International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 41, No. 6, December, 2024, pp. 2935-2946
Journal homepage: http://iieta.org/journals/ts

3D Point Cloud Generation to Understand Real Object Structure via Graph Convolutional Networks

Hamid Ashfaq[1], Abdulwahab Alazeb[2], Sultan Almakdi[2], Mohammed S. Alshehri[2], Nouf Abdullah Almujally[3], Sard S. Rlotaibi[4], Asaad Algarni[5], Ahmad Jalal[1,6*]

[1] Department of Computer Science, Air University, E-9, Islamabad 44000, Pakistan
[2] Department of Computer Science, College of Computer Science and Information System, Najran University, Najran 55461, Saudi Arabia
[3] Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia
[4] Information Technology Department, King Saud University, Makkah 24382, Saudi Arabia
[5] Department of Computer Sciences, Faculty of Computing and Information Technology, Northern Border University, Rafha 91911, Saudi Arabia
[6] Department of Computer Science and Engineering, College of Informatics, Korea University, Seoul 02841, South Korea

Corresponding Author Email: ahmjal@yahoo.com

https://doi.org/10.18280/ts.410613

Received: 12 September 2023
Revised: 13 March 2024
Accepted: 12 July 2024
Available online: 31 December 2024

Keywords:
point cloud, 3D model reconstruction, generative adversarial network, graph convolution network, real object structure

ABSTRACT

Estimating and generating a three-dimensional (3D) model from a single image are challenging problems that have gained considerable attention from researchers in different fields of computer vision and artificial intelligence. Previously, there has been research work on single-angle and multi-view object use for 3D reconstruction. 3D data can be represented in many forms like meshes, voxels, and point clouds. This article presents 3D reconstruction using standard and state-of-the-art methods. Conventionally, to estimate the 3D many systems investigate multi-view images, stereo images, or object scanning with the support of additional sensors like Light Detection and Ranging (LiDAR) and depth sensors. The proposed semi-neural network system is the blend of neural network and also image processing filters and machine learning algorithms to extract features that have been used in the network. Three different types of features have been used in this paper that will help to estimate the 3D of the object from a single image. These features include semantic segmentation, depth of image, and surface normal. Semantic segmentation features have been extracted from the segmentation filter that has been exploited for extracting the object portion. Similarly, depth features have been used to estimate the object in the z-axis from NYUv2 dataset training using SENET-154 architecture. Finally surface normal features have been extracted based on estimated depth results using edge detection, and horizontal and vertical convolutional filters. Surface normal helps in determining the x, y and, z orientations of an object. The final representation of the object model has been in the form of a 3D point cloud. The resultant 3D point cloud has made it easy to analyze the model quality by points and distance representing intermodal and ground truth. In this article, three publicly available benchmark datasets have been used for system evaluation and experimental assessment including ShapeNetCore, ModelNet10 and ObjectNet3D datasets. The ShapeNetCore has archived an accuracy of 95.41% and chamfer distance of 0.00098, the ModelNet10 dataset has achieved an accuracy of 94.74% and chamfer distance of 0.00132 and finally, the ObjectNet3D dataset has achieved an accuracy of 95.53% and chamfer distance 0.00091. The results of many classes of the proposed system are outstanding at visualization as compared to standard methods.

1. INTRODUCTION

Nowadays, 3D perception has gained more importance in the vision domain. It is worth mentioning that the mainstream in the field of deep learning includes machine learning tasks, object detection, object recognition, object reconstruction, mapping of 2D information of an object in 3D, and so on. Human intelligence can easily map a 3D model of an object using CAD software and 3D development tools using its image. Extraction of the 3D model from a single RGB image is one of the difficult problems in the field of computer vision as most of the 3D information is lost during sampling [1] and the colour quantization process [2]. To estimate and reconstruct a 3D model from a single image, we use different encoders and deep-learning convolutional neural networks to get the features from the image. Furthermore, these features can be then later used to generate the 3D model in the form of a mesh, point cloud and voxel-based image. By using deep learning and parallel computing, the process of estimation, rendering and calculation of geometry and points are much easier and more efficient. The deep generative methods have not only done processing so easily on images but also on 3-dimensional complex object representation, segmentation [3], reconstruction, clustering and compression [4]. There are multiple systems for 3D point cloud generation like PSGN [5], RealPoint3D [6] and 3D-ReconstNet [7]. Most of these systems depend on end-to-end deep learning, as these systems require large datasets and a much longer time to train the model. Some systems depend on stereo imaging [8, 9] and multi-view [10, 11] to generate 3D of an object. Nowadays many types of sensors are also available like depth sensors [12, 13]. Most of the real-world data is available in single RGB image form. So, we need systems that are accurate in terms of estimation and generation of 3D using a single 2D image. Some 3D methods depend on single image archives and have good accuracies with end-to-end deep learning but their performance may drop when considering a training model based which is based on a large dataset upon reviewing the literature on image-based 3D reconstruction systems, we identified several shortcomings that served as the primary focus for our proposed system. Most of the previous systems are based on deep learning, and machine learning algorithms and the features details are missing. Some researchers worked on multi-view images, and depth information of the image to generate 3D models. Moreover, most of the work has been done by training the model on only a limited dataset and thus failed to recognize objects in an unseen environment. Furthermore, few studies have been based on generating the visible elevation of an object or scene. The proposed system has been based on feature extraction, machine learning algorithms and image processing techniques [14] that have saved the computational cost of neural networks.

This paper focuses on the structure of objects by converting 2D images to 3D using graph convolutional networks. The proposed system includes three main steps: First the pre-processing has been done to determine the projection of 3D in the form of 2D image [15]. Second, feature extraction for segmentation [16], depth [17] and surface normal [18] has been applied. Finally, extracted features have been used in the deep neural network to generate the 3D point cloud. The 3D model of 2D images has been visualized as:

1) The 3D model has been represented in meshes that contain the rich representations of object vertices and edges of a 2D model that result in the highly efficient mapping of the 2D image to the 3D model.

2) Voxels are one of the high geometric resolution 3D representations of 3D models [19]. The pixels have been taken by CMOS as a standard unit of image [20], and have been converted to dpi for printing purposes. Similarly, the voxel-based model has been converted to a point-based model for analysis.

3) Point Cloud is mostly saved in the standard Point Cloud file format. These files have information about the points and locations in sequence [21, 22]. It is a better representation than meshes and voxels because these points can easily be compared and we can also find the distance from one point to another to measure the quality and analysis of our model.

4) To test our method, many evaluations have been conducted. More specifically, simple distance and loss functions are mostly used in 3D-generated model evaluation. These distance and loss functions include chamfer distance (CD) [23] earth mover's distance (EMD) [24] and intersection over union (IoU) [25]. The quality of the 3D generated model has been measured concerning the ground truth, number of points in the point cloud, vertices and edges in mesh and number of voxels in the voxel-based 3D model. The proposed system has been tested on three state-of-the-art datasets: ShapeNetCore [26], Pascal3D [27] and ObjectNet3D [28].

The rest of the paper has been organized as follows. Section II presents a literature review of the existing methods. Section III discusses the proposed system of 3D point cloud generation. Section IV represents the experimental structure and analysis of three State-of-the-art datasets as compared to existing systems. Section V presents experimental settings and results. Finally, Section VI discuss the conclusion and recommendation for future work.

2. RELATED WORK

Many researchers have extensively worked on 3D reconstruction using a single image. Different methods mostly consisted of Deep Neural Networks (DNN), Encoder and Decoder of features, Graph Convolutional Networks (GCN) and semi-supervised methods to construct 3D models easily and efficiently.

