Cascade Convolution Neural Network for Point Set Generation

Chenmou Wu,1 Hyo Jong Lee,1,2 Member, IEEE
1Department of Computer Science & Engineering, Jeonbuk National University, Jeonju 54896, South Korea.
2Center for Advanced Image and Information Technology, Jeonbuk National University, Jeonju 54896, South Korea
Corresponding author: Hyo Jong Lee (e-mail: hlee@chonbuk.ac.kr).

This research has been supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (GR2019R1D1A3A03103736)

ABSTRACT Automatic and efficient 3D object modeling has become critical in industrial applications. The advancement of deep convolutional neural networks (CNNs) has prompted researchers to use CNNs for learning 3D geometry information directly from images. However, the feature maps directly extracted by CNNs are more suitable for image processing tasks because they contain more deep texture information of the entire 2D image. Compared with this, 3D reconstruction tasks using CNNs demand geometric information about a specific object. The existing architecture mainly tries to infer the geometric structure through texture information, which leads to an uneven distribution of points in the generated point cloud object. To address this problem, we propose a cascade point set generation network (CPSGN) that deforms the predicted object while more effectively inferring the object’s 3D geometric information from the 2D image so that the distribution of the final object becomes more uniform and denser. The CPSGN consists of a point set generation part for producing a basic 3D object and a point deformation part for fine-tuning the basic 3D object. In addition, we designed a projection loss that optimizes the geometry of the model by measuring shape differences from multiple perspectives. Experimental results on different benchmark datasets indicate that the produced point-based model outperforms existing approaches.

INDEX TERMS 3D Reconstruction, Point Cloud, CNN, Point-based Model

I. INTRODUCTION
The reconstruction of 3D objects from 2D images is attracting much attention for various applications such as mapping in robotics navigation, 3D animation, and 3D printing [1]. According to the information of an input 2D image and the representation of an output 3D object, 3D reconstruction includes multiple approaches. For the visual representation, researchers commonly adopt meshes, voxels, or point clouds to represent generated objects [2]. Although all three types of representation can effectively represent and visualize 3D objects, point cloud provides more advantages in industrial applications that require special geometrical information, and offers a higher degree of freedom with less memory [3]. More specifically, voxel-based representation takes an enormous amount of memory with low computational efficiency, especially at high resolution [4], and mesh-based representation indicates poor flexibility and scalability since it is commonly derived from point clouds [5]. Furthermore, point cloud is a standard 3D acquisition format that is especially suitable for geometric operations and transformation. It is widely used in common 3D scanning devices such as Kinect, LiDAR and ToF camera [6]. We also adopted point cloud as our final presentation to benefit from its efficiency and flexibility compared with the other representations.

From 2D input information, 3D reconstruction can simply be split into a single-view image and multiple view images reconstruction. Traditional 3D reconstruction research has focused on multiple view images. The main idea is that multiple view geometry [7] recovers the 3D geometric structure information of the object from images of the static object from different viewpoints. The common approaches include structure from motion [8], simultaneous localization and mapping [9], scene reconstruction [10], and point cloud alignment [5]. Compared with multiple images 3D reconstruction, it is more difficult to predict 3D geometric information from a single-view image.
By benefiting from deep learning technology and large-scale 3D object datasets [11], single-view 3D reconstruction methods [3,4,12,13] have achieved great success in recent years. A common method is to use variational autoencoder (VAE) theory [12] to generate a 3D object. It encodes the 2D information of the image in a latent space and directly constructs a 3D object from latent information and constrains the distribution of the predicted point set close to ground truth by a specific objective function. However, methods similar to VAE architecture might have the following challenges:

(i) In the encoder part, the convolution layer is commonly used to extract deep features from the original image. However, texture features make up a large part of constructed feature maps extracted by CNN, especially feature maps from deep layers [13]. Thus, the feature vector in a latent space will lose a lot of geometry information. It is difficult to directly infer the geometry of the object though texture features. Excessive texture features promote the network to focus on the large surface part of the image, which turns 3D point set prediction into a recognition or classification problem [14]. The core task of the network changes from reconstruction to the closest shape searched according to the database [4].

(ii) In 3D reconstruction, the Chamfer distance (CD) [15] is a loss function used by most researchers that can measure the difference between two different point-based distributions. It forces each point in a predicted point set approach to the corresponding position by minimizing the function [3]. However, two similar point-based distributions only guarantee that the overall shape of a product is close to real objects. Therefore, the constraint ability of the CD function has a limitation in geometric structures, especially in complex geometric structures. Additionally, the CD function induces a splashy shape that blurs the shape of the geometric structure, while it is sensitive to outliers [14]. With the constraint of the CD function, the predicted points may be clustered at the center of the body part of the label, which causes the body part of a product to become dense and other parts to become sparse. When the distribution of the predicted point set is close to the label, the CD function will directly ignore a small number of outliers during the optimization process. Therefore, optimization easily falls into a local minimum and the network learns poorly for sparse distributions and the geometric information based on CD functions.

