Jin *et al. Chin. J. Mech. Eng.* (2021) 34:93
https://doi.org/10.1186/s10033-021-00615-x

Chinese Journal of Mechanical Engineering

ORIGINAL ARTICLE Open Access

Weakly-Supervised Single-view Dense 3D Point Cloud Reconstruction via Differentiable Renderer

Peng Jin[1], Shaoli Liu[1], Jianhua Liu[1*], Hao Huang[1], Linlin Yang[2], Michael Weinmann[2] and Reinhard Klein[2]

Abstract

In recent years, addressing ill-posed problems by leveraging prior knowledge contained in databases on learning techniques has gained much attention. In this paper, we focus on complete three-dimensional (3D) point cloud reconstruction based on a single red-green-blue (RGB) image, a task that cannot be approached using classical reconstruction techniques. For this purpose, we used an encoder-decoder framework to encode the RGB information in latent space, and to predict the 3D structure of the considered object from different viewpoints. The individual predictions are combined to yield a common representation that is used in a module combining camera pose estimation and rendering, thereby achieving differentiability with respect to imaging process and the camera pose, and optimization of the two-dimensional prediction error of novel viewpoints. Thus, our method allows end-to-end training and does not require supervision based on additional ground-truth (GT) mask annotations or ground-truth camera pose annotations. Our evaluation of synthetic and real-world data demonstrates the robustness of our approach to appearance changes and self-occlusions, through outperformance of current state-of-the-art methods in terms of accuracy, density, and model completeness.

Keywords: Point clouds reconstruction, Differentiable renderer, Neural networks, Single-view configuration

1 Introduction

The inference of underlying object or scene geometry is among the classical goals of computer vision and graphics, and a fundamental prerequisite for numerous applications in entertainment, robotics, navigation, and architecture. Examples include guidance of robot interactions with objects in a scene based on their shape, as well as augmented and virtual reality solutions for gaming, interior design [1], remote collaboration [2–4] and teleoperation [5, 6]. The geometry reconstruction is also significant for microscopic scale objects. Such as the surface morphology inference based on the surface profile reconstruction [7] is served for the assembling deviation estimation [8] and analysis of the replacement of the actual machining surface [9].

Besides the well-established multi-view approaches, such as multi-view stereo [10], structure-from-motion (SfM) [11], simultaneous localization and mapping (SLAM) [12] and single-view-based three-dimensional (3D) scanning based on structured light systems [13] or laser scanners [14], more approaches are now focusing on learning-based scene representation schemes [15], especially for single-view scenarios. When taken into account, prior knowledge derived from large-scale datasets can yield remarkable reconstruction results from single images [16].

Common 3D scene representations include depth images [17–23], voxel-based representations [24–30], triangular meshes [31–34], and point clouds [35–40]. However, 3D convolutional neural network (CNN) approaches designed for voxel-based scene representations trade off

*Correspondence: jeffliu@bit.edu.cn
[1] School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China
Full list of author information is available at the end of the article

the benefits of structured input data, with the limitation of representing surface information with relatively few voxels. Hence, the granularity of the reconstruction result is strongly limited by the computational burden and memory consumption associated with 3D CNNs. Furthermore, considering structured input data in terms of the point connectivity of meshes is more efficient due to the direct consideration of points on the surface; however, it is non-trivial to efficiently integrate the connectivity information in the training process. In turn, unstructured point clouds offer the aforementioned advantage of direct representation of the surface with high granularity, without the need to consider the connectivity between points during training; however, the lack of any grid structure and permutation invariance must be considered within point cloud specific architectures and loss definitions [37–40]. Key challenges include the generation of dense point clouds to avoid incomplete object representation, has a high computational burden and high memory requirements.

The reconstruction quality of single image-based approaches depends heavily on the available training data. In general, impressive single image-based reconstruction results have been obtained using large datasets of ground truth annotations. Obtaining perfect 3D computer-aided design (CAD) models as ground truth data for real-world environments is highly challenging; therefore, several approaches have focused on weakly supervised [25, 41, 42] or unsupervised [43, 44] learning to reduce/mitigate the need to acquire 3D ground truth data for explicit supervision. However, neural scene representation and rendering, as applied in Ref. [43], does not well represent the 3D structure, thereby limiting the quality of 3D structure recovery from a small number of observations. The structure-aware scene representation network presented in Ref. [44] encodes both geometry and appearance; however, the applied ray marcher cannot accommodate surfaces with holes and boundaries of self-occluding structures, as commonly encountered in the 'chair' object category. Nevertheless, these approaches show potential, particularly for multi-view observations.

The key proposition of this paper is that an accurately predicted shape should provide reasonable depth estimates from any viewpoint. For this purpose, we take the depth maps as supervision signals and propose a novel weakly supervised approach to reconstruct a dense 3D point cloud. Given the input of a single red-green-blue (RGB) image, we use an encoder-decoder architecture to first encode the RGB information in a latent representation and then predict the 3D structure of the considered object from different viewpoints. Then, we combine these individual 3D structure predictions into a common coordinate system to reconstruct the point clouds, and further synthesize the depth maps from novel viewpoints to optimize the two-dimensional (2D) prediction error.

Most optimization processes [25, 42, 45, 46] rely on the availability of ground truth data for novel viewpoint poses. For instance, Navaneet et al. [45] and Lin et al. [46] specified the viewpoint poses for CAD models. Tulsiani et al. [25] and Gwak et al. [42] trained models based on viewpoint pose annotations. Developing setups for low-cost object digitization without the requirement for expensive annotations or calibration requires that these restrictions be overcome. Therefore, we designed a differentiable rendering module and combined it with a pose estimation network to identify the poses for novel viewpoints. The rendering module is capable of handling the appearance changes and self-occlusions that may occur from certain viewpoints, and can estimate the camera poses even with large baselines, which makes it possible to randomly set the novel viewpoints.