2.1 Generation of 3D models using deep neural networks

DNN for 3D reconstruction has been widely deployed in extensive research. Hu et al. [29] used Principal Component Analysis (PCA) that used 3D point clouds to create its feature vector that helped in the reconstruction of point clouds of an object with its image. Similarly, the Depth feature is extracted using a 3D mesh with its image by Liu et al. [30]. Faster region-based convolutional neural networks (RCNN) by Ren et al. [31] used multi-scale feature extraction using an image pyramid for feature extraction. According to Han et al. [32] used deep learning for the generation of point clouds. These methods are known as variational auto-encoders (VAEs), adversarial auto-encoders (AAEs) and generative adversarial networks (GANs). Yu and Lee [33], Gadelha et al. [34] used VAEs for the generation of a 3D point cloud in their research. They used VGG-11 [35] and MRT-Encoding techniques. For the good compact representation, some AAEs are also used as discussed in study by Zamorski et al. [4].

2.2 3D Reconstruction using graph convolutional networks

With the development of parallel processing like DNN, GCN and Graphic Processing Units (GPUs), it's possible to work on 3D reconstruction-related work. GCN system is based on a Convolutional Neural Network (CNN). Yang et al. [36] present the mesh model of an object as composed of a set of edges and vertices, which resembles nodes and connections represented in a graph. So, mesh models can be generated using MGCN. Similarly, for the generation of other forms of 3D like point cloud and voxel-based 3D.

Zhang et al. [37] have proposed a method named scene graph convolutional networks that is very helpful in understanding the scene in 3D using a single image. It has 3 main nodes that help determine the layout, a bound box and object node that categories different objects in the scene and a relationship node in distances and 2D appearance of the object. A method using Nodeshuffle, based on PixelShuffle and image super-resolution techniques, has been employed to intercept multiscale features in the upsampling layer of the multiscale graph convolution layer [38, 39].

3. PROPOSED 3D GEOMETRIC GENERATION

The proposed system reconstructs 3D point clouds and 3D voxel-based models using a single image. Primarily, an image has been captured using a digital camera. The detected image has been then converted to an array of pixels using quantization and sampling. Each pixel consists of 3 channels (Red Green Blue) RGB. Most of the information has been lost in the process of digitizing the 3D data into 2D images. Our main goal is to estimate the lost information and reconstruct a 3D model of the object represented in the image. Three different features have been extracted from the image that helped in the estimation of the 3D model. These features have been categorized as depth, surface normal and intrinsic. Features that helped in the generation of different forms of 3D. Figure 1 shows the architecture of a 2D image to 3D point cloud generating system. Our proposed system has been based on feature extraction using image processing filters and CNN.

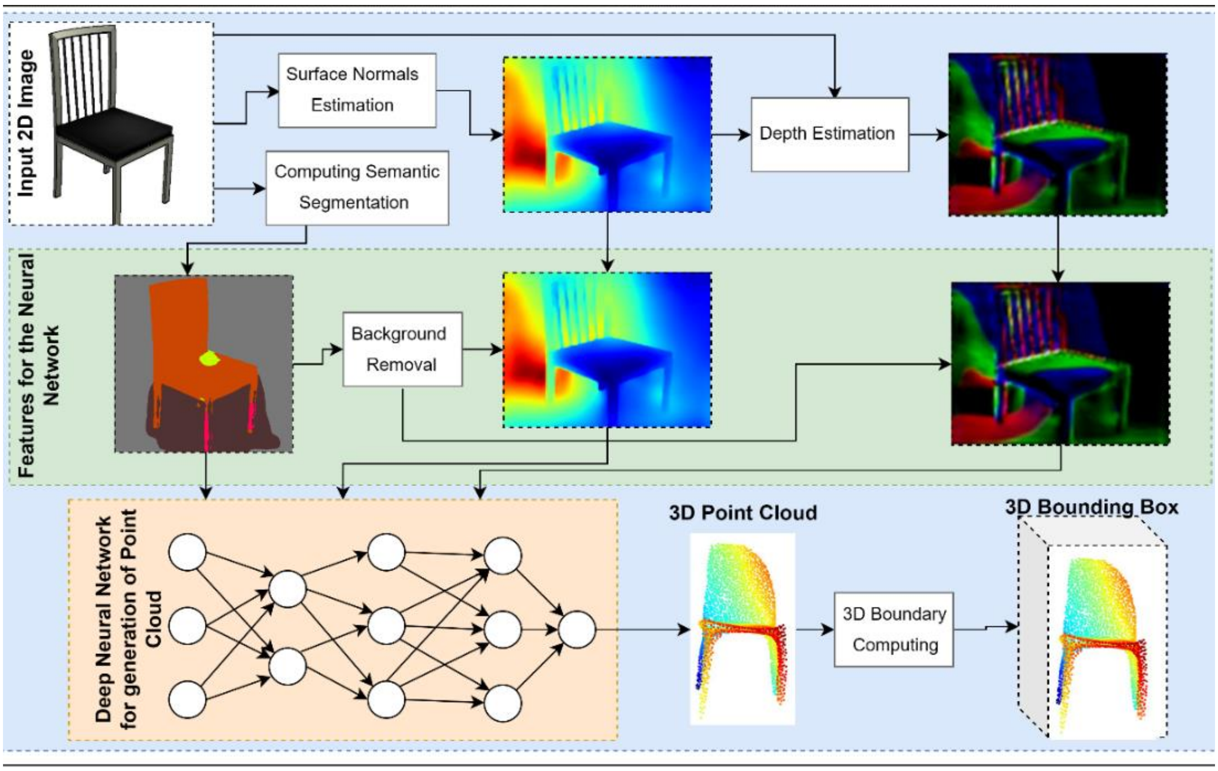

Figure 1. Flow architecture of the proposed 3D generating system

3.1 Data preprocessing and rendering 2D image

The proposed system has taken 3D information for training purposes to test the final result with the initial ground truth. Next, a 2D image has been rendered using 2D projection from the 3D models of data. According to Dhome et al. [40] using perspective view determines the projection of 3D in the form of 2D image. In this process, depth, orientation, and z-axis information are lost from 3D and as a result, get the 2D image we get from the RGB camera sensor. It is the perfect simulation of a camera. Using the 2D projection method, we already have a ground truth for the end to analyze the final generated 3d using our proposed system. The 2D projection from 3D space is shown in Figure 2.

$$line(a,b) = B_i b + \{C_i c * (A_i a - B_i b)\} \quad (1)$$

where, the line(a,b) is used to calculate the intersection point of the line between points a and b. a is the initial point of 3D space from there we start calculating on the other end of the 2D projection side we use point b. Ai, Bi and Ci are the components of the line.

$$(e, f, g) = (cx, cy, cz) \quad (2)$$

$$(u, v, 256) = (xc, yc, 256) \quad (3)$$

where, (e, f, g) points on the line where c exists. While, u=cx, v=cy are the new projection points if we look from point z=256/c. We can easily get the value of u and v using the value of c. Figure 3 shows the result of 2D projection.