(iii) Furthermore, the number of points and the degree of sparseness depend on the application. In fact, we usually set a fixed value as the size of predicted points for training of the rule of neural network methods. The common size of the predicted point set that can basically describe the shape of a 3D object is 1024 [14]. However, many real applications may require a larger number of points for a single object.

To address the above problems, we propose a novel neural network, called a cascade point set generation network (CPSGN), to generate a 3D object from a single-view image. As shown in Figure 1, the CPSGN is composed of two parts: a point generation network and a point deformation network. Rather than directly synthesizing a point cloud object, we adopted a coarse-to-fine strategy that generates a primary point-based 3D object with a basic geometric shape from a 2D image and then deforms the primary object to large size point sets in a point-based CNN, as depicted in Figure 2. In the cascade network, the final 3D object not only benefits from the deep texture features of a 2D image, but also increase detail information and predicted points resolution while each layer maintains the global features of the primary point cloud object. The deformation network can cover the shortage of sparse and rough points from the generation network. At the same time, dynamic graph convolutional neural networks (DGCNN) have presented an edge
convolution method to learn both local and global geometric features from point clouds [15]. Edge convolution generates edge features that describe the relationships between a point and its neighbors. We designed a deformation network based on the edge convolution module.

Furthermore, the CPSGN regularizes predictions from a global perspective and thus can work in complement with previous CD loss for better object reconstruction from a single image [8]. Due to the characteristics of a point cloud, it is difficult to measure the differences in a 3D shape on a point cloud directly, yet it is relatively easy in 2D projection [16]. The geometric information can be preserved well from different views. Thus, we propose a projection loss that uses multiple 2D projection images to focus on measuring geometric similarity. It regularizes prediction globally by enforcing the prediction to be consistent with the ground-truth among different 2D views and following the 3D semantics of a point cloud.

The main contributions of this work are as follows:
1. We propose a novel cascade architecture that models a 3D object from a 2D image. The network first models a 3D object from 2D texture information, and then deforms the point set into a dense point set.
2. We propose a loss function, called projection loss, to regularize the generated object from the geometric shape closest to the real object.
3. We evaluated the CPSGN on benchmark datasets and found that it achieves state-of-the-art performance.

II. RELATED WORK

A. 3D RECONSTRUCTION FROM A SINGLE IMAGE

Most traditional 3D reconstruction methods [7-9,33] focus on multi-view images. Recently, data-driven methods for single-image 3D reconstruction [4],[17] have received more attention. Based on object representations, a single-image 3D reconstruction includes three streams. It is easier to transform 2D features extracted by CNN to reconstruct a 3D object using voxels, a regular grid representation. Choy et al. [17] proposed a 3D recurrent neural network (3D-R2N2) using long- and short-term memory to convert 2D features to 3D features and to generate a voxel-based object from a single or multiple images. Follow-up work [18] used a backpropagated ray trace pooling operation to learn 2D silhouettes and adversarial constraint to regularize voxel-based objects with unlabeled 3D shapes. Wu et al. [19] proposed a VAE architecture by combining two generative models that encode images in a latent space and reconstruct a voxel-based model. Tulsiani et al. [20] proposed a geometric consistency constraint to jointly learn a pose estimation network and voxel-based object-generated network using unsupervised learning.

For clearer visualization, mesh is another direction in 3D object generation. Wang et al. [4] proposed a cascade ResNet network based on graph convolution to generate a mesh-based 3D object. Pontes et al. [21] used free-form deformation and a sparse linear combination to infer the parameters of a compact mesh representation. Kato et al. [22] proposed a gradient-based 3D mesh network using silhouette image supervision. Fan et al. [3] first proposed a novel point set generation network (PSG) that encodes a 2D image and outputs an unordered point-based object. Based on PSG, Jiang et al. [23] proposed adversarial loss to constrain prediction geometry.

B. POINT CLOUD FEATURE EXTRACTION

Although the CNN achieves excellent success in image feature extraction, it is hard to maintain the same effect since a point cloud is an unordered representation. Qi et al. [24] proposed the pioneering PointNet, which directly extracts
point-based representation by using multilayer perceptrons and global pooling, to address this problem. However, PointNet was unable to extract local features adequately. As another approach, PointNet++ [25] was proposed to integrate global and local representations with increased computation cost. Furthermore, many researchers have applied machine learning methods such as KNN and KD-tree to aid PointNet in comprehensively extracting point cloud features. KD-Net [26] built a KD-tree for input point clouds, followed by hierarchical feature extraction from leaves to root. SO-Net [27] proposed using the spatial distribution of point clouds extracted by a self-organizing map before PointNet architecture. Wang et al. [15] proposed an edge convolution method to extract geometric relationships among points, which considers neighborhoods of points rather than acting on each one independently. In this work, we adopted edge convolution as our feature extractor in the deformation part since geometric information is essential for upsampling and reordering a point set.