2 Structure Estimation

In an initial step, we aim to derive a dense 3D point cloud representation from a single RGB image acquired from an arbitrary view. For this purpose, we attempted to leverage the potential of deep learning for generative 3D modelling. Key challenges include efficient and accurate 3D representation of the considered object, as well as the design of a pipeline that allows end-to-end-learning without requiring annotated data. The proposed pipeline is shown in Figure 1.

To meet these challenges, we use an encoder-decoder architecture that encodes information from the input image in a latent scene representation that, in turn, allows 3D structure predictions $D_v(v \in V)$ from V different fixed viewpoints. These view-dependent structures D_v correspond to an image-based representation of 3D point cloud coordinates $x_i = (x_i, y_i, z_i)$ according to the respective viewpoint v. Representing point clouds in terms of multi-channel images allows for the use of 2D convolutions instead of memory-intensive 3D convolutional operations for calculation of the volumetric structure. During training, the encoder learns the latent scene representation, and the decoder learns to generate 3D structures $\hat{D}_v$ from that representation. Finally, the structure images $x_i = (x_i, y_i, z_i)$ predicted for different

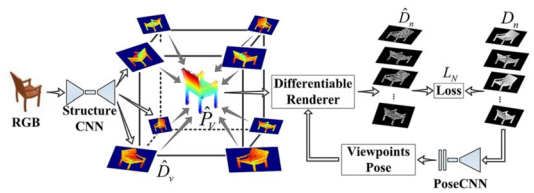


Figure 1 Proposed pipeline to reconstruct 3D point cloud

viewpoints $v \in V$ are fused into a single 3D point cloud $\hat{p}_i \in \hat{P}_V$ by transforming the respective view-centric point cloud coordinates into a common world coordinate system (WCS) according to

$$\hat{p}_i = R_v^{-1}(K^{-1}\hat{x}_i - t_v), \tag{1}$$

where K denotes the camera calibration matrix, and the views are specified based on pairs of rotations and translations (R_v, t_v). Thus, applying (R_v, t_v) shifts a point from the world coordinate frame to the view-centric coordinate frame of view v, and the inverse transform is then applied to transfer points from view-specific coordinate frames to the global coordinate system.

Note that training the StructureCNN does not rely on ground truth annotations of 3D structures D_v or 3D shapes P_V for direct supervision as required in the approach of Lin et al. [46]. Instead, we jointly train the structure network and a component that optimizes 2D projection errors and the camera pose prediction.

3 Optimization Based on 2D Projections from Multiple Views

The 3D point cloud reconstruction obtained by fusing the multi-view structure predictions from the aforementioned structure network is noisy and needs further optimization. Further optimization of our point cloud avoids the need for novel viewpoint pose annotations by integrating a pose estimation network into the designed differentiable rendering module.

3.1 Differentiable Rendering Module

The renderer represents the forward imaging process of a camera. In our pipeline, the renderer takes the reconstructed point cloud $\hat{P}_V$ as the input to render depth images $\hat{D}_n$ for novel views (R_n, t_n), which are then used for 2D projection optimisation to minimise the depth errors $L_N = \sum_{n=1}^{N} ||D_n - \hat{D}_n||_1$. Here, the image coordinates $\hat{x}$ of the individual points of the common point cloud under the view (R_n, t_n) are obtained according to

$$\hat{x}_i = K(R_n\hat{p}_i + t_n), \tag{2}$$

This process can be inverted. Given the depth information and respective image coordinates, the points of the surface parts visible in a particular view can be reconstructed by back projection of the 2D depth maps.

Unlike the approach of Fan et al. [35], wherein the number of points in the point cloud is fixed and predefined, our approach allows for the generation of dense and disordered points varying in number for different objects. As shown in Figure 2(a), many points may project onto the same pixels, and the resulting discretization may reduce the image quality. Lin et al. [46] developed a pseudo-renderer to resolve this issue based on upsampling the depth image by a factor U, such that points are projected onto isolated pixels (Figure 2(b)). This results in high memory consumption. Unfortunately, as the point clouds are dense, different points may still be projected onto the same pixels. The optimization does not consider these pixels, and their gradients do not contribute to the outputs; this resulting in more outliers. Furthermore, discretization error of the 2D projections of the point clouds caused by rounding, according to the depth image resolution [46], means that the model undifferentiated among view poses.

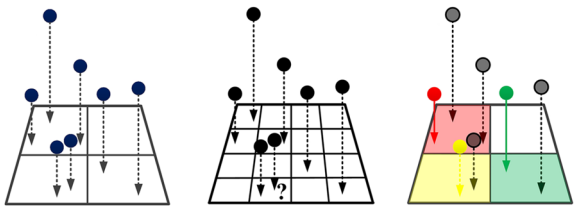


Figure 2 The projection of dense point cloud

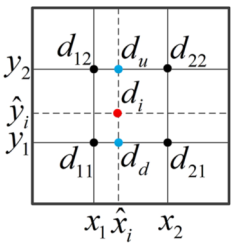


Figure 3 The GT depth value is sampled by bilinear interpolation

To obtain better point cloud optimization results, we only store the point with minimum depth $\hat{z}_i = \min_{j \in s} \hat{z}_j$ in the respective image $\hat{x}_i = (\hat{x}_i, \hat{y}_i, \hat{z}_i)$ in cases of s projected points per pixel (Figure 2c). In other words, we only consider visible aspects in the respective views. All projected pixels contribute to the optimization process, to reduce the influence of outliers on the results. Furthermore, to also achieve differentiability with respect to the viewpoint, we compute the ground-truth depth value d_i corresponding to the rendered depth value $\hat{d}_i = \hat{z}_i$ at location $(\hat{x}_i, \hat{y}_i)$ by bilinear interpolation:

$$\begin{cases} d_d \approx \frac{x_2 - \hat{x}_i}{x_2 - x_1} d_{11} + \frac{\hat{x}_i - x_1}{x_2 - x_1} d_{21}, \\ d_u \approx \frac{x_2 - \hat{x}_i}{x_2 - x_1} d_{12} + \frac{\hat{x}_i - x_1}{x_2 - x_1} d_{22}, \\ d_i \approx \frac{y_2 - \hat{y}_i}{y_2 - y_1} d_d + \frac{\hat{y}_i - y_1}{y_2 - y_1} d_u. \end{cases} \tag{3}$$

