Efficient Dense Point Cloud Object Reconstruction Using Deformation Vector Fields

Kejie Li[1,2][0000−0001−8821−7762], Trung Pham[3] *[0000−0002−8039−282X], Huangying Zhan[1,2][0000−0002−2899−8314], and Ian Reid[1,2][0000−0001−7790−6423]

[1] The University of Adelaide
[2] Australian Center for Robotic Vision
[3] NVIDIA
{kejie.li, huangying.zhan, ian.reid}@adelaide.edu.au
{trungp}@nvidia.com

Abstract. Some existing CNN-based methods for single-view 3D object reconstruction represent a 3D object as either a 3D voxel occupancy grid or multiple depth-mask image pairs. However, these representations are inefficient since empty voxels or background pixels are wasteful. We propose a novel approach that addresses this limitation by replacing masks with "deformation-fields". Given a single image at an arbitrary viewpoint, a CNN predicts multiple surfaces, each in a canonical location relative to the object. Each surface comprises a depth-map and corresponding deformation-field that ensures every pixel-depth pair in the depth-map lies on the object surface. These surfaces are then fused to form the full 3D shape. During training we use a combination of per-view loss and multi-view losses. The novel multi-view loss encourages the 3D points back-projected from a particular view to be consistent across views. Extensive experiments demonstrate the efficiency and efficacy of our method on single-view 3D object reconstruction.

Keywords: 3D object reconstruction, dense point clouds, deep learning

1 Introduction

Although humans can effortlessly infer the 3D structure of an object from a single image, it is, however, an ill-posed problem in computer vision. To make it well-posed, researchers have been using hand-crafted 3D cues such as "Shape from X" (e.g., shading, texture) [4, 1, 26] , and planarity [28, 23]. More recently, there has been considerable interest in using deep networks to regress from an image to its depth [22, 8, 10, 38] for scene geometry reconstruction, and in particular from an image of an object to its 3D shape for object geometry reconstruction. There is no settled or best way to represent 3D objects, with methods including meshes [29, 17], point clouds [7, 21, 33], or voxel occupancy grids [5, 9, 32] , each having both advantages and disadvantages in terms of the efficiency and convenience of the representation, and – importantly for our purposes – for learning.

* This work was done prior to joining NVIDIA

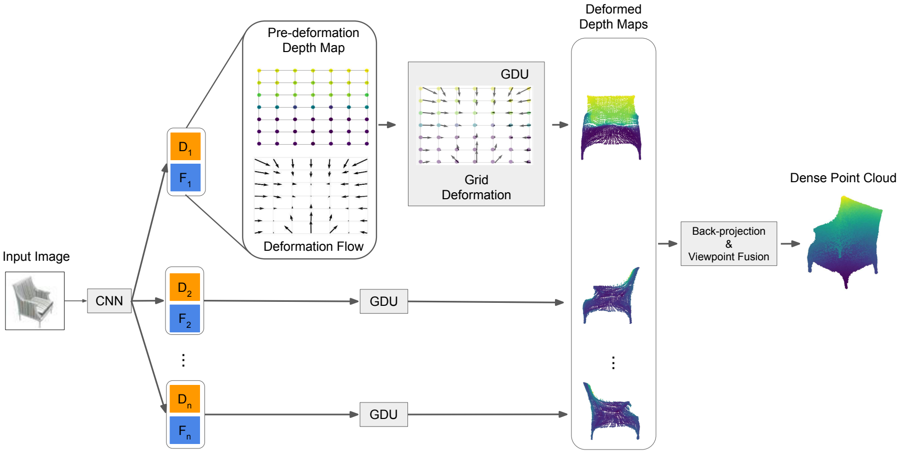


Fig. 1. The overall pipeline of our approach. Given a single RGB image of an object from an arbitrary viewpoint, the CNN outputs a set of 2D pre-deformation depth maps and corresponding deformation fields at pre-defined canonical viewpoints. These are each passed to a Grid Deformation Unit (GDU) that transforms the regular grid of the depth map to a "deformed depth map". Finally, we transform the deformed depth maps into a common coordinate frame to fuse all 3D points into a single dense point cloud. The colors of points indicate different depth values, and the arrow directions represent the image grid deformation (for visualization effect, the magnitude of arrow is not actual scale, the image size is also scaled down)

For volumetric representation methods, most existing CNN-based methods simply extend 2D deconvolutions to 3D deconvolutions on 3D regular grids, as done in [5, 9]. For each voxel, the network predicts the score of being occupied by the object. Thresholding the volumetric score map results in a 3D occupancy object representation. Nevertheless, 3D volumetric representations are very expensive in both computation and memory when working with deep networks — the required memory increases cubically with the grid resolution. Importantly, only a small portion of the voxels (less than 10%) are occupied by the object, leaving the rest wasteful. Consequentially, most existing methods are only able to reconstruct objects with low resolutions (i.e., 32x32x32), thus losing the surface granularity. Furthermore, it is generally non-trivial to find a suitable threshold to generate precise object surfaces for different object classes or even different objects in the same class.