III. METHOD
The CPSGN adopts a cascade strategy to design the architecture to force the output point cloud in uniform distribution within the geometric information form of the 2D image. More specifically, given a single-view image, the first stage network (point generation network) generates a fundamental point cloud with rough shapes and sparse points, then a dense and uniformly distributed point cloud is generated by the next stage network (point deformation network) based on the fundamental output and 2D geometric feature. Compared to the previous single-view point-cloud reconstruction methods, which directly generate the 3D object from 2D image, our method reduces the problem of point cloud concentration on edges, corners and excessive reliance on deep texture features.

In the following section, we first briefly introduces the edge convolution kernel used in the deformed part. Then, we explain the architecture of the point generation network and the point deformation network in detail. The loss function is discussed at the end.

A. PRELIMINARY: EDGE CONVOLUTION
We first provide some background about edge convolution used in the deformation part; a more detailed introduction can be found in [15]. Given a point cloud with n points \( P = \{p_1, \ldots , p_n\} \subseteq \mathbb{R}^3 \), the edge convolution layer (EdgeConv) produces an F-dimensional point cloud with the same number of points. More specifically, EdgeConv builds a directed graph \( \mathcal{G} = (\mathcal{V}, \mathcal{E}) \) using k-nearest neighbor to represent the local structure of the point cloud, where \( \mathcal{V} = \{1, \ldots , n\} \) denote vertices and \( \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \) denotes edges. Assume that there are \( k \) points \( p_j = \{p_{j(1)}, \ldots , p_{j(k)}\} \) close to point \( p_i \in \mathcal{P} \), which include directed edges \( (i, j(1)), \ldots , (i, j(k)) \). The edge features are defined as \( \mathcal{E}_{ij} = h_\theta(p_i, p_j) \), where \( h_\theta : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^F \) is a nonlinear function constructed by some learnable parameters \( \theta \). In detail, EdgeConv adopts an asymmetric edge function \( h_\theta(p_i, p_j) = \tilde{h}_\theta(p_i, p_j - p_i) \) where the centers of the patch point set \( p_i \) get the global geometric information and \( p_j - p_i \) capture the local neighborhood point information. Thus, the feature is denoted as shown in Equation (1).

\[
e_{ijm} = \text{ReLU}(\theta_m \cdot (p_j - p_i) + \mu_m \cdot p_i)
\]

where \( \theta \) is a set of learnable parameters. Finally, EdgeConv adopts a shared multilayer perceptron with ReLU to symmetrically aggregate a channel. Then, the max-pooling of edge feature is defined as \( p_{im} = \max_{j \in \mathcal{E}(i)} e_{ijm} \).

B. POINT GENERATION NETWORK
The point generation network aims at generating a 3D object with basic shapes and sparse points from a single-view image. As shown in Fig.2, we build this architecture from the core idea of U-net [28], which transfers the low- and high-resolution features in encoder part to decoder part. More specifically, the network can capture more geometric and texture feature from pixel-level information in encoder part and regress the localization information of points through two different ways in decoder part. For the encoder, the low- and high-resolution features \( f \) of image is extracted by the VGG-like encoder \( E_g(I) \) which removes the pooling layer between each convolution block. For the decoder, different resolution features are fused and sent into two branches to predict a fundamental point set \( S_1 = DC(f) + FC(f) \). Here, the FC and DC branches are expected to predict different part of the object. FC branch focuses on the corners and edges of objects with less points (240 pts) by stacking the fully connected layers. Rather than using only the low resolution and depth features, we flatten and combine the features from different layers (conv2, conv3, conv4, conv5). At the end of the FC branch, we add them together and output 240*3 units as 240 points by stacking the final fully connected layer activation. DC branch focuses on the body of object with majority points (768 pts) through multiple deconvolution blocks. The skip connection adds-in Deconv6, Deconv5, and Deconv4. Fig.3 is the visualization of the DC’s and FC’s output. It clearly shows that the DC branch generates the main part of the object, whereas the FC branch generates the remainder. Finally, a

![FIGURE 3. The visualization of the FC and DC output. The DC branch generates the body of the object and the FC branch generates the rest of the details.](image-url)
fundamental point cloud is composed from the output of the FC branch and DC branch.

C. POINT DEFORMATION NETWORK
The goal of the point deformation network is to make primary point cloud (spare and rough) finer and denser while maintaining its geometric structure. More specifically, the primary point cloud $S_1$ is supplemented and reordered based on the feature of the primary point cloud and projection loss (see in section D) to final point cloud $S_2$. The feature of primary point cloud provides a rough geometrical structure information to maintain the object shape. The projection loss constrains the location of reorganization points in point deformation network.