As shown in Figure 3, where d_{11}, d_{12}, d_{21}, and d_{22} are the depth values of the local four-pixel neighborhood on the ground truth, d_i is approximated by two linear interpolations, d_d and d_u. The bilinear sampling at location $(\hat{x}_i, \hat{y}_i)$ is differentiable with respect to the camera pose (R_n, t_n), and the reconstructed point $\hat{p}_i$ is augmented in Eq. (2), such that the framework is differentiable with respect to point cloud generation and viewpoint pose prediction, and can be trained end-to-end.

3.2 Camera Pose Estimation Network

The above designed renderer can use different camera poses (R_n, t_n), so we can estimate the poses in novel views. For this purpose, we integrate a pose estimation network into the rendering module. Here, we use a convolutional network (PoseCNN) that takes the depth maps $D_n(n \in N)$ at N novel viewpoints as input and estimates their respective poses (R_n, t_n). This allows us to avoid dependence on pre-defined depth maps with pose annotations, as there are likely to only be depth maps with unknown poses available for supervision.

As illustrated in Figure 4, we take the depth maps D_f with known poses as references, to train the PoseCNN and estimate the poses (R_n, t_n) in N novel views. In theory, only one reference D_f with an unknown pose can be used to successfully train PoseCNN. We can take the local coordinate system of D_f as the WCS and estimate other camera poses with respect to D_f. As the larger number of D_f contributes to the pose estimation accuracy (see the experimental results in Section 4.2.2), we use eight reference views D_f in this paper.

The point cloud $\hat{P}_N$ fused with accurate estimated camera poses is expected to align with the ground-truth point cloud P_N; the Euclidean distance between them is very small. There are 3D metrics for comparing point clouds, such as the Chamfer distance [35], which determines the distance from each point to the nearest neighbor in another set of point clouds. To avoid the need for a costly 3D-based optimization using computationally intensive 3D metrics, the rendered depth map $\hat{D}_f$ should be consistent with D_f. The 2D optimization based on minimizing the L_1 loss between $\hat{D}_f$ and D_f is more efficient. Furthermore, the 3D metrics are invalid when there are dramatically different appearances between views, as

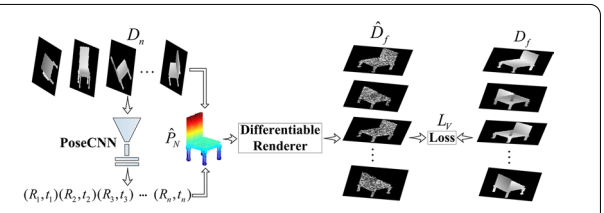
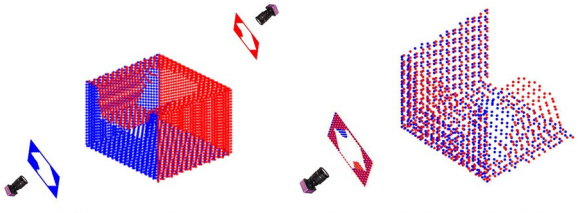


Figure 4 Novel viewpoint poses prediction



(a) Two opposite views (b) Incorrect alignment of views

Figure 5 Two different view configurations

shown in Figure 5(a), in which two views with opposite orientations capture completely different aspects of the scene. Here, the 3D optimization will incorrectly estimate the novel view and will instead largely coincide with the reference view, as shown in Figure 5(b). The proposed 2D optimization is effective for this situation and robust to appearance changes and self-occlusion, as verified experimentally (see Section 4.2.2).

4 Experiments

4.1 Implementation Details

We used the most recent and relevant research results of Lin et al. [46] and Navaneet et al. [45], based on their state-of-the-art single-view point cloud reconstruction methods, as the baseline and prepared identical datasets to allow comparison with our proposed method. The details of the experimental setup and qualitative and quantitative results are as follows.

4.1.1 Data Preparation

(1) Synthetic dataset: ShapeNetCore [47] contains about 55 object categories, from which a subset of 3D models is used for experimental evaluations. For each 3D model, we rendered 24 RGB images ($64 \times 64 \times 3$) with azimuth angle steps of 15° and elevation angles 30°, 100 depth maps $D_n(64 \times 64)$ at random novel viewpoints, and eight 3D structures D_v at fixed viewpoints (i.e. the eight corners of a central cube). More D_v will inferred denser point clouds. And D_n located in supervised views, should capture more details and there are almost no occluded areas in the field of views.

(2) Real-world dataset: Pix3D [48] contains real images and corresponding 3D CAD models. We selected four categories for our experiment, i.e. 'bed', 'chair', 'desk', and 'sofa', rendered D_n, and generated ground-truth point clouds based on the CAD models. Additionally, we tested the 'chair' object category from the Stanford Online Products dataset [49].

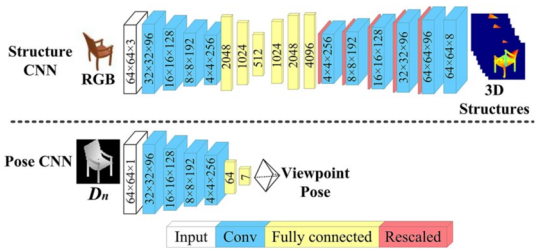


Figure 6 The neural network architectures

4.1.2 Network Architecture

We designed 2D convolutional neural networks. As shown in Figure 6, StructureCNN and PoseCNN share the same encoder architecture. The encoder consists of four convolution layers having 96, 128, 192, and 256 channels, and three fully connected layers having 2048, 1024, and 512 neurons. For StructureCNN, the decoder consists of three fully connected layers with 1024, 2048, and 4096 neurons. The feature maps are rescaled by nearest neighbor interpolation, followed by convolution layers. Batch normalization and rectified linear unit (ReLU) layers were added between the convolution layers. The fixed filter size was 3×3. There were two and one strides in the encoder and decoder, respectively. PoseCNN used two fully connected layers with 64 and 7 neurons each. The outputs included a quaternion and the x, y, z position of the viewpoints. The PoseCNN can predict the viewpoints of the depth maps scattered in supervised views, which facilitate the training of the StructrueCNN. After the training strategy, the point cloud of an object can be inferenced by feeding the single RGB image into the StructureCNN.

