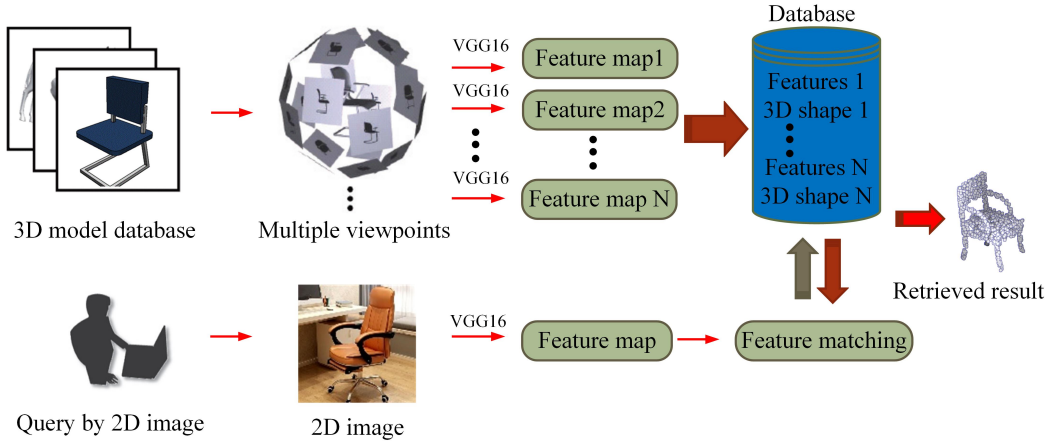


Figure 3. Nearest shape retrieval based 2D image. First, a group of 2D images are rendered from different views for each 3D model. Then, VGG-16 [33] network is applied to extract deep features from these 2D images. When a query image comes, the nearest 3D model of the object related to the image can be found by comparing the 2D features extracted from the query image and the database.

After these images are obtained, they must be transformed into compact features for further steps. Here, the VGG-16 network [33] is employed for feature computation. We choose VGG-16 as our preliminary work without the deliberate pursuit of efficiency and speed to verify whether our ideas are valid or not. In addition, VGG-16 is easy to understand and implement. A 4096-dimension feature vector is generated for each 3D model. The VGG-16 network extracts the good features of images. Our dataset is used to fine tune the pretrained model on ImageNet [34]. Finally, a feature map database dictionary is built for all 3D objects. When a query image appears, the nearest object model is obtained by comparing the features extracted from the query image with the feature map stored in the database dictionary. Similar to other image retrieval methods, the similarity between two feature vectors is measured by using the cosine distance, defined as:

$$Sim(X,Y) = \frac{XY}{\|X\|\|Y\|} \tag{1}$$

where X and Y are the two features to be compared.

3.3. RealPoint3D

After obtaining the nearest 3D object model, N (e.g., 1,024) points are uniformly sampled for the RealPoint3D network. For easy understanding, an overview of the proposed network is illustrated in Figure 4. From the encoder, the 2D Convolutional Neural Network (CNN) is used to extract the features from the 2D image. Influenced by PointNet++ [24], Multilayer Perceptron (MLP) is employed

to extract spatial information from the 3D point cloud. Then, to obtain more comprehensive features, the two types of features are combined via several fully connected layers. Subsequently, a generator stage, which includes convolution and deconvolution layers, is used to recover the 3D point cloud of the object.