Intuitively, it is more efficient to predict the surface of an object directly, rather than the whole 3D space. Fan *et al.* [7] proposed Point Set Generation Net (PSGN), where the surface is represented by an orderless point cloud. As point clouds are unstructured, computing the training loss is highly expensive — for instance, for N prediction points along with N ground-truth points, the complexity of the Chamfer Distance used in [7] is $O(N^2)$. To tackle this issue, works from Lin *et al.* [21] and Tatarchenko *et al.* [33] used a set of depth maps from different viewpoints relative to the object, which are easily fused into a point cloud. In this representation, depth is used for supervision, rather than point-wise distance between orderless predicted and ground-truth point clouds. However, the predicted depth maps from a deep neural net inherently cover not only object points but also unnecessary background. To classify foreground and background points, [21, 33] also predicted a binary (i.e., foreground/background) segmentation mask for each depth map, where each pixel is the score of being foreground. Similar to the 3D volumetric representation, this depth-mask representation is also inefficient as background points are unused, and also suffers from the non-trivial foreground/background thresholding.

Moreover, learning to regress depth from images often generates noisy points around the surface [33]. The fusion of multiple partial-view point clouds escalates the noise. In [21], they propose to improve the quality of fused point clouds by a multi-view consistency supervision based on binary masks and depth maps. The idea is to project the predicted 3D point cloud to novel viewpoints to generate new depth maps and masks at these viewpoints, which are then supervised by the corresponding ground-truth depth maps and binary masks. However, this supervision, being similar to the shape from silhouette technique [24] and voxel based multi-view consistency supervision in deep learning based methods [37, 34], encourages masking out the points projected to the background. This further reduces the density of the predicted point clouds and harms the surface coverage.

In this paper, we present a novel and highly efficient framework (shown in Fig. 1) to generate dense point clouds for representing the 3D shape of objects. Given a single image of an object of interest, taken from an arbitrary viewpoint, our network generates multiple partial surfaces of the object, each at a pre-defined canonical viewpoint. Although it looks similar to the depth-mask representation as a multi-view representation, each surface is defined by a depth map and corresponding deformation field (instead of a binary mask). In the Grid Deformation Unit (GDU), a point on the surface is obtained by first shifting a pixel on the depth map image grid by the amount given by the deformation field and then back-projecting (to the corresponding depth). The resulting set of points can then be considered a (dense) point cloud, though it is not an orderless one. The final 3D object representation is obtained by fusing the point-cloud surfaces into a single point cloud.

Both the depth maps and the deformation fields are regressed from the original image using a deep network trained to predict the set of canonical views. At training time we use a combination of per-view and multi-view losses. Our unique representation ensures that the per-view loss can be evaluated in $\mathrm{O}(n)$ time (where n is the number of points) because there is no need to establish correspondence between predicted and ground-truth depths. This in contrast to, for instance, Chamfer Distance usually required for unordered point-sets, leading to $\mathrm{O}(n^2)$ complexity.

The novel multi-view loss encourages the 3D points back-projected from a particular view to be consistent across novel views. More specifically when a predicted 3D point is re-projected into a novel viewpoint but falls outside of the object silhouette, our network incurs a loss based on the distance of the point to the boundary, rather than penalizing a binary cross entropy loss as done in [21, 37, 34]. Our extensive experiments demonstrate that using these combined per-view and multi-view losses yields more accurate and dense point cloud representations than any previous method.

Our contributions in this paper are summarized as follows:

- We propose a novel deformed depth map representation for 3D object reconstruction based on multiple canonical views that is efficient and bypasses foreground/background thresholding that lead to structural errors;
- We show how this representation can be effectively regressed using a deep network from a single view;
- We introduce a novel loss for our network that combines a per-view loss – that can be efficiently calculated thanks to our unique representation – with a novel multi-view loss based a distance field.
- We evaluate our method extensively showing more accurate and denser point clouds than current state-of-the-art methods. We include ablation experiments that demonstrate the value of the contributions above separately.

2 Related Work

3D reconstruction from single images has been a long-standing question in computer vision community. While a single image can provide abundant information about scene or object appearance, it barely provides any information about 3D geometry. Therefore, one has to resort to other source of information as additional input for 3D reconstruction.

The use of additional images is a typical example. This branch of works try to find the geometry correspondence between views to recover geometry, such as SfM [13] and SLAM [25]. However, these methods require dense viewpoints because the local appearance has to be preserved for feature matching. To relax the constraint of dense viewpoints, silhouette carving [24, 19] and space carving [20] have been proposed. These methods feature the downsides of failure on concave structures and multiple views needed.

Another type of additional information is prior knowledge. Using prior knowledge improves resilience to incorrect feature matching and concavity (e.g., chairs should be concave between two arms). Some prior works used simple geometry entities as shape prior [27, 2]. Recently, Kar *et al.* [16] leveraged the strong regularity of objects. For a specific object category, they learn deformable templates from a large collection of images with object of interest presented and the corresponding segmentation masks. Dame *et al.* [6] proposed a framework that combines a SLAM with deformable object templates. Rather than learning a single or a few deformable templates, methods like Huang *et al.* [12] and Kurenkov *et al.* [18] used image features to retrieve similar 3D shapes, from which they deform to the target shapes.