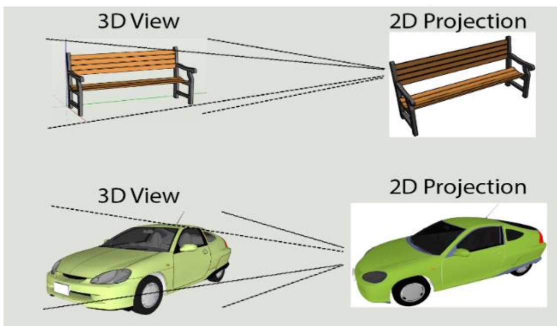

Figure 2. Graphical visualization of objects form 3D space to 2D using projection mechanism

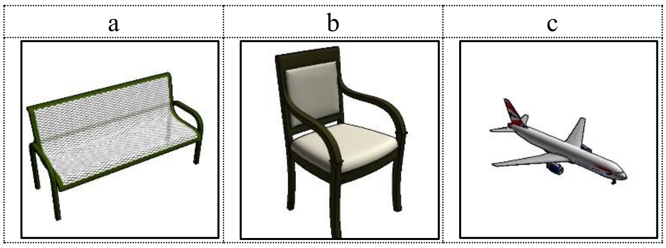

Figure 3. Rendered images results of 2D projection from objects source dataset

3.2 Extraction of semantic segmentation

The proposed method has been used for the extraction of semantic segmentation. In this proposed research we have applied scaling, normalization, mean, Standard deviation, transposition and a combination of many other mathematical functions to extract the semantic segmentation of the object from the image. Using this method first image has been cropped to get the square-shaped image. Second, scale the image intensity values to 0.0 to 0.1. Third, normalized the image using the mean and standard deviation of the image. Fourth, the transpose of the image was calculated. Further, the image has been segmented based on its intensity values. After segmentation get the smoother output according to the object. Argmax has been used to get the max value and SoftMax was used to further scale the value. The output result has been the segmented portion of the object and its background. That has been used in creating a mask for removing background and unwanted portions. Figure 4 shows the semantic segmentation results of experimented data.

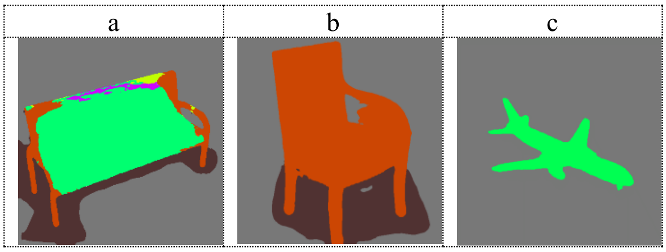

Figure 4. Semantic segmentation results of object colour partitioning in 2D images

$$Vec\_T(A) = [a_{ij}]^T, \qquad A \in R^{nxm} \tag{4}$$

Resized the original image as our algorithm requirement. Then change the scale of intensity in the 0 to 1 range. Then some normalization using mean and standard deviation. Later, the shape of that matrix was adjusted by using, finding maxima and some activation functions. For depth estimation, the Squeeze-and-Excitation Network (SENET) [41] was used for training and the model was then used for getting the depth of the image using its RGB inputs (See Algorithm (1)).

Algorithm 1: Semantic Segmentation using Torch library and normalization

Input: Original_Image = RGB_Image,
Output: S_Seg_Img = Semantic Segmentation Image

Resize: [512×512] Image ← Resize(Image)
Scale: [0.0 to 1.0] Image ← Image.float()/255
Normalize:
Mean = Mean[Mean_Value_of_image]
SD = list[SD_Value_of_image]
Image←T.Normalize((Mean),(SD))(image)
Foreach i in Image
i←(i-min(i))/(max(i)-min(i))
Add batch size
image = Transpose of image to change the sequence
image.shape
Shape adjustment and relative scaling
argmax: argmax f(x) where f(x)>= in all subsets of X
softmax: softmax [0 to 1]
Depth_Map_Module:
Dataset: NYU v2 (Depth Dataset),
Depth Model: After training using NYU v2(Depth Dataset)
CNN Architecture: SENet-154
Module Forward:
avg pooling: def avgpool($H_{feature}$, $W_{feature}$, f, s, Ch):
$H_{feature}$-f+1)/s*($W_{feature}$-f+1)/s*Ch
Full convolution:def full_conv(x,kernal):
{H(x)= f(x)* g(x)
Relu: Relu:max(0,x):
return [0 to max]
Full convolution repeat
Sigmoid: def Sigmoid(x):
return:e**x/(e**x+1)
return S_Seg_Image, Depth_Image

3.3 Extracting depth using SENET-154

This paper utilized the proposed SENET-154 model [42] SENET architecture consists of 3 major blocks. The first block squeezes a channel descriptor for global spatial information. To achieve this, global average pooling has been used that generate channel-wise statistics. The descriptor value has been calculated using shrinking of spatial dimension information. The second block named, excitation which was used for fully captured channel-wise dependencies. It is also called adaptive recalibration, this functionality has two main criteria: one, the function must be flexible (means capable of learning a nonlinear interaction between channels) and next, it must learn non-mutually exclusive multi-channel relationships. Block three, apply AlexNet [43] and VGGNet [44]. There are some more usable variants. NYUv2 depth dataset [45] has been used that consists of a single back-and-forth sweep. The trajectory of this dataset represents the scanned motion agents for better knowledge of the scene. In the current proposed approach, SENet-154 has been used for the training of images with their respective depth. By using this network, we can compute the depth of an object from its image as shown in Figure 5. We further use the depth feature for the estimation of surface normal and also used in neural networks for 3D point cloud generation.

$$X_{Channel} = F_{seq} = \Big(\sum_{i}^{H} \sum_{j}^{W} f(i,j)\Big)/(HXW) \tag{5}$$

where, F_{seq} is the squeeze function on image f with W rows and H columns. This descriptor shrinks the spatial domain data into a single channel that is used by the next layer.

$$F_{exc} = z\sigma\big(g(x_{Channel}, W)\big) = \sigma\big(W_i \delta(x_{Channel} W_{i-1})\big) \tag{6}$$

where, F_{exc} is the excitation layer function. We apply the sigmoid σ function on the result of δ function on $x_{channel}$ we got from the squeeze layer.

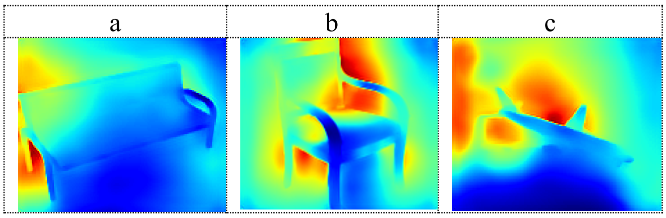

Figure 5. Using SENET-154 results of depth

3.4 Calculating surface normal

Semantic environment understanding is one of the toughest parts of using images. According to Klasing et al. [46], robust object recognition in 3D is the very important part. 3D object recognition algorithms need geometric segmentation and extraction. So, surface normal vectors turn out to be one of the most fundamental features. Plane SVE surface normal estimator [47] is one of the simplest ways of estimation point p=[x,y,z] T in camera coordinates in the local plane. In the current proposed system simple and fast method has been used that was based on image filters to compute surface normal from RGB and calculated depth of image. The following method has 3 main filters. First, horizontal gradient filter. Second vertical gradient filter. Both above filters get the horizontal and vertical edges of an object. Then lastly, the mean/ median filter in our case, the Gaussian filter has been used with 3×3 kernel size and stride rate of 1. This method highlighted the surface orientation feature from images. Then these features have been used in 3D estimation. The results of surface normal are shown in Figure 6.