As shown in Fig. 4, a point-based CNN is used to extract features of primary point cloud (1024 points), which is leveraged by deformation to progressively restructure a rough point cloud into a dense point-based object (2048 points). More specifically, the encoder $E_g(S_1)$ is consisted of five edge convolution blocks with multilayer perceptron layers (64, 64, 128, 256) and the k-nearest neighbors (k is 20). It captures both global and local latent information from the primary point cloud $S_1$. The feature of primary point cloud $z \in E_g(S_1)$ is concatenated from each layer of encoder, which fuses different space information and preserve the prime geometry information of $S_1$. Then, $z$ was fed into $D_g(z)$, which is a two-level parallel network of two convolution layers, where the weights of each level are independent, including a 256-dimension feature vector of 1024 points. Finally, we concatenated them together to get a feature map of 2048 points. Compared to latent information encoding from a 2D image, the latent information encoding from a point cloud does not rely on texture information, but location information. By restructuring the point set, we tried to put each point in a more reasonable position so that the output point cloud is more dense and evenly distributed.

D. LOSS FUNCTION
To make the CPSGN trainable, we defined three kinds of objective functions to constrain the quality of the predicted object. We adopted EMD [32] to constrain the distribution of primary point cloud $S_1$; the CD to regress the location of the point close to its correct position similarly to ground truth; and proposed a projection loss to focus on the geometry and reduce outliers.

1) EARTH MOVER’S DISTANCES LOSS
EMD measures the distance between two different point sets. For primary point cloud $S_1$, EMD calculates the minimum distance to ground truth point cloud $S_G$ over all possible arrangements of correspondence. More formally, given two sets of points $S_1$ and $S_G$, EMD is defined as shown in Equation (2).

$$d_{\text{EMD}}(S_1, S_G) = \min_{\theta: S_1 \rightarrow S_G} \sum_{p \in S_1}||p - \theta(p)||_2$$

where $\emptyset: S_1 \rightarrow S_G$ is a bijection. In this instance, the EMD loss function is defined by the standard EMD function $l_{\text{EMD}} = d_{\text{EMD}}$.

2) CHAMFER DISTANCES LOSS
CD is widely applied to 3D reconstruction research, which confirms the pairwise relationship based on suboptimal matching. Given a point set, the CD distance sums the squared distance between each point and the nearest neighbor in ground truth. Considering that the CPSGN is a cascade network that produces two different sizes of point clouds, we set two CD loss covering two parts according to the standard distance function. For the point generation network, we apply standard CD function as a loss function. Formally, given two sets of points $S_1$ and $S_G$, the CD loss is defined as shown in Equation (3).

$$l_{\text{CD}}(S_1, S_G) = \sum_{p_1 \in S_1} \min_{p_c \in S_G} ||p_1 - p_c||_2^2 + \sum_{p_c \in S_G} \min_{p_1 \in S_1} ||p_1 - p_c||_2^2$$

For the point deformation network, we apply the CD with L1 regularization to measure the difference between the final object and ground truth that the overall point cloud offsets to force the network to deform the template as little as possible.
Formally, given two sets of points \( S_2 \) and \( S_G \), the CD loss is defined as shown in Equation (4).

\[
l_{CD}(S_2, S_G) = \sum_{p_i \in S_2} \min_{p_c \in S_G} ||p_i - p_c|| + \sum_{p_c \in S_G} \min_{p_i \in S_2} ||p_i - p_c|| \quad (4)
\]

3) PROJECTION LOSS
Both CD loss and EMD loss focus on optimizing the location of points in a point set, yet they cannot guarantee the predicted point set’s geometry to be similar to the ground truth. Following this line of thought, we propose a projection loss that measures geometry inconsistencies between predicted point set \( S_2 \) and ground truth \( S_G \). Specifically, we simultaneously project \( S_2 \) and \( S_G \) toward 2D image planes from different camera angles so that 2D projection can preserve shape information as much as possible.

To project 3D points, we applied perspective projection in this section. In detail, we defined a point \( d_{x,y,z} \) as the transformation of point \( p_{x,y,z} \) with the 3D world coordinate to the camera coordinates system using Equation (5).

\[
\begin{bmatrix}
    1 & 0 & 0 \\
    0 & \cos(\theta_x) & \sin(\theta_x) \\
    0 & -\sin(\theta_x) & \cos(\theta_x)
\end{bmatrix}
\begin{bmatrix}
    \cos(\theta_y) & 0 & -\sin(\theta_y) \\
    0 & 1 & 0 \\
    \sin(\theta_y) & 0 & \cos(\theta_y)
\end{bmatrix}
\begin{bmatrix}
    \cos(\theta_z) & \sin(\theta_z) & 0 \\
    -\sin(\theta_z) & \cos(\theta_z) & 0 \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    p_x \\
    p_y \\
    p_z
\end{bmatrix}
\]

where \( c \) is the camera position, and \( \theta \) is the orientation of the camera. Then we applied projection of the transformed point \( d \) onto a two-dimensional plane by the following Equation (6).

\[
\begin{align*}
    l_x &= (d_x - e_x)(e_z/d_z) \\
    l_y &= (d_y - e_y)(e_z/d_z)
\end{align*}
\]

where \( l_x \) and \( l_y \) are the projection coordinates.