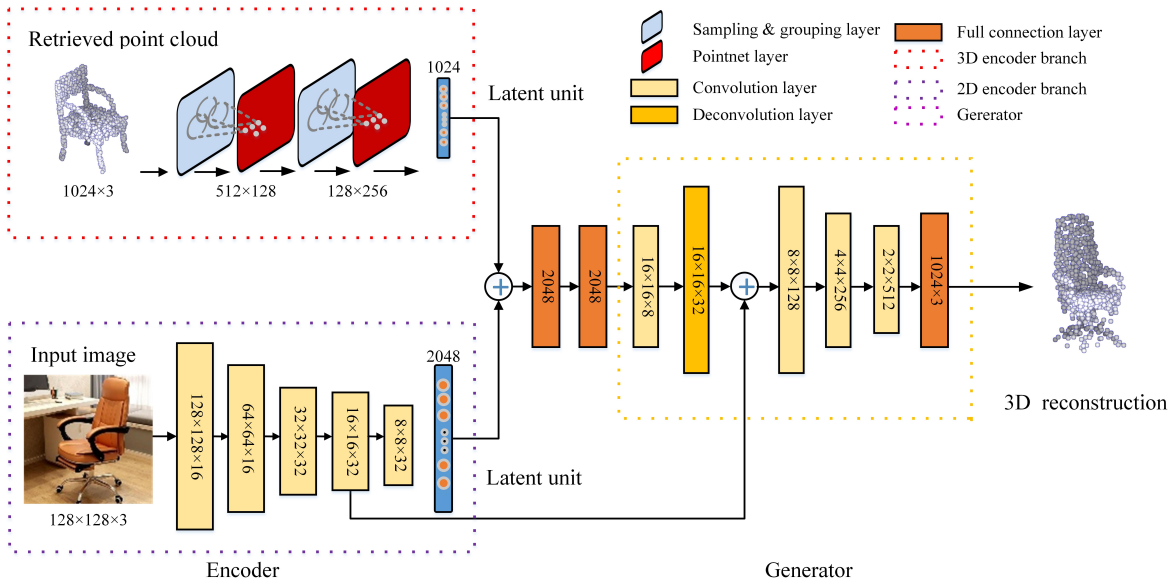


Figure 4. A flowchart of the proposed RealPoint3D network, in which the 2D feature is extracted by an encoder network and the 3D features from point cloud are extracted similarly as PointNet++ [24]. Then, the two kinds of features are concatenated together for the following generator.

3.3.1. Encoder Net

The encoder consists of two branches: one is used for 2D images, and the other is for the 3D point cloud. The 2D branch consists of several convolutional and ReLU layers. Finally, the 2D part outputs a 2048-dimensional feature vector. Similar to PointNet++, several set abstraction layers are employed in the 3D encoder. Each set abstraction layer consists of the sampling, grouping, MLP, and pooling layers. In RealPoint3D, we adopt two set abstraction layers and use the multi-scale grouping strategy [35] to obtain a global feature for the retrieved object model. Finally, the 3D encoder outputs a 1024-dimensional feature vector. Then, the two types of features are concatenated and fed to the following two fully connected layers. After obtaining bottleneck representations, we reshape the flat feature to a (16, 16, 8)-sized 3D tensor.

3.3.2. Generator

The generator consists of several convolutional, deconvolutional, and fully connected layers. Influenced by U-Net [8], the low-level features in the encoder stage are transferred directly to the generator stage to help recover the details of the object. Specifically, we concatenate the feature maps in the fourth convolutional layer to the deconvolutional layer. After three convolutional layers, the generator ends with a fully connected layer in the shape of 3072. Finally, we reshape the result to a point cloud with a size of 1024×3.

3.3.3. Influence of the 3D Encoder

To demonstrate the effectiveness of 3D feature extraction, a simplified version of the network has been designed for comparison. In this version, we have removed the 3D encoder and kept only the 2D image part. The following generator remains the same. Experimental results prove that, because of missing spatial information, performance drops dramatically. The comparison results are given in Section 4.3.

3.4. Loss Function

Enabling end-to-end training, a highly efficient and differentiable loss function must be designed. However, accurately measuring the topological similarity of two set 3D point clouds is very difficult. Unlike voxel-based approaches (e.g., 3D-R2N2), those methods output 0/1 signal to represent whether or not a point is inside a voxel. The proposed RealPoint3D network outputs the point cloud location. Therefore, the Softmax function cannot be used directly here. Usually, the Hausdorff distance is selected to measure the difference between two point sets; however, this distance is sensitive to a small number of outlier points in the sets. We explore Chamfer Distance (CD) to measure the difference between two point sets, defined as follows:

$$d_{CD} = \sum_{p \in S_1} \min_{q \in S_2} \| p - q \|_2^2 + \sum_{p \in S_2} \min_{q \in S_1} \| p - q \|_2^2 \quad (2)$$

where $S_1, S_2 \subseteq R^3$, p and q are the prediction and ground truth of each point cloud. CD, equal to the mean overall nearest neighbor distances, is differentiable concerning point locations and efficient to compute for two-point sets. Geometrically, CD induces a nice shape space. This is because CD is more robust to outliers, it is a better choice for the loss function. In addition, the range search for each point is independent, thus, trivially parallelizable. In addition, spatial data structures such as KD-tree are used to accelerate the nearest neighbor search.