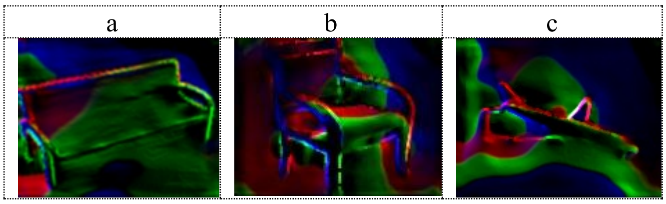

Figure 6. Surface normal results on selected classes a) bench, b) chair and c) airplane

$$Gx = [[1,1,1],[0,0,0],[-1,-1,-1]]^T \sum x \tag{7}$$

$$Gy = [[[-1,0,1],[-1,0,1],[-1,0,1]]]^T \sum y \tag{8}$$

$$RMS = \sqrt{Gx + Gy} \tag{9}$$

where, *Gx* is the horizontal gradient filter and *Gy* is the vertical gradient filter. Using Root Mean Square (RMS) value to combine them.

The final result was obtained after the RMS applied the blur filter using mean or Gaussian filter, and a slight difference was obtained using the blur filter. In our proposed method we used Gaussian filter window size 3×3.

3.5 Graph-based fusion

Represent the relationships between features from different modalities as a graph, where nodes represent features and edges represent relationships. Applied graph convolutional networks (GCNs) to perform fusion by aggregating information from neighbouring nodes in the graph.

$$H^{l+1} = \sigma(D^{-1/2}AD^{-1/2}H^lW^l) \tag{10}$$

where,

H^l is the feature matrix at the l-th layer, where each row corresponds to the features of a node in the graph. W^l is the weight matrix for the l-th layer.

A is the adjacency matrix of the graph, which represents the relationships between nodes. It may be normalized, such as by row-wise normalization to represent the strength of connections.

D is the degree matrix of the graph, a diagonal matrix where D_{ii} is the sum of the elements of row i of A.

σ is the activation function, sigmoid.

3.6 3D point cloud generation using GCN and 3D bounding box computation

Point cloud generation is one of the most vital modules of our proposed method, which is based on GCN to reconstruct a 3D point cloud. The point cloud is the representation of 3D using the grouping of points within 3D space, each point in the point cloud is just like a node in a graph so it can be reconstructed using a graphical convolutional network [48, 49]. In this step, a 3D shape has been constructed using GCN with the help of convolutional layers on the same result refined form of 3D generated, the computation depends on number of points required in each 3D point cloud. If each point of point cloud is connected with its neighboring points. These connections between points will represent edges and it will form a 3D mesh. Also point cloud is easy for analyzing the 3D object in space. Because, when comparing points of the predicted 3D object point cloud with respect to the ground truth of object model points.

$$G(X) = f(V_n, E_n) \tag{11}$$

where, $G(X)$ is the Graph of X which consists of vertices V and Edges E.

$$H^i = F^i * N \tag{12}$$

where, H^i is the hidden layers. F^i represents a number of features and N is the number of features in each hidden layer.

$$L^i = f(L^i - 1, X) \tag{13}$$

where, L^i is the number of layers in network on a specific input matrix X. On initialisation L0 is the initial input matrix X0.

Algorithm 2: GCN and point cloud

Input: Depth, SurfaceNormal.
Output: 3d_Point_Cloud =.pcd file
Training Dataset: ShapeNet,
3D_Point_Cloud_Model:Function

```
GCN_3D_PointCloud(dep [], SN[], no_of_.points)
  { x=64;n=0
    While: exit condition n>4
    {
      Convolve2D(dep[3*3],x)
      Convolve2D(dep[3*3],x)
      If: x<128
         Convolve2D(SN[3*3],x)
      Pooling(2*2)
```

If: x<512
x=x*2
n++}

}
CNN Architecture: **GCN_3D_PointCloud** (depth, Surface_Normals, no_of_.points)
return 3D_PointCloud

According to Shu et al. [50], 3D point cloud is generated using a generative adversarial network (GAN) known as tree-GAN. Based on Tree GCN that performs 3D graph convolutions in a tree the information has been boosted using this method. The tree has n number of branches that use graph convolution at each layer and pass to the next layer. The tree expends from set {z} to {z, p1, p2, …pn-1}. Figure 7 shows the visual representation of the deep learning method based on GCN that has been used for generating a 3D point cloud.

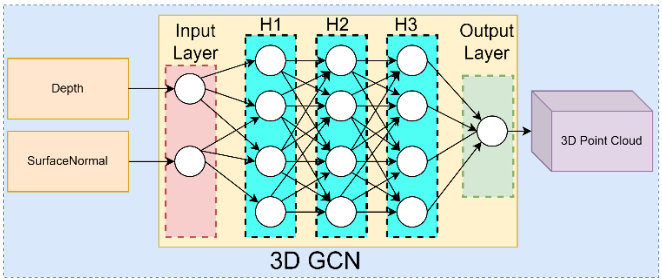

Figure 7. Visual representation of 2D projection from 3D space

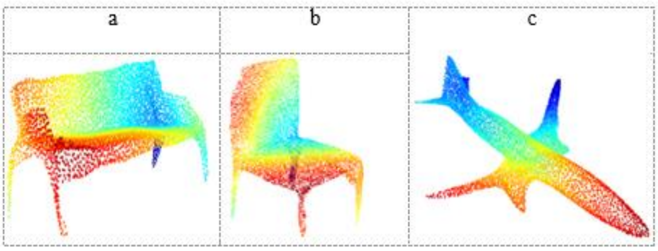

Figure 8. Using GCN to generate 3D point cloud of the class bench, chair and airplane

GCN architecture has been used to generate 3D models in the form of the point cloud. Each model has been rendered using 2000 points to create a refined 3D point cloud. Diverse numbers of points have been used to generate a 3D model. The quality of the reconstructed model depends on the quantity of points. So, when we increase the number of points it will increase the scalability of the model but needs more time to render and also takes more memory. The created point cloud has been then further useful in the development of 3D games and VR/AR. Figure 8 shows 3D point clout generated using depth and surface normal features that have been calculated using Extracting Depth using SENET-154 and Calculating Surface Normal.

4. EXPERIMENTAL SETTINGS AND ANALYSIS

All experiments have been performed on google-collaborator equipped with Intel Xeon 2.3GHz processing power and 12GB RAM and Nvidia K80/T4 GPU and a laptop with the following specifications Intel Core i5-4th Gen 2.70Hz processing power, 12GB RAM, x64 Bit Windows 10 and PyCharm 2020 tool. The experiment has been divided into 2 sections. In the first section, the 3D point cloud of object generation performance has been evaluated. The accuracy of the proposed method has been evaluated using a confusion matrix and precision, sensitivity, specificity and F1 scores with state-of-the-art (SOTA) methods. In the second section, a distance matrix has been used to analyse the 3D model. True relative distance has been measured using Chamfer distance and Euclidean distance. These distances measure the orientation of points in the point cloud according to ground truth.

4.1 Datasets description

The three datasets that have been used for experimentation including: The ShapeNetCore dataset, ModelNet-10 dataset and ObjectNet3D dataset. Details of each dataset given has been depicted in following subsection.

4.1.1 Shapenetcore dataset

In the benchmark dataset named ShapeNetCore. It is the subset of full dataset ShapeNet [51]. It has 55 common object categories that consist of 51300 unique models of 3D objects. The 7 categories have been used for our research purpose including: airplane, bed, bench, car, chair, sofa and table. Few categories of ShapeNetCore dataset have been mentioned. In Figure 9, the models are created using CAD software and arranged using WordNet dataset [52].