As shown in Fig. 5, this pair of projections on the feature map is similar to the mask in the segmentation problem. Therefore, we adopted dice-confidence loss to measure the overlap between predicted projection and ground-truth projection. First, we defined the projection of point cloud \( S \) and \( S_G \), which are denoted as \( I \) and \( I_G \), respectively. The dice loss is defined as shown in Equation (7).

\[
l_{dice}(I, I_G) = 1 - 2 \frac{l_c}{l_I + l_G} \quad (7)
\]

where \( l_c \) means the intersection of the projection image \( I \) and \( I_G \). We found that only using dice confidence distance is unstable during the training. Thus, the L2 distance is added, and the projection loss function is defined as shown in Equation (8).

\[
l_{proj}(I_n, I_G) = \sum_n \|I_n - I_G\|_2^2 + \beta * l_{dice}(I_n, I_G) \quad (8)
\]

where \( \alpha \) is the different camera angles for projection, \( n = S_1 \) or \( S_2 \) is the corresponding point cloud, and \( \beta \) is a factor to control the projection of the sparse and dense point cloud.

4) FINAL LOSS
The final loss function consists of the losses of the point generation network and the point deformation network, as shown in Equation (9).

\[
L_{CP-SGN} = L_{gen} + L_{def} \quad (9)
\]

For the generation network, we adopt the EMD loss as main loss and standard CD loss as auxiliary loss, as shown in Equation (10),

\[
L_{gen} = w_1 * l_{CD}(S_v, S_c) + l_{EMD}(S_v, S_c) \quad (10)
\]

where \( w_1 \) is a factor to balance the two losses.
For the deformation network, we adopt the CD in L1 regularization as main loss and projection loss as auxiliary loss, as shown in Equation (11),

\[ L_{def} = l_{cd}(S_2, S_G) + w_2 * l_{proj}(I_2, I_G) \]  

(11)

where \( w_1 \) is a factor to balance the two losses and \( l_2 \) and \( l_G \) are projections of point cloud \( S_2 \) and ground truth \( S_G \), respectively.

IV. EXPERIMENT

In this section, we first introduce the dataset, implementation details, and evaluation metrics details. Second, we compare different representation methods on three evaluation metrics. Thirdly, we focus on comparing the reconstructed details of the point-based objects.

A. DATASET

To train and evaluate our networks, we use the dataset provided by Choy et al. [17], which includes rendering images of 50,000 models belonging to 13 object categories from ShapeNet [11]. The ShapeNet dataset collected 3D CAD models generated from the WordNet hierarchy with various camera viewpoints, camera intrinsic, and extrinsic metrics. To compare each method fairly, the ratio of training and testing dataset was set as the same as Choy et al. [3], which split each category into 80% for training and 20% for evaluation.

B. IMPLEMENTATION DETAILS

Our network is implemented with TensorFlow on an Nvidia Titan 1080. First, we initialize the camera intrinsic for projection loss, where the camera position is set to 2,1,2 on the x, y, z, axes, respectively, and focal length is set to 100. Second, we conduct the experiment with seven different angles for projection loss. We optimize the network using ADAM with a batch size of 12 and 150 total epochs. The initial learning rate is 0.0001 and then decrease after 15k iterations using natural exceptional decay. We use two different weight initialization methods for whole networks. The point generation network adopted He initialization [29] since ReLU activation is connected with each convolution layer. The point deformation network use Xavier initialization [30] to reduce the local minimization problem.

C. EVALUATION METRICS

Following previous research [3], we adopt EMD as standard evaluation metrics. EMD calculates the distance between a pair of point sets; the smaller the value, the more similar both are. Since the standard 3D reconstruction metric may not thoroughly reflect the geometry quality, we evaluated our network on F-score [31]. CD and EMD often capture occupancy or point-wise distance, which focus on the location of each point that is generated for an object. It has been reported in some articles that the CD value does not reflect the quality of model reconstruction well [6]. Thus, we do not adopt the CD score to evaluate performance in this work. By contrast, F-score calculates precision and recall by checking the percentage of points in prediction or ground truth that can find the nearest neighbor from the other within a certain threshold \( \tau \). In this case, precision and recall are defined as shown in Equation (9) and (10).

\[ \text{Precision}(S_P, S_G) = \sum \left( \sum_{p_1 \in S_P} \min_{p_2 \in S_G} \| p_1 - p_2 \|^2 < \tau \right) \]  

(9)
\[
\text{Recall}(S_P, S_G) = \sum \left( \sum_{p_1 \in S_P} \min_{p_2 \in S_G} \|p_1 - p_2\|_2^2 < \tau \right)
\]

where \(S_P\) and \(S_G\) are the predicted point set and ground truth point set, respectively; \(p_1\) and \(p_2\) are the coordinate values of each point in the corresponding point set; and \(\tau\) is the threshold set as 0.001.