4. Experiments

For assessing the performance of our proposed network, we performed several experiments on the rendered images from ShapeNet dataset [13] and real scene images from ObjectNet3D dataset [14].

4.1. Datasets and Implementation Details

The ShapeNet dataset [13] is an ongoing large-scale 3D model source widely used in 3D related research fields. Our experiments were based on one of its subsets, namely ShapeNetCore55, which covers 55 common object categories with about 51×10^3 unique 3D models. We split this dataset into training and testing sets, with 4/5 for training and the remaining 1/5 for testing. Therefore, the training sets and testing sets of the 3D models do not have any overlap. We selected five common categories which have high percentages of shape variations for evaluation here. In addition, it is especially difficult to generate 3D ground truth models for real images. We rendered CAD models to 128×128 images with complex backgrounds for our training and testing. ObjectNet3D [14] is another large-scale 3D model dataset including about 100 categories, 44,147 3D shapes and 201,888 objects in 90,127 images. We tested our model with this dataset because it has many real-world photos. The images were scaled to 128×128 before being fed to the network.

The proposed network RealPoint3D was implemented in the framework of TensorFlow, and Adam was taken as the optimizer. To improve performance, we chose the batch size as 32 and the gradient step as 2×10^5. The learning rate automatically decayed based on the number of iterations. The input image was resized to $128 \times 128 \times 3$, and the last fully connected layer produced 1024 3D points. In the encoder stage, kernel size was set at 3×3 for all convolutional layers. In the generator stage, we set the kernel size at 5×5 for all convolutional and deconvolutional layers. In addition, the multi-scale grouping strategy was employed in the 3D point cloud encoder. Ball query strategy was used to find neighboring points within a radius. We set $r = 0.2$ and 0.4 for two scales. ReLU was taken as the activation function in the whole network.

4.2. Results and Comparisons

4.2.1. Object Reconstruction on Rendered Images

We compared our approach with the PSGN of Fan et al. [2] and OGN of Tatarchenko et al. [3] and we selected five common categories for evaluation. We uniformly sampled 1024 points from each object for training. To have a fair comparison, following their experimental settings, we re-trained PSGN and OGN on our rendered images with complex backgrounds. In addition, the results were compared under two different metrics: CD, defined in Equation (2), and IoU (Intersection over Union), defined as:

$$IoU = \frac{PC_{pre} \cap PC_{gt}}{PC_{pre} \cup PC_{gt}} \tag{3}$$

where PC_{pre} and PC_{gt} are the predicted and ground truth of each point cloud.

Following their corresponding papers, we used IoU for evaluation with OGN and CD scores for PSGN. The results are shown in Tables 1 and 2, respectively.

Table 1. CD scores for different methods on images of complex scenarios.

Category	PSGN	Retrieval	RealPoint3D
Sofa	2.20	6.83	**1.95**
Airplane	1.00	3.67	**0.79**
Bench	2.51	2.11	**2.11**
Car	1.28	1.96	**1.26**
Chair	2.38	6.91	**2.13**

We omitted the coefficient 10^{-3} for all the values. A smaller number represents better performance.

The CD scores of the testing set for the five categories are shown in Table 1, where "Retrieval" is the retrieved nearest shape model and "RealPoint3D" is our proposed method. Here, all point clouds were normalized before the evaluation. The numbers are average point-wise distances. Our approach outperforms PSGN for all categories, especially for the bench. The reason is that the high retrieval accuracy of the bench can guide the network to generate a more accurate 3D point cloud. On the contrary, for cars, PSGN obtained a result similar to that of our model. This can be explained from two aspects: On the one hand, the variation of car models is not too considerable, which makes generating a 3D point clouds easier. On the other hand, because there are many similar car shapes in the dataset, the retrieval of cars is relatively tricky. A relatively inaccurate retrieval shape may even mislead the generation process. Nevertheless, our generated model is more accurate. In addition, the selected five categories have high variations in shape, which strongly indicates that the proposed approach works well for different kinds of objects.

For the voxel-based methods, we chose the OGN [3] for comparison. OGN converts the original organization voxel grids to a compact octree-based structure, which significantly improves the computational efficiency and reduces storage consumption. Five categories were evaluated. The results are shown in Table 2, which shows that our proposed network outperforms OGN for all categories. Especially for the bench, the value improved from 0.046 to 0.359. It is also worth noting that the proposed model generates a quite good point cloud, although the retrieved model (e.g., car) is not good enough.