4.1.3 Training Paradigm

As the inferred viewpoint in initial training iterations is often inaccurate, which will result in the learned point cloud unmeaningful. Thus, learning these together is susceptible to local minima. Followed by the suggestions of prior work [34], a two-step training paradigm was employed. First, we trained PoseCNN with eight fixed viewpoints taken as references $F = V = 8$. $D_f = D_v$, $Df = \hat{D}_v^{'}$ corresponds to the rendered result. Then, we trained StructureCNN based on estimated novel viewpoint poses. An RGB image randomly selected from 24 views was used as input for each iteration. In Eq. (4), the loss is defined as the L_1 distance. We used TensorFlow to implement our framework, with a learning rate of 0.0001 and ADAM optimisation.

$$\begin{cases} L_N = \sum_{n=1}^{N} ||D_n - \hat{D}_n||_1, \\ L_V = \sum_{v=1}^{V} ||D_v - \hat{D}_v^{'}||_1. \end{cases} \tag{4}$$

4.1.4 Experimental Design

The experiments mainly addressed three questions: (1) the accuracy and robustness of the viewpoint pose prediction, (2) the performance of the point cloud reconstruction for a single object category, and (3) the generality of the proposed framework to multiple and unseen categories.

For the first two questions, we trained and tested the network on the 'chair' category. For the third question, we trained and tested the network on multiple categories. All of the datasets were randomly split into training and test sets (80% and 20%, respectively). We also tested unseen categories.

4.2 Viewpoint Pose Prediction

4.2.1 Accuracy of the Pose Prediction

We used the eight fixed views as a reference to estimate 10 random novel viewpoints. Table 1 shows the averaged results of the test split. The camera orientation was represented by a quaternion. The error is the angle between the optical axes of the camera for the estimated and GT results. The largest error was 0.340°. According to the following results, the pose prediction was sufficiently accurate to guarantee point cloud reconstruction accuracy.

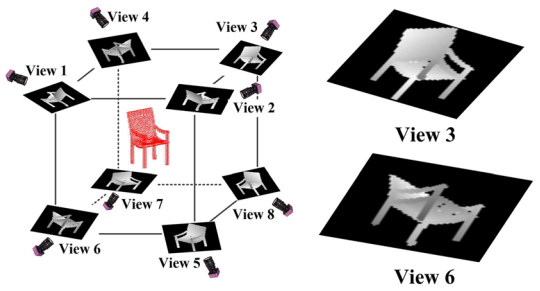
4.2.2 Robustness of the Viewpoint Pose Prediction

We used eight fixed views to evaluate the robustness of the viewpoint pose prediction and the impact of the number of reference views on the results. Figure 7(a) shows the eight fixed views; view 3 was selected as the reference. The estimated poses are listed in Table 2. Relative to view 3, to some degree there are appearance changes in the other seven views; for view 6 in particular in Figure 7(b), the appearance is completely different to that of the reference. Beside the appearance changes, every image has self-occlusions caused by the arms or legs of the chair. The orientations are all estimated accurately, indicating that the proposed renderer can not only differentiate among viewpoints, but is also robust to appearance changes and occlusions.

The accuracy of the results shown in Table 2 was lower than that of those in Table 1, indicating that the number of reference views affects the pose estimation accuracy. Figure 8 shows the training process for view 8 pose estimation according to the number of reference views.

Table 1 Pose prediction results of 10 noel views

Views	Rotations (quarternion)					Error (°)
View 1	Ours	− 0.589	− 0.793	− 0.115	0.097	0.134
	GT	0.589	0.793	0.113	− 0.096	
View 2	Ours	− 0.790	0.027	0.600	− 0.116	0.100
	GT	0.790	− 0.027	− 0.599	0.117	
View 3	Ours	0.331	− 0.175	− 0.422	0.824	0.210
	GT	0.330	− 0.175	− 0.423	0.825	
View 4	Ours	− 0.587	− 0.375	− 0.263	0.666	0.269
	GT	0.588	0.374	0.261	− 0.667	
View 5	Ours	− 0.556	− 0.320	− 0.435	− 0.630	0.340
	GT	0.556	0.322	0.436	0.628	
View 6	Ours	0.000	− 0.272	− 0.551	0.788	0.082
	GT	0.001	− 0.271	− 0.551	0.788	
View 7	Ours	− 0.695	− 0.362	0.127	0.607	0.262
	GT	0.695	0.361	− 0.129	− 0.607	
View 8	Ours	0.242	0.657	0.293	0.650	0.234
	GT	0.240	0.658	0.294	0.649	
View 9	Ours	− 0.511	− 0.470	0.422	0.581	0.197
	GT	0.511	0.470	− 0.424	− 0.580	
View 10	Ours	− 0.746	− 0.217	− 0.048	− 0.627	0.288
	GT	0.747	0.219	0.049	0.625	



(a) Eight fixed views (b) Two opposed views

Figure 7 The fixed views are scattered at 8 corners of a cube

The learning rate was 0.001. Figure 8 shows the first and final 50 iterations of the training process. The *Y*-axis is the angle between the optical axes of the estimated and GT results, where a greater number of reference views will improve convergence speed and accuracy. However, using a very high number of views is redundant and expensive; thus, the proposed framework used eight reference views.