D. COMPARISON WITH THE SOTA
For a comprehensive comparison, we evaluated our model from two aspects. First, we compared it with three representations on different metrics. It is difficult to measure the differences between an object generated by different representation methods. Thus, we selected three well-known studies of each representations as comparative experiments. More specifically, the voxel-based object generated by Choy et al. [10], the point-based object generated by Fan et al. [4], and the mesh-based object generated by (N3MR) [18]. All the models were trained with the same data using the same amount of time. Note that the labels corresponding to the dense model are 2048 points, and the other corresponding labels are 1024 points.

| Category    | 3D-R2N2 | N3MR | PSG | Ours (1024 pts) | Ours (2048 pts) |
|-------------|---------|------|-----|----------------|----------------|
| Airplane    | 0.606   | 7.498| 0.396| 0.738          | 0.744          |
| Bench       | 1.136   | 11.766| 1.113| 0.888          | 1.001          |
| Cabinet     | 2.520   | 17.062| 2.986| 0.835          | 0.836          |
| Car         | 1.670   | 11.641| 1.747| 0.771          | 0.872          |
| Cellphone   | 0.912   | 17.649| 1.019| 0.697          | 0.798          |
| Chair       | 1.466   | 11.809| 1.946| 0.894          | 0.895          |
| Couch       | 2.114   | 14.876| 2.207| 0.896          | 0.897          |
| Firearm     | 0.688   | 11.889| 0.397| 0.843          | 0.848          |
In this work, we build a cascade network to generate a more flexible and dense point cloud from single-view image. Compared with previous research [5], we generate more uniform and dense point cloud. However, there are still some details lost, especially in the edge of point cloud. For example, the leg of rarely seen chair is hard to reconstruct from the image. One possible reason is the limitation of the single-view image reconstruction. It cannot provide some detailed geometric information. The other possible reason comes from the single-view image reconstruction. It cannot provide some detailed geometric information.

In this part, every input image is standard, which has a higher geometric similarity to the training data. The PSG results seem to be comparable to ours when generated from these input images. However, the generated 3D objects of PSG lost some details around edges. For example, PSG failed to generate the legs and the tail part of the airplane correctly.

On the other hand, we input 2D images with special shapes. As shown in Figure 7 and Figure 8, for inputting images with special geometrics, PSG generates objects that only have a rough outline. By contrast, our model is more accurate and refined in generating the details of objects, such as the legs, the connections, and the hollows of the table, than PSG.

### Table I

| Category   | Mean  | 1.500 | 3.835 | 1.652 | 0.873 | 0.921 |
|------------|-------|-------|-------|-------|-------|-------|
| Lamp       | 1.424 | 14.741 | 1.222 | 0.981 | 1.190 |
| Monitor    | 1.667 | 14.097 | 1.891 | 0.984 | 0.986 |
| Speaker    | 2.732 | 16.720 | 3.490 | 1.037 | 1.083 |
| Table      | 1.641 | 12.842 | 2.121 | 0.801 | 0.798 |
| Watercraft | 0.935 | 11.425 | 0.945 | 0.989 | 1.030 |

Table I shows the EMD with different methods for 13 objects. Our method achieved the best mean score and shows better scores than the other methods for the most objects. When generating a dense point-based object, one step is upsampling the sparse model. Thus, the EMD value fluctuates to a certain extent.

Compared with the previous two metrics, the F-score can quantify the shape information of the model better. Table II summarizes the F-score for the 13 objects for various methods. The qualitative results from Table II show that our method is superior for the 13 objects. We achieved significantly better results than the others. Every object has a higher model completion with fewer missing parts in our method.

Fig. 6 shows the visual results compared with PSG. For the major comparison, PSG only applied Chamfer loss, acting like a regression loss, which gives too much freedom to the point cloud. In contrast, our model does not suffer from these issues by using cascade architecture, integration of perceptual features, and carefully defined losses during the training. Our result was not restricted by the resolution due to the limited memory budget and contains both a smooth continuous surface and local details.

We focused on comparing the visual effect of PSG because of the same representation. We generated both a sparse object (1024 points) and dense object (2048 points) for comparison.