Figure 9. ShapeNetCore dataset CAD sample images of selected classes

4.1.2 ModelNet-10 dataset

Figure 10. Pascal3D dataset CAD sample images of selected classes

ModelNet-10 [26] has been composed of 3D CAD models of inhouse objects. ModelNet-10 includes 10 classes: bathtub, bed, chair, desk, dresser, monitor, night stand, sofa, table and toilet. Some samples from the ModelNet-10 dataset are shown in Figure 10. This dataset has been compiled from the ShapeNet dataset.

4.1.3 ObjectNet3D dataset

The ObjectNet3D [28] dataset consists of 100 categories, 90,127 images and in these images 201,888 objects and 44,147 3D shapes. We selected 10 classes for testing the following objects: plane, bed, car, chair, dining table, sofa, and, rifle. The reason for using this dataset is all 3D shapes have been aligned concerning their 2D images. Figure 11 depicts a few classes from the ObjectNet3D dataset. Each 3D CAD model has been aligned with its 2D image very useful for 3D pose recognition and estimating the 3D shape of object retrieval from the image.

Figure 11. Illustration of ObjectNet3D models dataset

5. PERFORMANCE EVALUATION

The performance of our 3D point cloud generation system has been evaluated using the quality of the model having the number of points in the point cloud, chamfer distance (CD) and earth mover's distance (EMD) over ShapeNetCore, Pascal3D and ObjectNet3D datasets.

Table 1. Chamfer distance, edge loss, normal loss and Laplacian loss on ShapeNet

Objects	Chamfer Distance	Edge Loss	Normal Loss	Laplacian Loss
Airplane	0.0006	0.0033	0.0358	0.0051
Bed	0.0014	0.0039	0.0149	0.0037
Bench	0.0006	0.0022	0.0156	0.0031
Car	0.0009	0.0021	0.0084	0.0023
Chair	0.0009	0.0025	0.0158	0.0034
Sofa	0.0018	0.0039	0.0163	0.0043
Table	0.0007	0.0035	0.185	0.0038
Mean	0.00098	0.00305	0.041686	0.003671

The numerical experimentation of 3D reconstruction and four loss functions has been calculated on the Benchmark dataset for result analysis. First, chamfer distance showed the orientation and directional loss using [23]. Second, edge loss has been computed using the conversion of mesh edge length regularization loss average. Third, normal loss has been computed using consistency between each pair of neighbours and fourth Laplacian loss has been calculated using difference in each batch using Laplacian smoothing. Table 1 represents the compiled result of the final loss after 2000 iterations. Each object's distance has been calculated from a 3D ellipsoidal point cloud with 2048 points on the ShapeNet Dataset. Similarly, Table 2 depicts the results on ModelNet and Table 3 on the ObjectNet3D dataset.

Table 2. Chamfer distance, edge loss, normal loss and Laplacian loss on ModelNet10

Objects	Chamfer Distance	Edge Loss	Normal Loss	Laplacian Loss
Bathtub	0.0022	0.0033	0.0282	0.0051
Bed	0.0005	0.0024	0.0083	0.0022
Chair	0.0009	0.0037	0.0325	0.0061
Desk	0.0008	0.0032	0.0207	0.0044
Dresser	0.0018	0.0039	0.0162	0.0045
Monitor	0.0009	0.0022	0.0168	0.0033
Night_Stand	0.0009	0.003	0.0147	0.0032
Sofa	0.0008	0.0023	0.014	0.003
Table	0.0009	0.0037	0.0172	0.0047
Toilet	0.0035	0.00566	0.0155	0.0055
Mean	0.00132	0.00333	0.01841	0.0042

Table 3. Chamfer distance, edge loss, normal loss and Laplacian loss on ObjectNet3D

Objects	Chamfer Distance	Edge Loss	Normal Loss	Laplacian Loss
Bed	0.0016	0.0037	0.0194	0.0042
Car	0.0005	0.0019	0.011	0.0023
Chair	0.0006	0.0024	0.016	0.0034
Dining Table	0.0008	0.0047	0.0205	0.0052
Plane	0.0006	0.0027	0.0382	0.0048
Rifle	0.0003	0.0014	0.008	0.0016
Sofa	0.0020	0.0043	0.0209	0.0050
Mean	0.00091	0.00301	0.01914	0.00378

Table 4. Accuracy, precision, recall and F1-Score on ShapeNet

Objects	Accuracy (%)	Precision	Recall	F1-Score
Airplane	95.208	0.337	0.814	0.477
Bed	94.234	0.977	0.956	0.967
Bench	96.686	0.587	0.784	0.671
Car	96.686	0.395	0.683	0.500
Chair	96.074	0.767	0.936	0.843
Sofa	93.722	0.855	0.831	0.857
Table	95.264	0.992	0.814	0.894
Mean	95.411	0.701	0.831	0.744

Table 5. Accuracy, precision, recall and F1-Score on ModelNet10

Objects	Accuracy (%)	Precision	Recall	F1-Score
Bathtub	93.783	0.969	0.9626	0.966
Bed	96.794	0.909	0.906	0.907
Chair	94.415	0.636	0.935	0.757
Desk	95.298	0.794	0.933	0.857
Dresser	93.701	0.326	0.511	0.398
Monitor	96.409	0.782	0.661	0.716
Night_Stand	95.585	0.919	0.732	0.815
Sofa	96.397	0.855	0.809	0.831
Table	94.733	0.770	0.552	0.643
Toilet	90.268	0.481	0.614	0.540
Mean	94.738	0.744	0.761	0.743

Tables 4-6 present the accuracy, precision, recall and F1-score on different classes of datasets. For accuracy, precision, recall and F1-Score point-by-point difference in the local points has been calculated and compared the predicted model with its ground truth model. Also, distance from the ground truth has been computed. Then on the basis of these distance values accuracy, precision, recall and F1-Score values have been computed. Accuracy has been measured using the difference between predicted points and ground truth points. Precision, and recall values has shown the distance from ground truth to predict. F1-Score has shown the harmonic mean of precision and recall values.

Table 6. Accuracy, precision, recall and F1-Score on ObjectNet3D

Objects	Accuracy (%)	Precision	Recall	F1-Score
Bed	94.155	0.813	0.905	0.856
Car	97.251	0.801	0.823	0.856
Chair	96.533	0.227	0.8172	0.356
Dining Table	93.759	0.860	0.677	0.758
Plane	95.897	0.397	0.704	0.508
Rifle	98.045	0.718	0.652	0.683
Sofa	93.063	0.818	0.621	0.706
Mean	95.529	0.662	0.743	0.675