The robustness and effectiveness of the viewpoint prediction allowed the reference and novel views to be set flexibly, without considering appearance changes or occlusions.

4.3 Point Cloud Reconstruction for a Single Object Category

Figure 9 shows the 3D point clouds generated for the chair test split. The reconstruction errors are defined by the point-wise 3D Euclidean distance using Eq. (5), which represents the 3D shape similarity [50]; $\hat{P}$ and P are generated and ground truth point clouds, respectively. According to Table 3, E is scaled by a factor of 100; our results are more accurate.

$$E = \left(\sum_{\hat{p}\in\hat{P}} \min_{p\in P} ||\hat{p} - p||_2\right)/||\hat{p}||_0. \tag{5}$$

Although, the network of Lin et al. [46] is pretrained based on the GT 3D structures, the training process does not consider pixels with more than one projection, which leads to outlier points, as shown in Figure 10(a). Navaneet et al. [45] also calculated the gradient of each pixel for optimization and obtained fewer outliers. However, they considered the masks as 2D observations and failed to resolve the concavity or finer details, as shown in Figure 10(b). We successfully generated these structures.

Table 2 Pose prediction results of the 7 fixed views

Views	Poses (quaternion)					Error (°)
View 1	Ours	− 0.310	0.181	0.419	0.833	1.046
	GT	0.339	− 0.175	− 0.424	− 0.820	
View 2	Ours	− 0.396	0.828	0.347	0.188	2.871
	GT	0.424	− 0.820	− 0.339	− 0.175	
View 4	Ours	− 0.173	0.339	− 0.822	− 0.421	0.292
	GT	0.175	− 0.339	0.820	0.424	
View 5	Ours	− 0.149	0.333	0.821	0.438	1.354
	GT	0.175	− 0.339	− 0.820	− 0.424	
View 6	Ours	− 0.815	0.432	0.174	0.342	0.995
	GT	0.820	− 0.424	− 0.175	− 0.339	
View 7	Ours	0.341	− 0.172	0.422	0.821	0.395
	GT	0.339	− 0.175	0.424	0.820	
View 8	Ours	0.416	− 0.823	0.344	0.174	0.996
	GT	0.424	− 0.820	0.339	0.175	

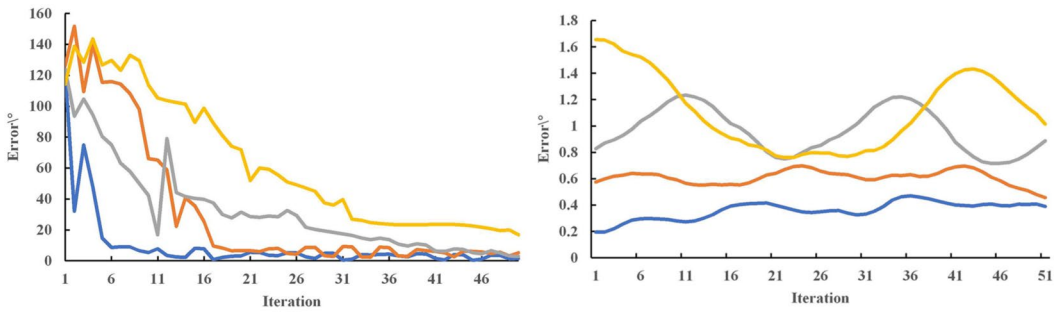


Figure 8 Training process to estimate the pose of view 8

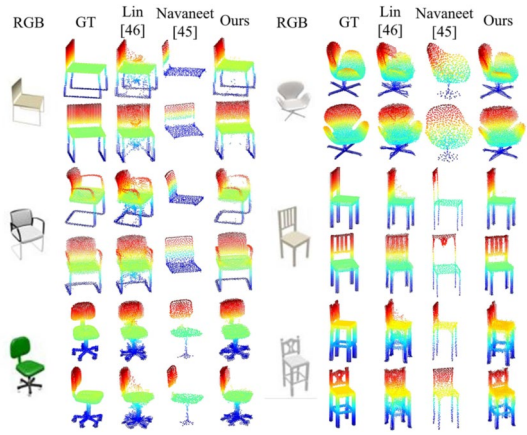


Figure 9 The reconstructed point clouds of the chair

Table 3 Reconstruction error of the chair

Method	Lin et al. [46]	Navaneet et al. [45]	Ours
E	1.72	1.68	1.61

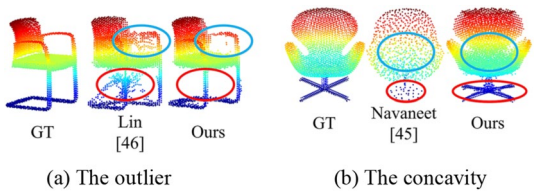


Figure 10 The results of outliers and concavity

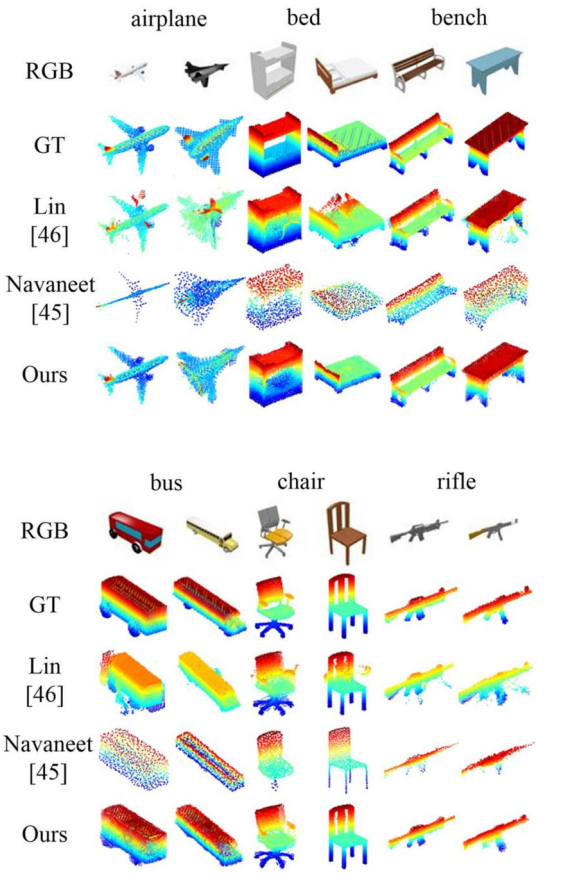


Figure 11 Multi-class reconstructed point clouds

4.4 Generative Reconstruction of Multiple Categories

4.4.1 Training/testing on Multiple Categories Using ShapeNetCore

The categories included 'airplane', 'bed', 'bench', 'bus', 'chair', and 'rifle'. The qualitative and quantitative results of the test split are shown in Figure 11 and Table 4.