### Table II

| Category   | 3D-R2N2 | N3MR | PSG | Ours 1024 pts | Ours 2048 pts |
|------------|---------|------|-----|---------------|---------------|
| Airplane   | 41.46   | 62.10| 68.20 | 68.05         | 40.20         |
| Bench      | 34.09   | 35.84| 49.29 | 39.15         | 47.37         |
| Cabinet    | 49.88   | 21.04| 39.93 | 38.30         | 44.07         |
| Car        | 37.80   | 36.66| 50.70 | 40.32         | 54.67         |
| Cellphone  | 42.31   | 27.96| 55.95 | 44.14         | 68.19         |
| Chair      | 40.22   | 30.25| 41.60 | 51.21         | 43.64         |
| Couch      | 40.01   | 25.04| 36.59 | 33.92         | 40.44         |
| Firearm    | 28.34   | 52.22| 69.96 | 89.16         | 67.35         |
| Lamp       | 32.35   | 27.97| 41.40 | 37.72         | 42.72         |
| Monitor    | 34.38   | 28.77| 40.53 | 39.80         | 46.79         |
| Speaker    | 45.30   | 19.46| 32.61 | 26.95         | 33.55         |
| Table      | 43.79   | 28.40| 53.44 | 54.56         | 65.89         |
| Watercraft | 37.10   | 43.71| 51.28 | 47.93         | 59.59         |
| Mean       | 39.00   | 33.80| 48.57 | 47.16         | 54.16         |

Table III shows the EMD and F-score with different loss function for 13 objects. After adding the projection loss, both EMD and F-score’s mean score are better. In the most categories, the network with the projection loss performs better on both the EMD and F-score. It confirms that projection loss has a positive influence on point cloud generation task.

### Table III

| Category   | Without projection loss | Add projection loss |
|------------|-------------------------|---------------------|
| EMD        | F-score                 | EMD                 | F-score             |
| Airplane   | 0.810                   | 0.744               | 0.900               |
| Bench      | 1.120                   | 1.001               | 47.37               |
| Cabinet    | 0.836                   | 0.836               | 44.07               |
| Car        | 0.918                   | 0.872               | 54.67               |
| Cellphone  | 0.972                   | 0.798               | 68.19               |
| Chair      | 0.985                   | 0.895               | 43.06               |
| Couch      | 0.897                   | 0.897               | 40.44               |
| Firearm    | 0.842                   | 0.848               | 67.35               |
| Lamp       | 1.081                   | 1.190               | 42.72               |
| Monitor    | 1.004                   | 1.083               | 33.55               |
| Speaker    | 1.103                   | 35.55               |
| Table      | 0.810                   | 0.798               | 65.89               |
| Watercraft | 0.992                   | 1.030               | 59.59               |
| Mean       | 0.951                   | 0.921               | 54.16               |

V. DISCUSSION

In this work, we have presented a cascade network for single-view 3D object reconstruction. According to the coarse-to-fine strategy, the CPSGN includes a point generation part and a point
deformation part. The generation network aims to generate a sparse point-based object with a basic shape and provide it to the deformation part. The deformation part regularizes the basic object from a sparse point distribution to a dense point-based object. In addition, we proposed a projection loss to aid the network in obtaining more geometric information in reconstruction. We evaluated our CPSGN on a ShapeNet dataset. The results of the experiment and analysis show that the model trained by our CPSG achieves better performance than others.

ACKNOWLEDGMENT
This research was supported by Basic Science Research Program through the NRF of Korea funded by the Ministry of Education (GR 2019R1D1A3A03103736) and was also supported by “Leaders in Industry-university Cooperation +” Project by the Ministry of Education and National Research Foundation of Korea.

REFERENCES
[1] R. Khilar, S. Chitrakala and S. SelvamParvathy, "3D image reconstruction: Techniques, applications and challenges," 2013 International Conference on Optical Imaging Sensor and Security (ICOSS), Coimbatore, 2013, pp. 1-6.
[2] S. Qi, X. Ning, G. Yang, L. Zhang, P. Long, W. Cai, W. Li. “Review of multi-view 3D object recognition methods based on deep learning,” Displays, vol. 69, pp. 102053, 2021.
[3] H. Fan, H. Su, L. J. Guibas, “A point set generation network for 3d object reconstruction from a single image,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii, 2017, pp. 605-613.
[4] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, Y. G. Jiang, “Pixel2mesh: Generating 3d mesh models from single rgb images,” in Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 2018, pp. 52-67.
[5] A.E. Khazari, Y. Que, TL. Sung, HJ. Lee, “Deep Global Features for Point Cloud Alignment,” Sensors. 2020; 20(14):4032.
[6] P. Achlioptas, O. Diamanti, I. Mitliagkas, L. Guibas, “Learning Representations and Generative Models for 3D Point Clouds,” in arXiv: 1707.02392, 2017, [online] Available: https://arxiv.org/abs/1707.02392.
[7] A. Laurentini, "The visual hull concept for silhouette-based image understanding," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 2, pp. 150-162, Feb. 1994.
[8] K. Hämning, G. Peters, “The structure-from-motion reconstruction pipeline – a survey with focus on short image sequences,” Kybernetika, vol. 46, no. 5, pp. 926-937, 2010.
[9] J. Fuentes-Pacheco, J. Ruiz-Ascencio, J. M. Rendón-Mancha, “Visual simultaneous localization and mapping: a survey,” Artificial intelligence review, vol. 43, no. 1, pp. 55-81, 2015.
[10] A. Knapitsch, J. Park, Q. Zhou, V. Koltun, “Tanks and temples: benchmarking large-scale scene reconstruction,” ACM Transactions on Graphics, vol. 36, no. 4, pp. 1-13, 2017.