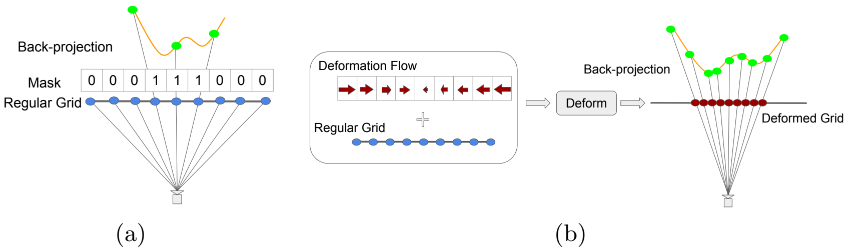
Even though our method also uses deformation, the differences between our deformation and that of their approaches are in twofold: firstly, we perform 2D deformation on an image grid, such that the deformed image grid matches the object silhouette, while they deform the 3D shape directly. More importantly, they perform deformation on 3D basis models with small variants while ours deforms a regular grid into any 2D shape.

Since large repositories of CAD models become available (e.g., ShapeNet [**?**]), it is easy to render vast amount of 2D images from CAD models [30]. The large number of 2D-3D ground-truth pairs make it seamless to use a powerful yet data hungry framework — deep neural net. Different deep learning based methods to generate 3D object shapes from single images are presented.

The pioneering works are from Choy *et al.* [5] and Girdhar *et al.* [9] who use 3D CNNs to perform voxel reconstruction that is limited to low resolution voxel grids. Octree data structure [32, 11] and Discrete Cosine Transform technique [15] have been used to scale up the voxel grid. More recently, Fan *et al.* [7] proposed an alternative approach that predicts an orderless point cloud to shape the surface of objects directly. Nevertheless, this method is limited to a sparse point cloud because 1) the number of learnable parameters increases linearly as the number of predicted points and 2) the direct 3D distance metrics (e.g., Chamfer Distance) are also intractable for dense point clouds. Thus, this method is not scalable in terms of memory and training time.

The most relevant works to us are [21, 33]. We all advocate that in order to generate dense point clouds, one should resort to partial surfaces each represented by a structured point cloud. However, the fundamental difference between our approach and that of [21, 33] is how to shape these surfaces. In their methods, they shape an object surface by predicting a binary mask along with the depth map to filter out points that are not back-projected to the surface in 3D space. Although Lin *et al.* [21] relaxed their network to predict x and y coordinates along with a depth map for more flexibility, they still rely on a binary mask to shape a 3D surface. The side effects of these depth-mask based approaches are firstly, it is a huge computation waste as a large number of points are discarded, especially for objects with thin structure such as lamp, airplane and chairs; secondly, foreground/background thresholding inherits the thresholding issue from 3D voxel grid representation. Instead, we predict the surface directly by deforming the regular depth map to circumvent issues above.

Moreover, although Lin *et al.* [21] have realized that the fusion of multiple partial surfaces generates noisy points and thus developed a multi-view consistency supervision based on binary masks and depths to address this issue. However, the binary cross entropy penalty leads more points to be discarded and thus the surface coverage is sacrificed. In contrast, we develop a novel multi-view supervision framework based on a continuous distance field that does not suffer from the surface coverage trade-off, and the comparison these two supervision frameworks shows that using our multi-view supervision framework outperforms their framework significantly.



(a)

(b)

Fig. 2. An 1D example of depth-mask back-projection and deformed depth back-projection. Blue points are pixel-depth pairs on 1D regular grid. Dark red arrows are deformation flow. Dark red points are pixel-depth pairs shifted by the deformation flow. Orange lines are the target 2D surface. Green points are 2D points back-projected to reconstruct the 2D surface. (a) In the depth-mask representation, because pixels are filtered out by the mask, there are less points to reconstruct the surface. (b) In our deformed depth representation, the deformed grid is align with the surface, so that all pixels are used to reconstruct the surface.

3 Method

Our goal is to learn a CNN that is able to reconstruct a dense 3D point cloud to represent 3D object shape from a single RGB image. We first introduce how we represent a partial surface of an object using a deformed depth map and the per-view supervision for the deformed depth map, followed by a multi-view consistency supervision based on distance fields. Lastly, we briefly introduce the network architecture and present the network training algorithm combining the per-view losses and multi-view loss.

3.1 Deformed Depth Map

One way to represent a view-dependent object surface is to use a depth map $\mathbf{D}$. For each pixel p at location (x_p, y_p) with a depth value z_p, we can back-project p to a 3D point $\mathbf{p}$ through an inverse perspective transformation, i.e.,

$$\mathbf{p} = \mathbf{R}^{-1}(\mathbf{K}^{-1}\left[x_p\ y_p\ z_p\right]^T - \mathbf{t}), \tag{1}$$

where $\mathbf{K}$, $\mathbf{R}$ and $\mathbf{t}$ are the camera intrinsic matrix, rotation matrix, and translation vector respectively. Learning a network to reconstruct the 3D object shape becomes learning to predict a set of depth maps, as done in [21, 33]. Note that the size of the depth images need not be equal to the size of the input RGB image. The main issue of this representation is that not all pixels are back-projected to the object's surface, therefore the network must additionally predict a binary segmentation mask for each depth map to suppress background points. The abandoned points become wasteful.