For convex objects, such as a bus, the results of Navaneet et al. [45] are comparable to our own. For concave objects and finer details, such as the arms on chairs and rifles, our network is more effective.

4.4.2 Testing Out-of-category in ShapeNetCore

The ability to generalize prior learning for seen categories to unseen categories is important for the intelligent agent. Within the training set, the motorbike and car are completely novel categories; as there are few instances with similar shapes, they were used for the out-of-category tests. The results are shown in Figure 12 and Table 5. Many finer structures were resolved, such as the wheel of the motorbike and the tailgate of the car and, compared to Lin et al. [46], there were fewer outliers. Navaneet et al. [45] simply reconstructed the bounding boxes of the objects. We were also largely able to reconstruct these structures.

4.4.3 3D Reconstruction Using Pix3D and Stanford Online Products Datasets

To deal with real images, the proposed framework was further fine-tuned using the Pix3D dataset. We assumed a default intrinsic matrix with an orthographic camera K. The qualitative results for the bed, chair, desk, and sofa categories are illustrated in Figure 13, and the quantitative results are presented in Table 6. Despite the items on the bed and desk that heavily occluded the objects, we still effectively captured the finer details and concave structures.

As there are no CAD models in the Stanford Online Products dataset, we cannot generate ground-truth point clouds, so instead manually selected some untruncated instances for qualitative tests. The results are illustrated in Figure 14.

Based on the above experimental results, the proposed framework offers only a slight advantage for single category reconstruction. For multiple classes, the advantages of our framework are obvious. Especially for out-of-category and real images, we successfully reconstructed the concavity and finer structures. Furthermore, across different experimental settings, including single, multiple, and unseen categories of rendered and real-world data, the error rates were similar, at 1.61, 1.571, 2.200, and 1.51. The accuracy was higher for multiple- versus single-object cases. Overall, the visual and quantitative results demonstrate that the proposed framework has better generalization ability for synthetic and real-world domains.

Table 4 Reconstruction error in multi-class tests

	Airplane	Bed	Bench	Bus	Chair	Rifle	Mean
Ref. [46]	1.870	3.383	1.668	2.241	1.844	2.589	2.264
Ref. [45]	1.421	3.010	1.601	1.278	1.742	1.620	1.778
Ours	1.052	3.009	1.571	1.286	1.625	0.882	1.571

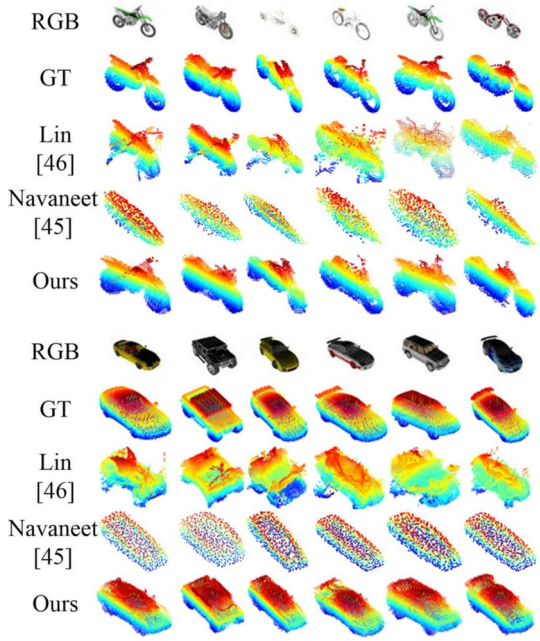


Figure 12 Out-of-category reconstructed point clouds

Table 5 Reconstruction error in Out-of-category tests

Category	Motorbike	Car	Mean
Lin et al. [46]	4.200	3.620	3.910
Navaneet et al. [45]	4.320	3.444	3.882
Ours	2.201	2.199	2.200

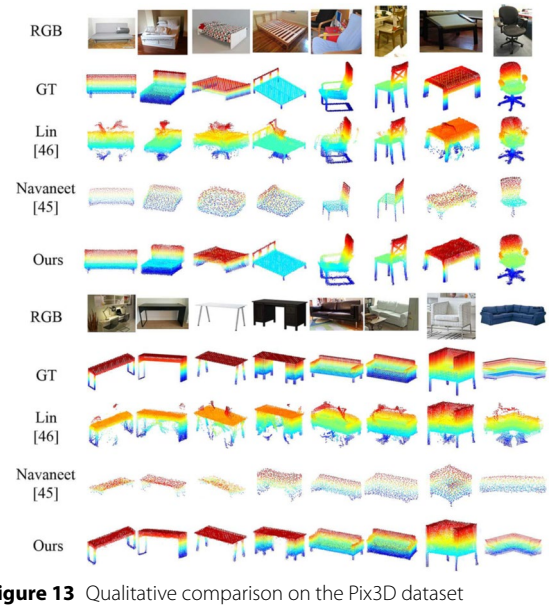


Figure 13 Qualitative comparison on the Pix3D dataset

Table 6 Reconstruction error of real images

Method	Lin et al. [46]	Navaneet et al. [45]	Ours
E	2.63	2.02	1.51

Figure 14 Qualitative comparison on the Stanford Online Products Dataset

5 Conclusions

We introduced an approach for complete 3D point cloud reconstruction from a single RGB image.