[11] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, F. Yu, “ShapeNet: An Information-Rich 3D Model Repository,” in arXiv, 2015, [online] Available: https://arxiv.org/abs/1512.03012

[12] D. P. Kingma, W. Max, “Stochastic gradient VB and the variational auto-encoder,” in Second International Conference on Learning Representations, ICLR. Vol. 19. 2014.

[13] J. Wu, C. Zhang, T. Xue, “Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling,” in neural information processing systems (NIPS), Barcelona Spain, 2016, pp. 82-90.

[14] M. Tatarchenko, S. R. Richter, R. Ranftl, Z. Li, V. Koltun, T. Brox, “What Do Single-view 3D Reconstruction Networks Learn?” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, 2019, pp. 3405-3414.

[15] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M. M. Bronstein, J. M. Solomon, “Dynamic graph cnn for learning on point clouds,” ACM Transactions on Graphics, vol. 38, no. 5, pp. 1-12, 2019.

[16] H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, “Multi-view convolutional neural networks for 3d shape recognition,” in Proceedings of the IEEE international conference on computer vision (ICCV), Santiago, Chile, 2015, pp. 45-953.

[17] C. B. Choy, D. Xu, J. Gwak, K. Chen, S. Savarese, “3d-z2n: A unified approach for single and multi-view 3d object reconstruction,” in European conference on computer vision (ECCV), Springer, Cham, 2016, pp. 628-644.

[18] J. Gwak, C. B. Choy, M. Chandraker, A. Garg, S. Savarese, “Weakly Supervised 3D Reconstruction with Adversarial Constraint,” in 2017 International Conference on 3D Vision (3DV), Qingdao, 2017, pp. 263-272.

[19] J. Wu, C. Zhang, T. Xue, “Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling,” in neural information processing systems (NIPS), Barcelona Spain, 2016, pp. 82-90.

[20] S. Tulsiani, A. A. Efros, J. Malik, “Multi-view consistency as supervisory signal for learning shape and pose prediction,” in Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR), Salt Lake City, UT, USA, 2018, pp. 2897-2905.

[21] J. K. Pontes, C. Kong, S. Sridharan, S. Lucey, A. Eriksson, C. Fookes, “Image2mesh: A learning framework for single image 3d reconstruction,” in Asian Conference on Computer Vision Springer, Cham, 2018, pp. 365-381.

[22] H. Kato, Y. Ushiku and T. Harada, “Neural 3D Mesh Renderer,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 3907-3916.

[23] L. Jiang, S. Shi, X. Qi, J. Jia, “Gal: Geometric adversarial loss for single-view 3d-object reconstruction,” in proceedings of the european conference on computer vision (ECCV), Munich, Germany, 2018, pp. 802-816.

[24] R. Q. Charles, H. Su, M. Kaichun and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 77-85.

[25] C. R. Qi, L. Yi, H. Su, L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in neural information processing systems (NIPS), 2017, pp. 5099-5108.

[26] R. Klokov and V. Lempitsky, "Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 863-872.

[27] J. Li, B. M. Chen and G. H. Lee, "SO-Net: Self-Organizing Network for Point Cloud Analysis," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, 2018, pp. 9397-9406.

[28] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, O. Ronneberger, “3D U-Net: learning dense volumetric segmentation from sparse annotation,” in International conference on medical image computing and computer-assisted intervention, Springer, Cham, 2016, pp. 424-432.

[29] K. He, X. Zhang, S. Ren and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 1026-1034.

[30] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in Proceedings of the thirteenth international conference on artificial intelligence and statistics 2010, pp. 249-256.

[31] K. He, X. Zhang, S. Ren and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 1026-1034.

[32] Y. Rubner, C. Tomasi, L. J. Guibas, “The earth mover’s distance as a metric for image retrieval,” International journal of computer vision, vol. 40, no. 2, pp. 99-121, 2000.

[33] S. Liu, D. B. Cooper, “Ray markov random fields for image-based 3d modeling: Model and efficient inference,” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 2010, pp. 1530-1537.

Chenmou Wu is currently a Ph. D. student in Computer Science and Engineering at Jeonbuk National University in Jeonju, South Korea. His research interests lie in image processing, artificial intelligence, 3D Reconstruction and pattern recognition. His more recent research focuses on point cloud generation and point cloud segmentation.

Hyo Jong Lee (M’91) received the B.S., M.S., and Ph. D. degrees in computer science from the University of Utah, USA, where he was involved in computer graphics, image processing, and parallel processing. He is currently a Professor of the Division of Computer Science and Engineering and the Director of the Center for Advanced Image and Information Technology, Jeonbuk National University, Jeonju, South Korea. His research interests include image processing, artificial intelligence, medical imaging, parallel algorithms, and brain science.