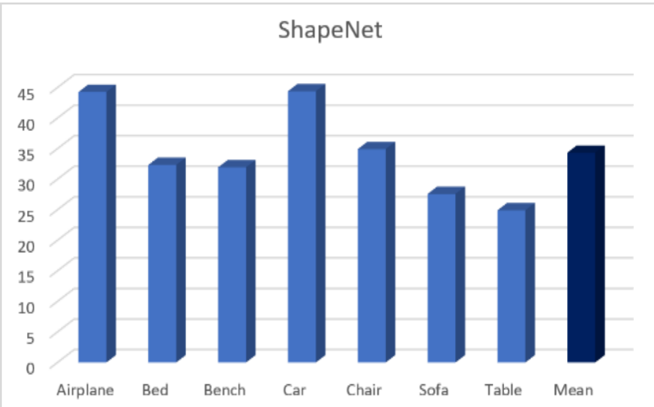

Figure 12. EMD graph of ShapeNet dataset

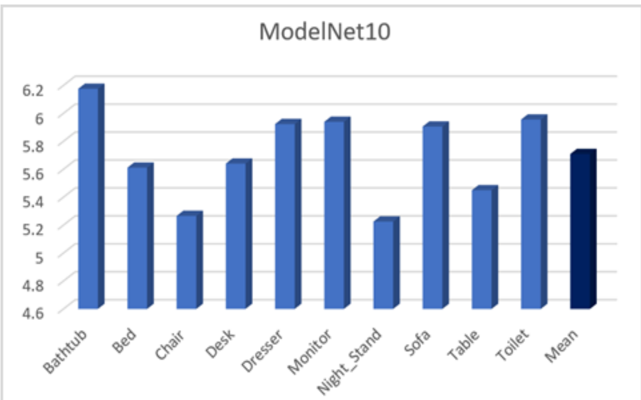

Figure 13. EMD graph ModelNet10 dataset

Figure 12 shows the EMD value graph of the ShapeNet dataset. The mean value is 34.32 for the selected sample classes. On our test results. The ground truth and predicted 3D model orientation have shown different results accordingly. EMD depends on the orientation of a 3D object because it is the distance between predicted and actual probability distribution over a region. To compute the EMD value. The methods contain Wasserstein distance [53] also known as sink horn distance is calculated from the 3D tensor of the predicted point-cloud and ground-truth point-cloud.

EMD has been evaluated on the ModelNet10 dataset as shown in Figure 13. The average value is 5.70 has been achieved as compared to the ShapeNet and ObjectNet datasets average value is very different because this dataset has been aligned with the predicted model.

The EMD graph of ObjectNet3D is shown in Figure 14. The ObjectNet3D has achieved close results to the ShapeNet dataset. There is another reason the point-cloud EMD value is much greater when the whole point cloud is inverted it shows a combined difference of all the points.

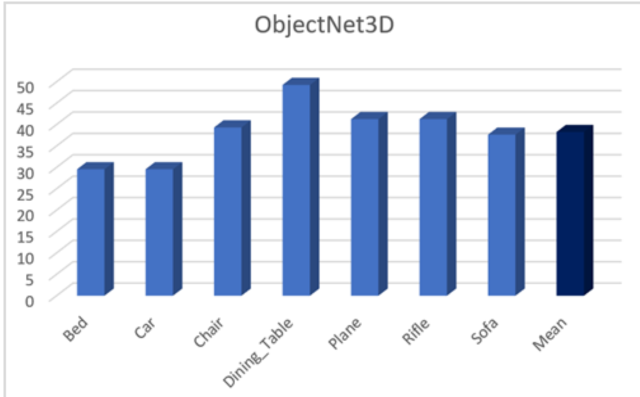

Figure 14. EMD graph of ObjectNet3D dataset

Two approaches have been used to compare the performance of our proposed system. These are compared using CD and EMD. We compare our method with reported state-of-the-art 3D object generation networks PSGN [5], RealPoint3D [6] and 3D-ReconstNet [7]. For evaluation five categories of single images are selected: airplane, bench, car, chair and sofa. To make this comparison fairer our proposed model has been trained on 1024 points because RealPoint3D, PSGN and 3D-ReconstNet CD values have been represented with a similar number of points.

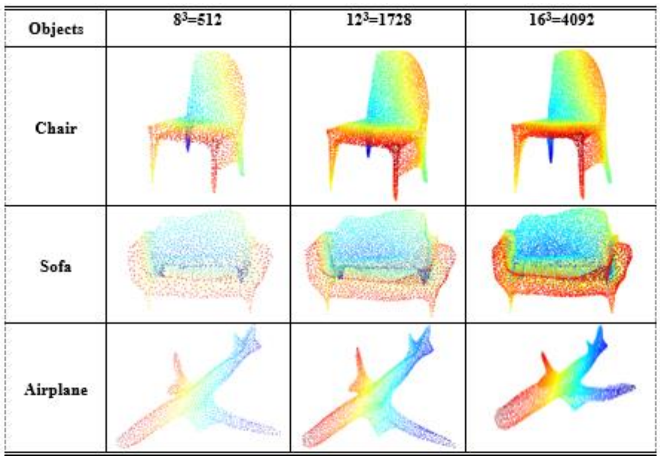

Figure 15. Results of point cloud with 8^3, 12^3 and 16^3 respectively

The difference in 3D visualization quality depends on a number of points in the 3D point cloud Figure 7 shows the representation of some classes with 8^3, 12^3 and 16^3 respectively. As we increase the number of points in our models it takes more time to render but the quality is better as compared to a smaller number of points. The quality also depends on less distance between the points. Our system constructs 3D according to a required number of points. As we see in Figure 15 the quality of the point cloud increases with the number of points in the model. The quality of the point cloud increase with a number of points but it takes more computational time and computation power to reconstruct and render the final model.

Table 7 contrasts many 3D object reconstruction techniques for a range of object types, including airplanes, benches, cars, chairs, and sofas. Reconstruction mistakes are represented by the numbers in the table; lower values indicate greater performance when recreating 3D objects from raw data.

To visualize the comparison between ground truth and the predicted model. Figure 16 represents the detailed view of some classes from the selected dataset.

Table 7. CD scores of different methods in our proposed system achieved lower CD in all compared classes (the smallest number represents better performance)

Object	RealPoint3D [6]	PSGN [5]	3D-ReconstNet [7]	Ours (ShapeNetCore)	Ours (ModelNet10)	Ours (ObjectNet3D)
Airplane	0.00079	0.00100	0.0242	**0.0006**	--	**0.0009**
Bench	0.00211	0.00251	0.0357	**0.0009**	--	--
Car	0.00126	0.00128	0.0359	0.0014	--	**0.0008**
Chair	0.00213	0.00238	0.0441	0.0013	0.0010	**0.0007**
Sofa	0.00195	0.00220	0.0614	0.0027	**0.0012**	0.0028

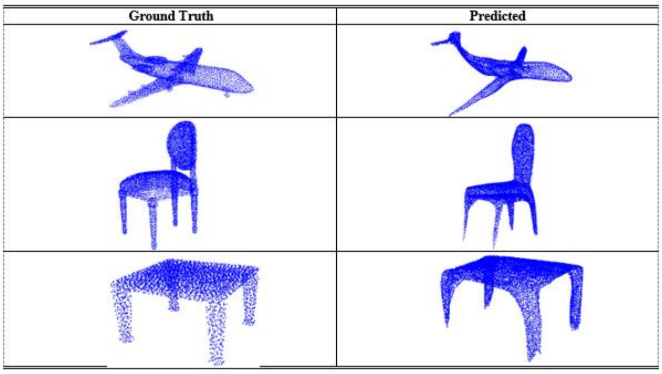

Figure 16. Visual comparison between ground truth and predicted model

Different types of standards and loss functions for the evaluation of 3D models with their ground truth have been used in past. These functions include chamfer distance (CD) [24, 54] and EMD [55] that calculate the overall performance of the 3D point cloud model.

$$CD(M_1, M_2) = \frac{1}{|M_1|}\sum_{n=1}^{M_1}||x-y||_2 + \frac{1}{|M_2|}\sum_{n=1}^{M_2}||x-y||_2 \tag{14}$$

where, M_1 is the generated model and M_2 is the ground truth model. M_1 and M_2 are $\subseteq\mathbb{R}3$.