(1) We combined an encoder-decoder framework, for generative structure prediction from a single RGB image, and an optimization framework based on a differentiable renderer module, whereby the training is supervised through 2D observations in novel views.
(2) By adding a pose estimation network, the renderer is designed to be differentiable for both point cloud reconstruction and viewpoint pose prediction, which allows end-to-end training and avoids the need for viewpoint pose, structure, or mask annotations in the datasets.
(3) Experimental results for synthetic and real-world datasets demonstrated that our approach is robust to appearance changes and self-occlusions, and shows superior accuracy, density, model completeness, and generalization potential compared to state-of-the-art methods.

Acknowledgements
Not applicable.

Authors' Contributions
JL was in charge of the whole trial; PJ wrote the manuscript; MW assisted with writing the manuscript. All authors read and approved the final manuscript.

Authors' Information
Peng Jin received the Ph.D degree from *School of Mechanical Engineering, Beijing Institute of Technology, China*, in 2018, and he is currently a postdoctor at *Beijing Institute of Technology, China*. His current research interests include computer vision, computer graphics, deep learning, image-based 3D reconstruction.

Shaoli Liu received the Ph.D degree from *Department of Precision Instruments and Mechanology, Tsinghua University, China,* in 2012, and she is currently an associate professor in *School of Mechanical Engineering, Beijing Institute of Technology, China*. Her current research interests include machine vision and on-line detection.

Jianhua Liu received the Ph.D degree from *School of Mechanical Engineering, Beijing Institute of Technology, China*, in 2005, and he is currently a Professor in *School of Mechanical Engineering, Beijing Institute of Technology, China*. He has authored more than 200 publications. His current research interests include digital design and manufacturing, computer vision and photogrammetry. Professor Liu is a council member of the National Defense Technology Industry

Science and Technology Committee, China.
Hao Huang is currently a Ph.D student at the *School of Mechanical Engineering,*

Beijing Institute of Technology, China. His current interests include computer vision, object recognition.

Linlin Yang received the M.Eng. degrees from *Beihang University, China,* in 2017, and he is currently a Ph.D student with the Institute of Computer Science II, at *University of Bonn, Germany*. His current interests include computer vision, computer graphics, deep learning, hand pose estimation. He has published multiple top-tier papers in refereed journals and proceedings, including ICCV, CVPR and ECCV.

Michael Weinmann studied Electrical Engineering and Information Technology at the *University of Karlsruhe, Germany*, where he received his degree Dipl.-Ing. in 2009. After joining the Visual Computing Group at the *University of Bonn* in 2010, he received his PhD in computer science in 2016. His research interests include machine learning, 3D reconstruction, reflectance reconstruction, semantic scene interpretation and visualization where he published respective work on high-ranked conferences including CVPR, ICCV and ECCV as well as reputable journals such as the *ISPRS Journal of Photogrammetry and Remote Sensing*, *ACM Transactions on Graphics*, *IEEE Transactions on Visualization and Computer Graphics, and Sensors*.

Reinhard Klein received the Ph.D degree in computer science from *University of Tübingen, Germany*, in 1995. In 1999 he received an appointment as lecturer in computer science also from the *University of Tübingen, Germany*, with a thesis in computer graphics. In September 1999, he became an Associate Professor at the *University of Darmstadt, Germany* and head of the research group Animation and Image Communication at the Fraunhofer Institute for Computer Graphics. Since October 2000, he is professor at the *University of Bonn, Germany* and director of the *Institute of Computer Science II*.

Funding
Supported by National Natural Science Foundation of China (Grant No. 51935003).

Competing Interests
The authors declare no competing financial interests.

Author Details
[1]School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China. [2]Institute of Computer Science II, University of Bonn, 53115 Bonn, Germany.

Received: 24 January 2021 Revised: 23 August 2021 Accepted: 3 September 2021
Published online: 30 September 2021