Notice that in Eq. (1), the pixel locations (x_p, y_p) are fixed in a regular image grid, which is not flexible to model the object's surface. Our insight is

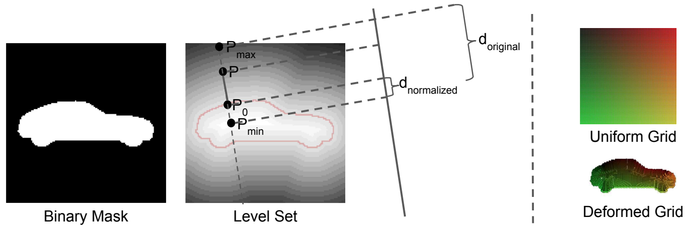


Fig. 3. Left image is a binary mask; middle image is the corresponding level set where the silhouette boundary is shown in red line. To define the deformed position for point p, the closest point on the boundary p_0 is located, followed by finding the point at the maximum level p_{max} and point at the minimum level p_{min} along the line of p and p_0. Then, p should be normalized to the range between p_0 and p_{min}. Right images show a pair of uniform grid and its corresponding deformed grid. The color indicates point correspondence before and after deformation.

that regardless of depth values, the projection ray of every pixel should hit the object's surface. This can be achieved by predicting a deformation flow (vector) $[\mathbf{U}, \mathbf{V}]$ for each pixel p (an 1D illustration is presented in Fig. 2). Specifically, for each pixel p at pixel location (x_p, y_p), our network predicts a deformation vector $[u_p, v_p]$ (besides its depth value z_p). The position of this pixel is shifted by the deformation flow. Then the new position of this pixel after deformation is $x'_p = x_p + u_p$, $y'_p = y_p + v_p$. The same inverse perspective transformation can be applied on the deformed depth map to back-project to 3D space,

$$\mathbf{p} = \mathbf{R}^{-1}(\mathbf{K}^{-1} \left[x'_p \; y'_p \; z_p\right]^T - \mathbf{t}). \tag{2}$$

Training Losses for Deformed Depth Map During training, the deformation flow is supervised by a pseudo ground-truth (see the section below). The pixel-wise L_1 deformation flow losses for x and y directions are given below,

$$L_U = \|\mathbf{U} - \mathbf{U_{gt}}\|_1 \quad L_V = \|\mathbf{V} - \mathbf{V_{gt}}\|_1. \tag{3}$$

where the $\mathbf{U_{gt}}$ and $\mathbf{V_{gt}}$ are the ground-truth deformation fields on x-direction and y-direction respectively.

However, the direct pixel-wise loss cannot be used between a regular ground-truth depth map and a deformed depth map as pixel-wise correspondences have been changed due to deformation. To supervise the deformed depth map, we use the pseudo ground-truth deformation flow to deform the ground-truth depth map to obtain the deformed ground-truth depth map, such that the pixel-wise loss can be used, which is given below,

$$L_d = \|\mathbf{D}' - \mathbf{D}'_{\mathbf{gt}}\|, \tag{4}$$

where the $\mathbf{D}'_{\mathbf{gt}}$ and $\mathbf{D}'$ are the deformed ground-truth and predicted depth respectively.

Pseudo Ground-truth Deformation Flow We define a function that takes an object binary mask (silhouette as foreground and the rest as background) on a regular grid as input, and outputs a vector field for deformation. This vector field is treated as the pseudo ground-truth for the deformation flow. The criteria of this function are 1) every pixel should be shifted into the silhouette (i.e., the regular grid should be deformed to fit the silhouette), 2) the *deformed* grid should be uniformly dense.

More specifically, we first convert the binary mask into a level set, where inside silhouette is negative levels, background is positive levels and the silhouette boundary is at the zero level set (shown by red line in Fig. 3). For each pixel p at coordinate (x_p, y_p) on the regular grid, it finds its closest pixel at the zero level set (silhouette boundary) called p_0. The deformation direction for a pixel outside of the silhouette is $\overrightarrow{\mathbf{pp_0}}$, and that of a point inside of the silhouette is $\overrightarrow{\mathbf{p_0p}}$.

After the direction is determined, we then calculate the magnitude for deformation. As illustrated in Fig. 3, along the line of p and p_0, we find the local maximum point p_{max} and the local minimum point p_{min} in the level set. The deformation flow for pixel p is defined below,

$$x'_p = x_p \frac{\|x_{min} - x_0\|}{\|x_{max} - x_{min}\|}, \quad y'_p = y_p \frac{\|y_{min} - y_0\|}{\|y_{max} - y_{min}\|} \tag{5}$$

$$\mathbf{U}[x_p, y_p] = x'_p - x_p, \quad \mathbf{V}[x_p, y_p] = y'_p - y_p. \tag{6}$$

The Eq. (5) ensures that a point along the line between p_{max} and p_{min} moves to a point on the line between p_0 and p_{min}, such that the pixel is in the silhouette (the first criteria satisfied). Moreover, no pixels are collided (the second criteria satisfied).