Besides, highlighting the advantage of our method, we also designed experiments with clean background images, which were used by Fan et al. [2] and Tatarchenko et al. [3]. We used IoU as the evaluation criterion. Here, the IoU values for PSGN and OGN were selected directly from their papers. The IoU values for sofas for PSGN and OGN are vacant in Table 3 because those values are missing in their papers. In the table, it is noteworthy that RealPoint3D achieved the highest scores. For the car category, all three methods yielded quite good results. As mentioned above, because of its simple structure and fewer variances in shape, the 3D generation of a car is relatively easy. Particularly, it can

be seen in Tables 2 and 3 that the IoU values for the same category are different for complex and clean backgrounds. Values for complex backgrounds are somewhat lower than those for clean backgrounds, demonstrating that backgrounds strongly affect generation, particularly for objects with relatively complicated structures (e.g., sofas and chairs). In this situation, RealPoint3D demonstrates it strongly benefits from the retrieved of the 3D model.

Table 2. IoU scores for different methods on images of complex scenarios.

Category	OGN	Retrieval	RealPoint3D
Sofa	0.11	0.12	**0.22**
Airplane	0.15	0.36	**0.53**
Bench	0.05	0.17	**0.36**
Car	0.44	0.24	**0.54**
Chair	0.14	0.13	**0.27**

A higher value represents better performance here.

Table 3. IoU scores for different methods on images of clean backgrounds.

Category	PSGN	OGN	RealPoint3D
Sofa	-	-	**0.63**
Airplane	0.60	0.59	**0.67**
Bench	0.55	0.48	**0.58**
Car	**0.83**	0.82	**0.83**
Chair	0.54	0.48	**0.58**

A higher value represents better performance here.

4.2.2. Reconstructed 3D Points Using Real Images

We compared our approach to PSGN [2] and OGN [3]. Five common categories were selected for evaluation. Some results are visualized in Figure 5. In the rows, from top to bottom, the five categories are bench, car, airplane, chair, and sofa, respectively. Particularly, for RealPoint3D and OGN, we took the whole image (the first column in Figure 5) as the inputs for 3D generation; for PSGN, only the foreground objects inside the masks (the fourth column of Figure 5) were used for generation because, if we took the whole image as input, PSGN would fail.

To highlight the strength of our proposed method, five types of objects were chosen for comparison. The second, third, and the fifth columns are reconstruction results of RealPoint3D, OGN, and PSGN, respectively. In Figure 5, it can be seen that RealPoint3D achieves more fine-grained object details than the other two methods and is applicable for both indoor and outdoor scenes. For example, in the first row of Figure 5, RealPoint3D recovers the legs of the bench that was shot in a park; whereas, while the other two approaches totally miss them. Even with object masks, PSGN still cannot recover the detailed structures of the bench. The fourth row is a chair in a study room; RealPoint3D reconstructs the armrests of the chair, while OGN fails and PSGN outputs a distorted point cloud. Based on this experiment, the proposed method is more applicable to the real world. In particular, a single image cannot provide enough information for 3D reconstruction. The geometry for invisible parts must be estimated by the network. Therefore, PSGN and OGN can obtain only a global shape without fine-grained details. With the help of the retrieved nearest 3D shape, the proposed network achieves a better result, especially for the invisible parts.