6. CONCLUSIONS AND FUTURE WORK

In this paper, the simplest approach has been used proposed and validated to extract features that further help in the generation of the 3D point cloud. Some features like depth have been estimated using the deep neural network method. When creating an image from a real scene many types of features for 3D information are lost like depth, and view from different angles. Once we got the depth. Furthermore, filters are applied to compute the surface normal. Then all these features were used in the GCN-based network to estimate the 3D and we got the result of a 3D point cloud in the form of object point clouds are one of the finest representations of 3D models for the purpose of analysis. One significant drawback of the current 3D point cloud network is its limited performance in scenarios involving occlusions and low-quality images. Occlusions occur when objects in the scene partially or completely block the view of other objects, resulting in missing or obscured information in the input images. Similarly, low-quality images lack sufficient detail or clarity, often due to factors such as low resolution, noise, or blurriness, making it challenging for algorithms to extract accurate depth and geometric information.

In future work, we will work on human face 3D reconstruction and human body pose estimation and 3D reconstruction. Also, optimization of 3D models using imaging datasets that improve the model alignment and shape detailing.

ACKNOWLEDGEMENT

The paper was funded by Deanship of Scientific Research at Najran University under the Research Group Funding program grant code (NU/RG/SERC/12/6). This research is supported and funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R410), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

REFERENCES

[1] Jain, A.K. (1989). Fundamentals of Digital Image Processing. Prentice-Hall, Inc..

[2] Chaddha, N., Tan, W.C., Meng, T.H.Y. (1994). Color quantization of images based on human vision perception. In Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing IEEE. Adelaide, SA, Australia, 5: 89-92. https://doi.org/10.1109/ICASSP.1994.389552

[3] Chen, X., Golovinskiy, A., Funkhouser, T. (2009). A benchmark for 3D mesh segmentation. Acm Transactions on Graphics (TOG), 28(3): 1-12. https://doi.org/10.1145/1531326.1531379

[4] Zamorski, M., Zięba, M., Klukowski, P., Nowak, R., Kurach, K., Stokowiec, W., Trzciński, T. (2020). Adversarial autoencoders for compact representations of

3D point clouds. Computer Vision and Image Understanding, 193: 102921. https://doi.org/10.1016/j.cviu.2020.102921
[5] Yang, G., Huang, X., Hao, Z., Liu, M.Y., Belongie, S., Hariharan, B. (2019). PointFlow: 3D point cloud generation with continuous normalizing flows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4541-4550.
[6] Zhang, Y., Liu, Z., Liu, T., Peng, B., Li, X. (2019). RealPoint3D: An efficient generation network for 3D object reconstruction from a single image. IEEE Access, 7: 57539-57549. https://doi.org/10.1109/ACCESS.2019.2914150
[7] Li, B., Zhang, Y., Zhao, B., Shao, H. (2020). 3D-ReConstnet: A single-view 3D-object point cloud reconstruction network. IEEE Access, 8: 83782-83790. https://doi.org/10.1109/ACCESS.2020.2992554
[8] Cardenas-Garcia, J.F., Yao, H.G., Zheng, S. (1995). 3D reconstruction of objects using stereo imaging. Optics and Lasers in Engineering, 22(3): 193-213. https://doi.org/10.1016/0143-8166(94)00046-D
[9] Sengupta, S., Greveson, E., Shahrokni, A., Torr, P.H. (2013). Urban 3D semantic modelling using stereo vision. In 2013 IEEE International Conference on robotics and Automation, Karlsruhe, Germany, pp. 580-585. https://doi.org/10.1109/ICRA.2013.6630632
[10] Shen, S. (2013). Accurate multiple view 3D reconstruction using patch-based stereo for large-scale scenes. IEEE Transactions on Image Processing, 22(5): 1901-1914. https://doi.org/10.1109/TIP.2013.2237921
[11] Jang, M., Lee, S., Kang, J., Lee, S. (2021). Active stereo matching benchmark for 3D reconstruction using multi-view depths. In 2021 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Terengganu, Malaysia, pp. 215-220. https://doi.org/10.1109/ICSIPA52582.2021.9576787
[12] Kang, J., Lee, S., Jang, M., Lee, S. (2021). Gradient flow evolution for 3D fusion from a single depth sensor. IEEE Transactions on Circuits and Systems for Video Technology, 32(4): 2211-2225. https://doi.org/10.1109/TCSVT.2021.3089695
[13] Kästner, L., Frasineanu, V.C., Lambrecht, J. (2020). A 3D-deep-learning-based augmented reality calibration method for robotic environments using depth sensor data. In 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, pp. 1135-1141. https://doi.org/10.1109/ICRA40945.2020.9197155
[14] Ballabeni, A., Apollonio, F.I., Gaiani, M., Remondino, F. (2015). Advances in image pre-processing to improve automated 3D reconstruction. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 40: 315-323. https://doi.org/10.5194/isprsarchives-XL-5-W4-315-2015
[15] Wang, Y., Chao, W.L., Garg, D., Hariharan, B., Campbell, M., Weinberger, K.Q. (2019). Pseudo-lidar from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, USA, pp. 8445-8453.
[16] Ahmed, A., Jalal, A., Rafique, A.A. (2019). Salient segmentation based object detection and recognition using hybrid genetic transform. In 2019 International Conference on Applied and Engineering Mathematics (ICAEM), Taxila, Pakistan, pp. 203-208. https://doi.org/10.1109/ICAEM.2019.8853834
[17] Hambarde, P., Murala, S., Dhall, A. (2021). UW-GAN: Single-image depth estimation and image enhancement for underwater images. IEEE Transactions on Instrumentation and Measurement, 70: 1-12. https://doi.org/10.1109/TIM.2021.3120130
[18] Qi, X., Liu, Z., Liao, R., Torr, P.H., Urtasun, R., Jia, J. (2020). Geonet++: Iterative geometric neural network with edge-aware refinement for joint depth and surface normal estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(2): 969-984. https://doi.org/10.1109/TPAMI.2020.3020800
[19] Laine, S., Karras, T. (2011). Efficient sparse voxel octrees. IEEE Transactions on Visualization and Computer Graphics, 17(8): 1048-1059. https://doi.org/10.1109/TVCG.2010.240
[20] Fowler, B., El Gamal, A., Yang, D.X. (1994). A CMOS area image sensor with pixel-level A/D conversion. In Proceedings of IEEE International Solid-State Circuits Conference-ISSCC'94, San Francisco, CA, USA, pp. 226-227. https://doi.org/10.1109/ISSCC.1994.344659
[21] Chen, S., Niu, S., Lan, T., Liu, B. (2019). PCT: Large-scale 3D point cloud representations via graph inception networks with applications to autonomous driving. In 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, pp. 4395-4399. https://doi.org/10.1109/ICIP.2019.8803525
[22] Cheng, S., Chen, X., He, X., Liu, Z., Bai, X. (2021). Pra-net: Point relation-aware network for 3D point cloud analysis. IEEE Transactions on Image Processing, 30: 4436-4448. https://doi.org/10.1109/TIP.2021.3072214
[23] Lu, J., Li, Z., Bai, J., Yu, Q. (2022). Oriented and directional chamfer distance losses for 3D object reconstruction from a single image. IEEE Access, 10: 61631-61638. https://doi.org/10.1109/ACCESS.2022.3179109
[24] Nguyen, T., Pham, Q.H., Le, T., Pham, T., Ho, N., Hua, B.S. (2021). Point-set distances for learning representations of 3D point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10478-10487.
[25] Chen, Y., Li, H., Gao, R., Zhao, D. (2020). Boost 3-D object detection via point clouds segmentation and fused 3-D GIoU-L_1 loss. IEEE Transactions on Neural Networks and Learning Systems, 33(2): 762-773. https://doi.org/10.1109/TNNLS.2020.3028964
[26] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J. (2015). 3D ShapeNets: A deep representation for volumetric shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1912-1920.
[27] Xiang, Y., Mottaghi, R., Savarese, S. (2014). Beyond pascal: A benchmark for 3D object detection in the wild. In IEEE Winter Conference on Applications of Computer Vision, Steamboat Springs, CO, USA, pp. 75-82. https://doi.org/10.1109/WACV.2014.6836101
[28] Xiang, Y., Kim, W., Chen, W., Ji, J., Choy, C., Su, H., Mottaghi, R., Guibas, L., Savarese, S. (2016). ObjectNet3D: A large scale database for 3d object recognition. In Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, Proceedings, Part VIII. Springer, Cham, 14: 160-176. https://doi.org/10.1007/978-3-319-46484-8_10