3.2 Distance Field Multi-view Consistency

As mentioned earlier, the 3D points back-projected from a predicted depth map are often noisy viewed from other viewpoints. Fig. 4(a) visualizes this problem, where the point cloud back-projected from the front view of a chair contains many noisy points between the front and back legs of the chair. To alleviate this problem, we introduce a novel multi-view consistency supervision, which encourages the 3D points to project into the object silhouette (i.e., foreground) but not the background at novel viewpoints.

To that end, we transform ground-truth binary masks (at novel viewpoints) into a distance field [3], where the values of the foreground pixels are zero in the distance field (meaning no penalty), whereas the values of the background pixels are the distance to the closest boundary. Fig. 4(b) demonstrates an example of distance field. Such distance fields are used as the supervision signal to pull outliers (i.e., points projected to the outside of the silhouettes) back to the object silhouettes (in 2D).

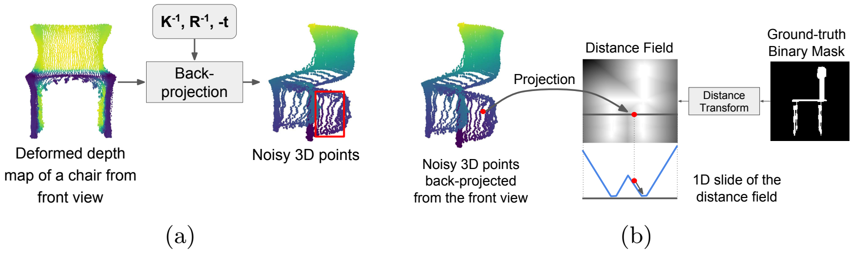

K-1, R-1, -t
Back-projection
Deformed depth map of a chair from front view
Noisy 3D points
(a)
Projection
Distance Field
Ground-truth Binary Mask
Distance Transform
Noisy 3D points back-projected from the front view
1D slide of the distance field
(b)

Fig. 4. (a) An example of the noisy reconstruction using only the depth as supervision. (b) Our distance field multi-view consistency. Given a ground-truth binary mask at a novel viewpoint, it is transformed to a distance field using distance transform [3]. 3D points are projected onto the distance field. The projection points are supervised by the multi-view consistency loss L_{df} to move toward the object silhouette. Note that in the example above, the project point move on the 1D surface only for the purpose of visualization. In reality, it moves on the 2D distance field surface.

Technically, a 3D point $\mathbf{p}$ (X_p, Y_p, Z_p) is projected to the distance field using the transformation and camera matrices at the novel viewpoint n, i.e.,

$$[x_p, y_p, 1]^T = \mathbf{K}_n(\mathbf{R}_n\mathbf{p} + \mathbf{t}_n), \tag{7}$$

where $[x_p, y_p]$ is the projection point coordinate in the distance field.

The multi-view consistency training loss L_{df} becomes:

$$L_{df} = \sum_{n}^{N}\sum_{p}^{P} L_{df}^{(n,p)}, \tag{8}$$

where N is the number of viewpoints and P is the number of 3D points. $L_{df}^{(n,p)}$ is defined as:

$$L_{df}^{(n,p)} = \sum_{h}^{H}\sum_{w}^{W} \mathbf{F}^n[h,w]\max(0, 1-|x_p-h|)\max(0, 1-|y_p-w|), \tag{9}$$

where H and W are the height and width of the distance field respectively, $\mathbf{F}^n$ is the distance field at viewpoint n, and $\mathbf{F}^n[h,w]$ is the distance value at pixel location $[h,w]$

Given a point at $[x_p, y_p]$, the values (distances) of the 4 neighboring pixels are interpolated to approximate the corresponding distance field $\mathbf{F}[x_p, y_p]$. By minimizing Eq. 9, this point is supervised to move toward the object silhouettes. This technique, called differentiable bilinear interpolation, was used in [14] for differential image warping.

3.3 Network Architecture and Training

We use an autoencoder-like network where the encoder extracts image features and project them into a latent space. The decoder is assembled by several 2D deconvolution layers to generate pairs of a pre-deformation depth map and a deformation flow map from 6 fixed viewpoints which are the faces of a cube centered at the object of interest. More details about the network architecture and training configuration can be found in the supplementary material.

We train the network with the deformed depth map loss, deformation flow loss, and the distance field loss jointly. The final loss function is

$$L = \sum_{m}^{M} (L_d^m + L_U^m + L_V^m) + \lambda \sum_{n}^{N} L_{df}^n, \tag{10}$$

where M is the 6 fixed viewpoints and N is the number of distance field from novel viewpoints.

4 Experiments

We evaluate our proposed method beginning with ablation study of key components of our framework: deformed depth map and the distance field based multi-view consistency loss, followed by comparison to the state-of-the-art methods on single view 3D object reconstruction. In addition, we test our method on a recently published real dataset to determine whether it can generalize to real images and comparison with other methods reported.