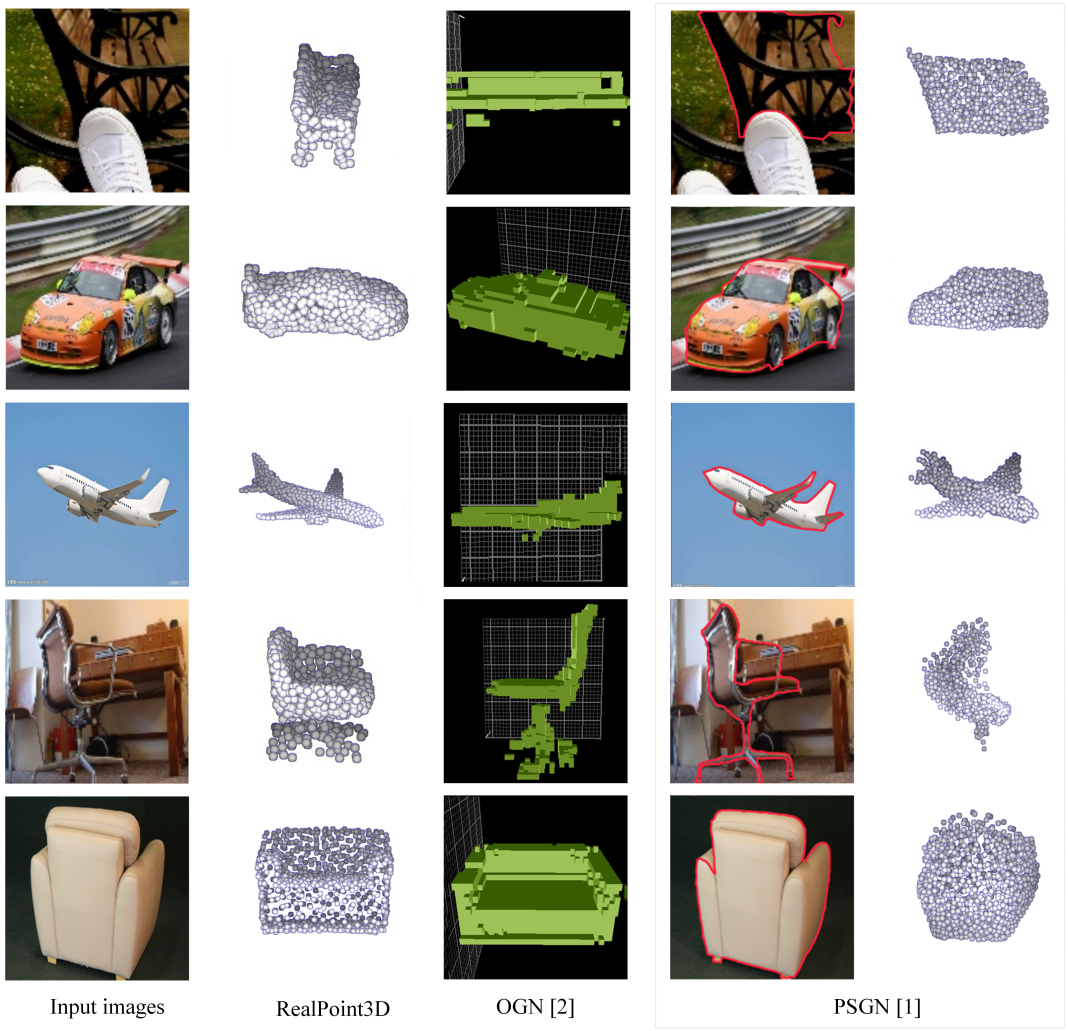


Figure 5. Reconstructed 3D points using real images from ObjectNet3D dataset. From left to right: input 2D images, RealPoint3D, OGN [3], and PSGN [2] with object segmentation masks.

4.3. Model Analysis

To further evaluate the effect of the prior-knowledge constraint, a group of experiments was designed and carried out. Specifically, three types of results along with the quantitative statistics are included. In Figure 6, the first column (Non-guidance) shows the results without any point cloud constraint (i.e., removing the 3D encoder and keeping only the 2D image part of RealPoint3D); the second column (Selected-guidance) shows the results when the 3D shape retrieval part fails to generate the strictly similar point cloud constraint of the input image; and the last column (Searched-guidance) shows the results under the perfect retrieval result. It is shown that the retrieval result indeed affects the reconstruction performance, while our proposed RealPoint3D framework also has a quite good tolerance for the failed guidance. In Figure 6, as shown in the second column, even the searched point cloud is not similar to the input image, and the result is prevented from ultimately failing. However, if the point cloud guidance were directly removed, as shown in the first column part, the reconstruction result would become unacceptable. On the contrary, RealPoint3D network with the searched-guidance recovers more details of the objects.