References

[1] E Zhang, M F Cohen, B Curless. Emptying, refurnishing, and relighting indoor spaces. *ACM Transactions on Graphics*, 2016, 35(6): 1–14.
[2] S O Escolano, C Rhemann, S Fanello, et al. Holoportation: Virtual 3D teleportation in real-time. *Proceedings of the 29th Annual Symposium on User Interface Software and Technology,* 2016: 741–754.
[3] A Mossel, M Kroter. Streaming and exploration of dynamically changing dense 3D reconstructions in immersive virtual reality. *IEEE International Symposium on Mixed and Augmented Reality*, 2016: 43–48.
[4] P Stotko, S Krumpen, M B Hullin, et al. SLAMCast: Large-scale, real-time 3D reconstruction and streaming for immersive multi-client live telepresence. *IEEE Transactions on Visualization and Computer Graphics*, 2019, 25(5): 2102–2112.
[5] G Bruder, F Steinicke, A Nuchter. Poster: Immersive point cloud virtual environments. *IEEE Symposium on 3D User Interfaces*, 2014: 161–162.
[6] P Stotko, S Krumpen, M Schwarz, et al. A VR system for immersive teleoperation and live exploration with a mobile robot. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019: 3630–3637.
[7] X Mu, W Sun, C Liu, et al. Numerical simulation and accuracy verification of surface morphology of metal materials based on fractal theory. *Materials*, 2020, 13 (18): 4158.
[8] Q Sun, B Zhao, X Liu, et al. Assembling deviation estimation based on the real mating status of assembly. *Computer-Aided Design*, 2019, 115: 244–255.
[9] X Mu, Q Sun, J Xu, et al. Feasibility analysis of the replacement of the actual machining surface by a 3D numerical simulation rough surface. *International Journal of Mechanical Sciences*, 2019, 150: 135–144.
[10] Y Furukawa, B Curless, S M Seitz, et al. Towards internet-scale multi-view stereo. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, 2010: 1434–1441.
[11] Y Zhu, J Yan. Reconstructing tree trunks by 3D bar filters. *Neurocomputing*, 2017, 253: 122–126.
[12] J Sturm, N Engelhard, F Endres, et al. A benchmark for the evaluation of RGB-D slam systems. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012: 573–580.
[13] J Geng. Structured-light 3D surface imaging: a tutorial. *Advances in Optics and Photonics*, 2011, 3(2): 128–160.
[14] G Pandey, J McBride, S Savarese, et al. Extrinsic calibration of a 3D laser scanner and an omnidirectional camera. *IFAC Proceedings*, 2010, 43 (16): 336–341.
[15] C B Choy, D Xu, J Gwak, et al. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. *European Conference on Computer Vision*, 2016, 2016: 628–644.
[16] D Eigen, C Puhrsch, R Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in Neural Information Processing Systems*, 2014: 2366–2374.
[17] A Saxena, M Sun, A Y Ng. Make3D: Learning 3D scene structure from a single still image. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(5): 824–840.
[18] D Eigen, R Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. Proceedings of the IEEE International Conference on Computer Vision, 2015: 2650–2658.
[19] W Zhuo, M Salzmann, X He, et al. Indoor scene structure analysis for single image depth estimation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2015: 614–622.
[20] R Garg, V K BG, G Carneiro, et al. Unsupervised CNN for single view depth estimation: Geometry to the rescue. *European Conference on Computer Vision*, 2016: 740–756.
[21] J Li, R Klein, A Yao. A two-streamed network for estimating fine-scaled depth maps from single RGB images. *Proceedings of the IEEE International Conference on Computer Vision*, 2017: 3372–3380.
[22] H Fu, M Gong, C Wang, et al. Deep ordinal regression network for monocular depth estimation. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018: 2002–2011.
[23] X Cheng, P Wang, R Yang. Depth estimation via affinity learned with convolutional spatial propagation network. *Proceedings of the European Conference on Computer Vision*, 2018: 103–119.
[24] J Wu, Y Wang, T Xue, et al. Marrnet: 3D shape reconstruction via 2.5 D sketches. *Advances in Neural Information Processing Systems*, 2017: 540–550.
[25] S Tulsiani, T Zhou, A A Efros, et al. Multi-view supervision for single-view reconstruction via differentiable ray consistency. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017: 2626–2634.
[26] P Henderson, V Ferrari. Learning single-image 3D reconstruction by generative modelling of shape, pose and shading. *International Journal of Computer Vision*, 2019: 1–20.
[27] M Gadelha, S Maji, R Wang. 3D shape induction from 2D views of multiple objects. *International Conference on 3D Vision*, 2017: 402–411.

[28] X Yan, J Yang, E Yumer, et al. Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. *Advances in Neural Information Processing Systems*, 2016: 1696–1704.
[29] X Li, Y Dong, P Peers, et al. Synthesizing 3D shapes from silhouette image collections using multi-projection generative adversarial networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019: 5535–5544.
[30] M Gadelha, R Wang, S Maji. Shape reconstruction using differentiable projections and deep priors. *Proceedings of the IEEE International Conference on Computer Vision*, 2019: 22–30.
[31] K Genova, F Cole, A Maschinot, et al. Unsupervised training for 3d morphable model regression. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018: 8377–8386.
[32] S Suwajanakorn, N Snavely, J J Tompson, et al. Discovery of latent 3D keypoints via end-to-end geometric reasoning. *Advances in Neural Information Processing Systems*, 2018: 2059–2070.
[33] B Gecer, S Ploumpis, I Kotsia, et al. Ganfit: Generative adversarial network fitting for high fidelity 3D face reconstruction. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019: 1155–1164.
[34] C H Lin, O Wang, B C Russell, et al. Photometric mesh optimization for video-aligned 3D object reconstruction. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019: 969–978.
[35] H Fan, H Su, L J Guibas. A point set generation network for 3D object reconstruction from a single image. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017: 605–613.
[36] Y Wei, S Liu, W Zhao, et al. Conditional single-view shape generation for multi-view stereo reconstruction. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2019: 9651–9660.
[37] C R Qi, H Su, K Mo, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017: 652–660.
[38] C R Qi, L Yi, H Su, et al. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 2017: 5099–5108.
[39] H Su, V Jampani, D Sun, et al. Splatnet: Sparse lattice networks for point cloud processing. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018: 2530–2539.
[40] Y Li, R Bu, M Sun, et al. Pointcnn: Convolution on X-transformed points. *Advances in Neural Information Processing Systems*, 2018, 31: 820–830.
[41] 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*, 2018: 2897–2905.
[42] J Gwak, C B Choy, M Chandraker, et al. Weakly supervised 3D reconstruction with adversarial constraint. *International Conference on 3D Vision*, 2017: 263–272.
[43] S A Eslami, D J Rezende, F Besse, et al. Neural scene representation and rendering. *Science*, 2018, 360 (6394): 1204–1210.
[44] V Sitzmann, M Zollhoefer, G Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 2019, 32: 1121–1132.
[45] K L Navaneet, P Mandikal, M Agarwal, et al. Capnet: Continuous approximation projection for 3D point cloud reconstruction using 2D supervision. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, 33(1): 8819–8826.
[46] C H Lin, C Kong, S Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018, 32: 1.
[47] A X Chang, T Funkhouser, L Guibas, et al. Shapenet: An information-rich 3d model repository. arXiv preprint arXiv: 1512.03012, 2015.
[48] X Sun, J Wu, X Zhang, et al. Pix3d: Dataset and methods for single-image 3d shape modeling. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018: 2974–2983.
[49] H O Song, Y Xiang, S Jegelka, et al. Deep metric learning via lifted structured feature embedding. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016: 4004–4012.
[50] A Tewari, M Zollhofer, H Kim, et al. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017: 1274–1283.