4.1 Data Preparation

Following previous methods, we use a subset of ShapeNet, which contains objects in 13 categories, to train and evaluate our network. We render 6 depth maps along with the binary masks from fixed viewpoints of 6 faces of a cube where a 3D object is centered. The binary mask is used to construct the pseudo ground-truth deformation field. Additionally, we also render 24 RGB images along with its binary mask from arbitrarily sampling azimuth and elevation in [0,360),[-20,30] respectively. The RGB images are input images to the network and the binary masks are preprocessed to a distance field for multi-view consistency loss.

4.2 Quantitative Measurement

To evaluate results quantitatively, we use the average point-wise 3D Euclidean distance called Chamfer Distance between predicted and ground-truth point clouds.

$$D(S_1, S_2) = \sum_{p_i \in S_1} \min_{p_j \in S_2} \|p_i - p_j\|^2 + \sum_{p_j \in S_2} \min_{p_i \in S_1} \|p_i - p_j\|^2 \tag{11}$$

$$D_{S_1 \to S_2} = \sum_{p_i \in S_1} \min_{p_j \in S_2} \|p_i - p_j\|^2 \tag{12}$$

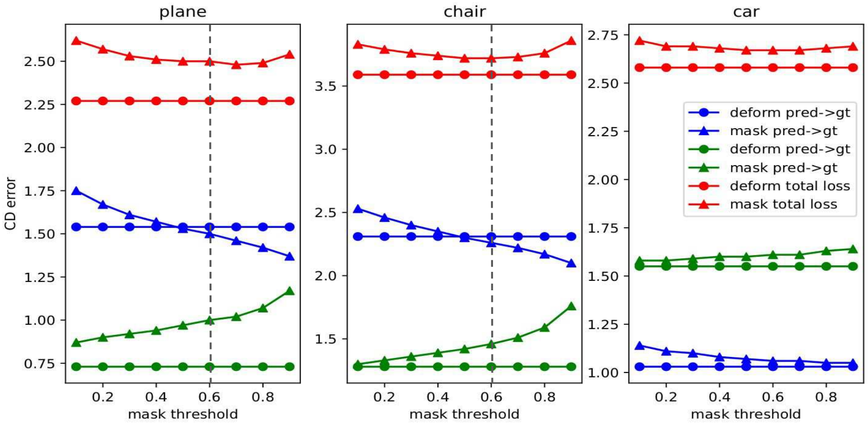


Fig. 5. Chamfer Distance comparison with different foreground/background thresholds for the depth-mask representation. The lower gt → pred loss of our method demonstrates that our method provides better coverage than the baseline. Even though the baseline can achieve lower pred → gt loss when setting the threshold higher (e.g., threshold ≥ 0.6) by only preserve points with high confidence, the penalty of coverage offset the accuracy gain and leading a higher overall loss.

$$D_{S_2 \to S_1} = \sum_{p_j \in S_2} \min_{p_i \in S_1} \|p_i - p_j\|^2 \tag{13}$$

where S_1 is the predicted point cloud and S_2 is the ground-truth point cloud.

As demonstrated by [21], while the Chamfer Distance can evaluate the overall performance, it is also essential to report the prediction to ground-truth distance Eq. (12) and ground-truth to prediction distance Eq. (13) individually as they evaluate different aspects of the prediction point cloud. The former shows how far each prediction point to the closest ground-truth point (i.e., how accurate the prediction is), and the latter reports the distance from each ground-truth point to the closest prediction point indicating the surface coverage. Note that all numbers reported in the experiment section are scaled up by 100 for readability.

4.3 Ablation Study

In this section, we evaluate two key components of our framework: the deformed depth map and the distance field based multi-view consistency loss invidiously.

Deformed Depth Map We train two networks with our proposed deformed depth representation and the depth-mask representation (as a baseline) respectively in an identical training setting.

The networks are trained and evaluated on three categories (plane, car and chair). The per category results are reported in Fig. 5. It shows that the deformed

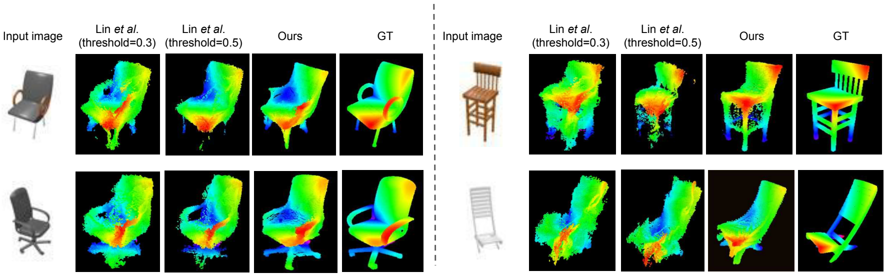

Input image
Lin et al. (threshold=0.3)
Lin et al. (threshold=0.5)
Ours
GT
Input image
Lin et al. (threshold=0.3)
Lin et al. (threshold=0.5)
Ours
GT

Fig. 6. Visual comparison to Lin *et al.* [21]

Methods	overall CD	prediction points
Binary mask multi-view loss [21]	3.240 / 3.531	31972 / 25401
Distance field multi-view loss (**Ours**)	**3.102** / **2.987**	**98304** / **98304**

Table 1. Comparison on different multi-view consistency losses. The numbers reported in overall CD and the number of prediction points are: without multi-view consistency / with multi-view consistency

depth representation consistently achieve lower loss under different threshold setting for the depth-mask baseline (shown in two red lines). There is a difficult trade-off between the prediction point accuracy and surface coverage in the baseline. When the $D_{pred \to gt}$ gradually decreases as the threshold rises (i.e., only good prediction points are preserved), the surface coverage loss increases significantly. More importantly, another disadvantage of the depth-mask representation revealed from the experiment is that it is not trivial to select an optimal threshold for different instances. To better visualize this issue, a few qualitative examples from Lin *et al.* [21] that use the depth-mask representation are given in Fig. 6.