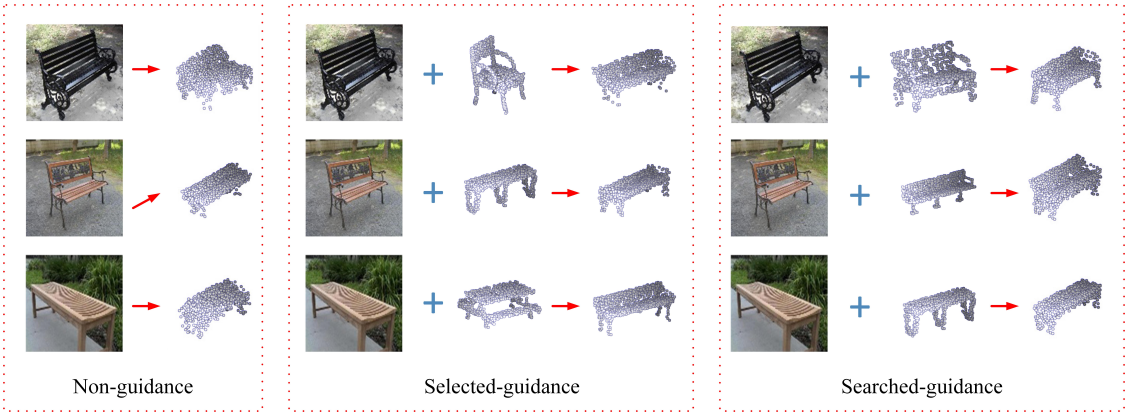


Figure 6. Visualization of model analysis on real images with different viewpoints. The first column is the results without any point cloud constraint; the second column part is the results with failed retrieval constraint; the third column part is the results with perfect retrieval guidance.

Corresponding quantitative statistics are shown in Table 4, where all five kinds of data are employed for testing. Results of non-guidance, random-guidance, and searched-guidance are shown in the second, third, and fourth columns, respectively. The performance of the non-guidance is, unsurprisingly, rather poor: about 40% lower than the searched-guidance results. For the random-guidance result, we selected 50 random point cloud guidance (with the same type of the input image) and list the average reconstruction performance in the third column. The random-guidance performance is about 9% lower than the results of the searched-guidance, which implies that the reconstruction part has good tolerance when the guidance is not satisfactory. If the 3D model retrieval part were able to provide satisfying guidance, the reconstruction results would be quite promising.

Table 4. CD scores for different models on images of complex scenarios.

Category	Non-Guidance	Random-Guidance	Searched-Guidance
Sofa	2.46	2.10	**1.95**
Airplane	1.38	0.88	**0.79**
Bench	3.55	2.39	**2.11**
Car	1.31	1.28	**1.26**
Chair	2.53	2.35	**2.13**

We omitted the coefficient 10^{-3} for all the values. A smaller number represents better performance. Non-guidance: the generated results without any point cloud constraint; Random-guidance: the generated results with 50 randomly selected point cloud guidance and calculating the average CD scores; Searched-guidance: the generated results under the perfect retrieval result.

4.4. Time Complexity

For the time complexity of our proposed method, in current experiments, 500 epochs were performed in the training stage, which took approximately 10 h on five NVIDIA Tesla P100 GPUs. In the testing stage, around 0.1 s per image were required on a laptop with a CPU. As a comparison, the computational efficiency of our proposed method is similar to that of PSGN, but it is significantly more efficient than OGN, which takes about 1.6 s per image for inference.

5. Conclusions

In this paper, we design a new generation network, RealPoint3D, that is more suitable for 3D fine-grained reconstruction from a single image in a real scenario. Different from previous generative methods, the retrieval of nearest 3D points is used as prior knowledge and implemented before the core generation network, which prompts the generator to reconstruct more object details from images with complex backgrounds and changing viewpoints. Compared with other

generative methods, state-of-the-art performance is achieved with reconstruction from real images with complex environments.