[29] Hu, Z., Han, T., Sun, P., Pan, J., Manocha, D. (2019). 3-D deformable object manipulation using deep neural networks. IEEE Robotics and Automation Letters, 4(4): 4255-4261. https://doi.org/10.1109/LRA.2019.2930476

[30] Liu, F., Shen, C., Lin, G. (2015). Deep convolutional neural fields for depth estimation from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, USA, pp. 5162-5170.

[31] Ren, S., He, K., Girshick, R., Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, Canada, 1: 91-99.

[32] Han, X.F., Laga, H., Bennamoun, M. (2019). Image-based 3D object reconstruction: State-of-the-art and trends in the deep learning era. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(5): 1578-1604. https://doi.org/10.1109/TPAMI.2019.2954885

[33] Yu, H.W., Lee, B.H. (2018). A variational feature encoding method of 3D object for probabilistic semantic SLAM. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, pp. 3605-3612. https://doi.org/10.1109/IROS.2018.8593831

[34] Gadelha, M., Wang, R., Maji, S. (2018). Multiresolution tree networks for 3D point cloud processing. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 11211: 103-118.

[35] Lim, I., Ibing, M., Kobbelt, L. (2019). A convolutional decoder for point clouds using adaptive instance normalization. Computer Graphics Forum, 38(5): 99-108. https://doi.org/10.1111/cgf.13792

[36] Yang, C., Xie, H., Tian, H., Yu, Y. (2021). Dynamic domain adaptation for single-view 3D reconstruction. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, pp. 3563-3570. https://doi.org/10.1109/IROS51168.2021.9636343

[37] Zhang, C., Cui, Z., Zhang, Y., Zeng, B., Pollefeys, M., Liu, S. (2021). Holistic 3D scene understanding from a single image with implicit representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville, USA, pp. 8833-8842.

[38] Qian, G., Abualshour, A., Li, G., Thabet, A., Ghanem, B. (2021). PU-GCN: Point cloud upsampling using graph convolutional networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville, USA, pp. 11683-11692.

[39] Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z. (2016). Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, pp. 1874-1883.

[40] Dhome, M., Richetin, M., Lapreste, J.T., Rives, G. (1989). Determination of the attitude of 3D objects from a single perspective view. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(12): 1265-1278. https://doi.org/10.1109/34.41365

[41] Hu, J., Shen, L., Sun, G. (2018). Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Saltt Lake, USA, pp. 7132-7141.

[42] Hu, J., Ozay, M., Zhang, Y., Okatani, T. (2019). Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In 2019 IEEE Winter Conference on Applications of Computer Vision (WACV) Waikoloa Village, USA, pp. 1043-1051. https://doi.org/10.1109/WACV.2019.00116

[43] Ghadi, Y.Y., Rafique, A.A., Al Shloul, T., Alsuhibany, S.A., Jalal, A., Park, J. (2022). Robust object categorization and Scene classification over remote sensing images via features fusion and fully convolutional network. Remote Sensing, 14(7): 1550. https://doi.org/10.3390/rs14071550

[44] Waheed, M., Javeed, M., Jalal, A. (2021). A novel deep learning model for understanding two-person interactions using depth sensors. In 2021 International Conference on Innovative Computing (ICIC) Lahore, Pakistan, pp. 1-8. https://doi.org/10.1109/ICIC53490.2021.9692946

[45] Ramamonjisoa, M., Lepetit, V. (2019). SharpNet: Fast and accurate recovery of occluding contours in monocular depth estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops Seoul, Korea, pp. 2109-2118.

[46] Klasing, K., Althoff, D., Wollherr, D., Buss, M. (2009). Comparison of surface normal estimation methods for range sensing applications. In 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, pp. 3206-3211. https://doi.org/10.1109/ROBOT.2009.5152493

[47] Jordan, K., Mordohai, P. (2014). A quantitative evaluation of surface normal estimation in point clouds. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chigaco, IL, USA, pp. 4220-4226, https://doi.org/10.1109/IROS.2014.6943157

[48] Fahim, G., Amin, K., Zarif, S. (2021). Single-View 3D reconstruction: A Survey of deep learning methods. Computers & Graphics, 94: 164-190. https://doi.org/10.1016/j.cag.2020.12.004

[49] Chelloug, S.A., Ashfaq, H., Alsuhibany, S.A., Shorfuzzaman, M., Alsufyani, A., Jalal, A., Park, J. (2023). Real objects understanding using 3D haptic virtual reality for e-learning education. Computers, Materials & Continua, 75(1): 1607-1624, https://doi.org/10.32604/cmc.2023.032245

[50] Shu, D.W., Park, S.W., Kwon, J. (2019). 3D point cloud generative adversarial network based on tree structured graph convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea (South), pp. 3859-3868. https://doi.org/10.1109/ICCV.2019.00396

[51] Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F. (2015). ShapeNet: An information-rich 3D model repository. arXiv preprint arXiv:1512.03012. https://doi.org/10.48550/arXiv.1512.03012

[52] Miller, G.A. (1995). WordNet: A lexical database for English. Communications of the ACM, 38(11): 39-41. https://doi.org/10.1145/219717.219748

[53] Shi, J., Wang, Y. (2019). Hyperbolic Wasserstein distance for shape indexing. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(6): 1362-1376. https://doi.org/10.1109/TPAMI.2019.2898400

[54] Fan, R., Wang, H., Xue, B., Huang, H., Wang, Y., Liu, M., Pitas, I. (2021). Three-filters-to-normal: An accurate and ultrafast surface normal estimator. IEEE Robotics and Automation Letters, 6(3): 5405-5412. https://doi.org/10.1109/LRA.2021.3067308

[55] Liu, M., Sheng, L., Yang, S., Shao, J., Hu, S.M. (2020). Morphing and sampling network for dense point cloud completion. In Proceedings of the AAAI Conference on Artificial Intelligence, Hilton Hawaiian Village, USA, 34(07): 11596-11603. https://doi.org/10.1609/aaai.v34i07.6827

NOMENCLATURE

LiDAR	Light Detection and Ranging
VAEs	Variational Auto Encoders
DNN	Deep Neural Networks
GCN	Graph Convolutional Networks
GPUs	Graphic Processing Units
RMS	Root Mean Square
SENET	Squeeze-and-Excitation Network
CD	Chamfer Distance
EMD	Earth Mover's Distance