Distance Field Multi-view Consistency To evaluate our distance field based multi-view consistency loss, we compare to a baseline in which the loss is disabled, and a prior art [21] that use a binary mask multi-view consistency loss. The results reported in Table 1 show that after applying our distance field multi-view consistency, the network performs better than the baseline. Since our consistency does not simply mask out more points to reduce outliers, our method also outperforms the binary masked multi-view consistency [21].

4.4 Comparison with Prior Art

We compare our method against previous (state-of-the-art) methods using point cloud or voxel grid representation in a synthetic dataset and a real dataset in this section.

Comparing to PSGN Because the pre-trained model of PSGN provided by the authors generates a point cloud aligned with the input image while ours is in

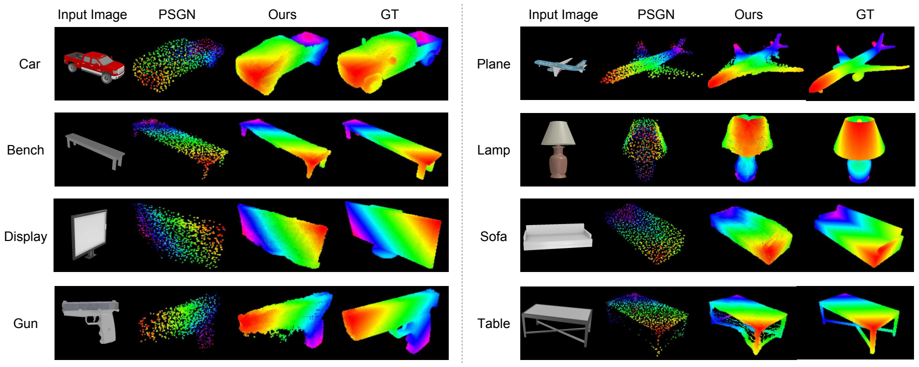


Fig. 7. Visual comparison with PSGN

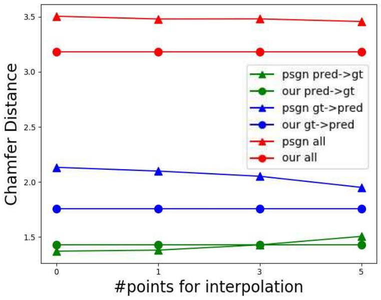


Fig. 8. Densified PSGN Comparison

Category	PSGN	Ours
plane	**2.582**	2.66
car	3.253	**2.897**
chair	4.110	**3.731**
table	3.916	**3.271**
bench	3.660	**3.357**
cabinet	4.835	**4.037**
display	5.010	**4.343**
lamp	5.105	**4.933**
speaker	5.707	**5.532**
gun	**2.949**	3.259
sofa	4.644	**4.267**
telephone	3.999	**3.588**
watercraft	3.921	**3.894**

Table 2. Comparison with PSGN on 13 classes using Chamfer Distance.

a canonical pose. To have fair comparison, we use their published code to retrain their network to output point clouds in the canonical pose. Both PSGN and our network train on our rendered input images using the same training-test split.

Our method outperforms PSGN in the overall performance in most of the categories (11 out of 13) as reported in Table 2. To resolve the measurement bias to the density of predicted points, we also report the results of a densified PSGN, where we densify the point cloud size from 1024 to 98304 (the same size of our prediction) using linear interpolation between neighboring points as a post-processing step. This densified PSGN is evaluated on five categories (chair, car, plane, bench and table) and the mean Chamfer Distance reported in Fig. 8. This graph shows that a densified/interpolated reconstruction cannot capture the finer 3D details that our method can. To contrast both methods visually, we present some qualitative examples in Fig. 7.