As a conclusion, the dominant superiority of our proposed RealPoint3D network is twofold. The first one is the advantage of our network framework. Actually, from a single image, it is difficult to fully determine the reconstruction of a 3D shape because the provided information is limited. In addition, if the object in an image is truncated, it is difficult to reconstruct with just a single image. To guide the 3D reconstruction, we propose adding a nearest 3D point cloud as prior knowledge. The proposed network, RealPoint3D, is effective with a single real image captured from any viewpoint and complex background, even if the object is truncated, which benefits from the fusion of image features and 3D spatial features. The second one is the benefit of using the nearest 3D point cloud. We first adopt the current highly accurate image retrieval method to obtain a similar 3D shape corresponding to the input image. We build a feature map database dictionary for all the objects in the pre-prepared 3D model database. Therefore, retrieval processing is robust. In addition, the similar 3D point cloud not only provides label information but reveals strong prior model information. Although our method achieves excellent performance in 3D reconstruction from a single image with any viewpoint and intricate backgrounds, the generated 3D point clouds are still sparse. We think that the distance function is not suitable as a loss function for the large-scale point cloud generation. In addition, when the input image has two models inside, RealPoint3D only reconstructs the more dominant object since the retrieved point cloud is more similar to the more dominant one. In the process of preparing training data, we only use the single object in one image, and the ground truth is a point cloud of this object. Thus, we believe it is a limitation for RealPoint3D and all current 3D object reconstruction methods listed in the paper, including PSGN, OGN, and so on. Another reason is that we do not implement any detection or attention mechanism, thus RealPoint3D will produce distorted output. In this work, RealPoint3D can only reconstruct a single object from a single image. We will explore the 3D scene reconstruction from a single image in the future.

Author Contributions: Conceptualization, Y.X. (Yan Xia); methodology, Y.X. (Yan Xia); formal analysis, Y.X. (Yan Xia) and C.W.; data curation, Y.X. (Yan Xia); writing—original draft preparation, Y.X. (Yan Xia); writing—review and editing, Y.X. (Yan Xia), C.W., Y.X. (Yusheng Xu), Y.Z., W.L., J.L., and U.S.; and funding acquisition, C.W.

Funding: This research was funded by Natural Science Foundation of China grant number U1605254. The first author Yan Xia was supported by the China Scholarship Council (No. 201906310130) for his PhD study at Technical University of Munich. This work was supported by the German Research Foundation (DFG) and the Technical University of Munich within the funding program Open Access Publishing

Acknowledgments: We are grateful to Dingfu Zhou, Feixiang Lu, Xinyu Huang and Ruigang Yang for their suggestions that greatly improved our manuscript.

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

References

1. Fristrup, K.M.; Harbison, G.R. How do sperm whales catch squids? *Mar. Mammal Sci.* **2002**, *18*, 42–54. [CrossRef]
2. Fan, H.; Su, H.; Guibas, L.J. A Point Set Generation Network for 3D Object Reconstruction from a Single Image. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; Volume 2, p. 6.
3. Tatarchenko, M.; Dosovitskiy, A.; Brox, T. Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs. *arXiv* **2017**, arXiv:1703.09438.
4. Wang, N.; Zhang, Y.; Li, Z.; Fu, Y.; Liu, W.; Jiang, Y.G. Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images. In Proceedings of the European Conference on Computer Vision (ECCV) 2018, Munich, Germany, 8–14 September 2018.
5. Su, H.; Huang, Q.; Mitra, N.J.; Li, Y.; Guibas, L. Estimating image depth using shape collections. *ACM Trans. Graph. (TOG)* **2014**, *33*, 37. [CrossRef]

6. Huang, Q.; Wang, H.; Koltun, V. Single-view reconstruction via joint analysis of image and shape collections. *ACM Trans. Graph. (TOG)* **2015**, *34*, 87. [CrossRef]
7. Hong, D.; Yokoya, N.; Chanussot, J.; Zhu, X.X. An augmented linear mixing model to address spectral variability for hyperspectral unmixing. *IEEE Trans. Image. Process.* **2019**, *28*, 1923–1938. [CrossRef] [PubMed]
8. Cicek, O.; Abdulkadir, A.; Lienkamp, S.S.; Brox, T.; Ronneberger, O. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2016; pp. 424–432.
9. Wu, J.; Zhang, C.; Xue, T.; Freeman, W.T.; Tenenbaum, J.B. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2016; pp. 82–90.
10. Choy, C.B.; Xu, D.; Gwak, J.Y.; Chen, K.; Savarese, S. 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 628–644.
11. Maturana, D.; Scherer, S. Voxnet: A 3d convolutional neural network for real-time object recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 922–928.
12. Hong, D.; Liu, W.Q.; Wu, X.; Pan, Z.K.; Su, J. Robust palmprint recognition based on the fast variation Vese–Osher model. *Neurocomputing* **2016**, *174*, 999–1012. [CrossRef]
13. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H. ShapeNet: An Information-Rich 3D Model Repository. *arXiv* **2015**, arXiv:1512.03012.
14. Xiang, Y.; Kim, W.; Chen, W.; Ji, J.; Choy, C.; Su, H.; Mottaghi, R.; Guibas, L.; Savarese, S. Objectnet3d: A large scale database for 3d object recognition. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 160–176.
15. Fuentes-Pacheco, J.; Ruiz-Ascencio, J.; Rendón-Mancha, J.M. Visual simultaneous localization and mapping: A survey. *Artif. Intell. Rev.* **2015**, *43*, 55–81. [CrossRef]
16. Häming, K.; Peters, G. The structure-from-motion reconstruction pipeline—A survey with focus on short image sequences. *Kybernetika* **2010**, *46*, 926–937.
17. Hong, D.; Yokoya, N.; Chanussot, J.; Zhu, X.X. Cospace: Common subspace learning from hyperspectral-multispectral correspondences. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 4349–4359. [CrossRef]
18. Barron, J.T.; Malik, J. Shape, Illumination, and Reflectance from Shading. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1670–1687. [CrossRef] [PubMed]
19. Malik, J.; Rosenholtz, R. Computing Local Surface Orientation and Shape from Texture for Curved Surfaces. *Int. J. Comput. Vis.* **1997**, *23*, 149–168. [CrossRef]
20. Savarese, S.; Andreetto, M.; Rushmeier, H.; Bernardini, F.; Perona, P. 3D Reconstruction by Shadow Carving: Theory and Practical Evaluation. *Int. J. Comput. Vis.* **2007**, *71*, 305–336. [CrossRef]
21. Karimi Mahabadi, R.; Hane, C.; Pollefeys, M. Segment based 3D object shape priors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015, Boston, MA, USA, 7–12 June 2015; pp. 2838–2846.
22. Qi, C.R.; Su, H.; Niebner, M.; Dai, A.; Yan, M.; Guibas, L.J. Volumetric and Multi-view CNNs for Object Classification on 3D Data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 5648–5656.
23. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, Honolulu, HI, USA, 21–26 July 2017; Volume 1, p. 4.
24. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; pp. 5099–5108.
25. Huang, R.; Ye, Z.; Hong, D.; Xu, Y.; Stilla, U. Semantic Labeling and Refinement of LIDAR Point Clouds Using Deep Neural Network in Urban Areas. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *IV-2/W7*, 63–70. doi:10.5194/isprs-annals-IV-2-W7-63-2019. [CrossRef]

26. Yang, Y.; Feng, C.; Shen, Y.; Tian, D. FoldingNet: Point Cloud Auto-Encoder via Deep Grid Deformation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA, 18–22 June 2018.
27. Li, J.; Chen, B.M.; Hee Lee, G. SO-Net: Self-Organizing Network for Point Cloud Analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), in Salt Lake City, UT, USA, 18–22 June 2018
28. Hong, D.; Liu, W.Q.; Su, J.; Pan, Z.K.; Wang, G.D. A novel hierarchical approach for multispectral palmprint recognition. *Neurocomputing* **2015**, *151*, 511–521. [CrossRef]
29. Bronstein, M.M.; Bruna, J.; Lecun, Y.; Szlam, A.; Vandergheynst, P. Geometric Deep Learning: Going beyond Euclidean data. *IEEE Signal Process. Mag.* **2016**, *34*, 18–42. [CrossRef]
30. Yi, L.; Su, H.; Guo, X.; Guibas, L. SyncSpecCNN: Synchronized Spectral CNN for 3D Shape Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 6584–6592.
31. Masci, J.; Boscaini, D.; Bronstein, M.M.; Vandergheynst, P. Geodesic Convolutional Neural Networks on Riemannian Manifolds. In Proceedings of the IEEE International Conference on Computer Vision Workshop 2015, Santiago, Chile, 11–18 December 2015; pp. 832–840.
32. Su, H.; Qi, C.R.; Li, Y.; Guibas, L.J. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In Proceedings of the IEEE International Conference on Computer Vision 2015, Santiago, Chile, 11–18 December 2015; pp. 2686–2694.
33. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
34. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
35. Wu, X.; Hong, D.; Ghamisi, P.; Li, W; Tao R. Msri-ccf: Multi-scale and rotation-insensitive convolutional channel features for geospatial object detection. *Remote Sens.* **2018**, *10*, 1990. [CrossRef]