Comparing to Hierarchical Surface Prediction (HSP) [11] As PSGN has shown superior performance of point cloud representation over low resolution voxel grid representation (e.g., 3D-R2N2), we provide a quantitative comparison

Methods	Chair	Plane	Car	Table	Bench
HSP	4.716	3.878	3.487	4.072	4.467
Ours	**3.731**	**2.660**	**2.897**	**3.271**	**3.357**

Table 3. Chamfer Distance between ours and HSP

Methods	3D-R2N2	PSGN	3D-VAE-GAN	DRC	MarrNet	AtlasNet	Pix3D	Ours	Pix3D
	[5]		[36]	[34]	[35]	[?]	[31]		(w/ pose)
EMD	0.211	0.216	0.176	0.144	0.136	0.128	**0.124**	**0.124**	0.118
CD	0.239	0.200	0.182	0.160	0.144	0.125	**0.124**	0.125	0.119

Table 4. Real images evaluated using Earth Movers Distance and Chamfer Distance

between our method and HSP in Table 3. HSP up-scales the voxel grid to 512^3 by using Octree data structure. Five categories of the ShapeNet are evaluated using input images and a pre-trained model provided by the author of HSP. To calculate Chamfer Distance for HSP outcomes, we generate a mesh from the voxels using the marching cube algorithm and sample uniformly the same number of points as ours from the mesh. Although the Octree method increases the resolution efficiently, it still suffers from the non-trivial occupancy thresholding and the "structurally unaware" cross entropy loss (i.e., missing thin structures), leading to poorer performance than ours.

4.5 Generalization to Real Images

Pix3D [31] (a large-scale real dataset with high quality image-shape pairs) has become available and we compare our results to that benchmark in Table 4. We train our method on synthetic images with backgrounds randomly selected from the SUN dataset, then evaluate on Pix3D images directly. Table 4 shows our method generalizes well to real images and achieves comparable performance to the state-of-the-art method. Note that the best performance reported in Pix3D uses joint training of object pose and 3D shape to boost the network performance, and thus excluded from the comparison.

5 Conclusions

In this work, we present a novel deformed depth representation. By using deformation, we bypass the need of foreground/background thresholding leading to denser point clouds and reconstruction with high fidelity. Moreover, to refine the fused point cloud, we propose a distance field based multi-view consistency which outperforms the existing multi-view consistency loss. Our completed framework outperforms the prior art methods in single-view object reconstruction. However, the expedient approach that we adopted in the current paper can be replaced by an energy optimization that might lead to a more uniformly distributed deformed grid.

Acknowledgement This work was supported by the UoA Scholarship to KL and HZ, the ARC Laureate Fellowship FL130100102 to IR and the Australian Centre of Excellence for Robotic Vision CE140100016.

References

1. Aloimonos, J.: Shape from texture. Biological cybernetics **58**(5), 345–360 (1988)
2. Biederman, I.: Recognition-by-components: a theory of human image understanding. Psychological review **94**(2), 115 (1987)
3. Borgefors, G.: Distance transformations in digital images. Computer vision, graphics, and image processing **34**(3), 344–371 (1986)
4. Braunstein, M.L., Liter, J.C., Tittle, J.S.: Recovering three-dimensional shape from perspective translations and orthographic rotations. Journal of Experimental Psychology: Human Perception and Performance **19**(3), 598 (1993)
5. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: European Conference on Computer Vision. pp. 628–644. Springer (2016)
6. Dame, A., Prisacariu, V.A., Ren, C.Y., Reid, I.: Dense reconstruction using 3d object shape priors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1288–1295. IEEE (2013)
7. Fan, H., Su, H., Guibas, L.: A point set generation network for 3d object reconstruction from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. vol. 2, p. 6 (2017)
8. Garg, R., BG, V.K., Carneiro, G., Reid, I.: Unsupervised cnn for single view depth estimation: Geometry to the rescue. In: European Conference on Computer Vision. pp. 740–756. Springer (2016)
9. Girdhar, R., Fouhey, D.F., Rodriguez, M., Gupta, A.: Learning a predictable and generative vector representation for objects. In: European Conference on Computer Vision. pp. 484–499. Springer (2016)
10. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. vol. 2, p. 7 (2017)
11. Häne, C., Tulsiani, S., Malik, J.: Hierarchical surface prediction for 3d object reconstruction. In: 3D Vision (3DV), 2017 International Conference on. pp. 412–420. IEEE (2017)
12. Huang, Q., Wang, H., Koltun, V.: Single-view reconstruction via joint analysis of image and shape collections. ACM Transactions on Graphics (TOG) **34**(4), 87 (2015)
13. Hming, Klaus, P.G.: The structure-from-motion reconstruction pipeline a survey with focus on short image sequences. Kybernetika **46**(5), 926–937 (2010), http://eudml.org/doc/197165
14. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: Advances in neural information processing systems. pp. 2017–2025 (2015)
15. Johnston, A., Garg, R., Carneiro, G., Reid, I., van den Hengel, A.: Scaling cnns for high resolution volumetric reconstruction from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 939–948 (2017)
16. Kar, A., Tulsiani, S., Carreira, J., Malik, J.: Category-specific object reconstruction from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1966–1974 (2015)
17. Kong, C., Lin, C.H., Lucey, S.: Using locally corresponding cad models for dense 3d reconstructions from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. vol. 2 (2017)

18. Kurenkov, A., Ji, J., Garg, A., Mehta, V., Gwak, J., Choy, C., Savarese, S.: Deformnet: Free-form deformation network for 3d shape reconstruction from a single image. arXiv preprint arXiv:1708.04672 (2017)
19. Kutulakos, K.N.: Shape from the light field boundary. In: Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on. pp. 53–59. IEEE (1997)
20. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. International journal of computer vision **38**(3), 199–218 (2000)
21. Lin, C.H., Kong, C., Lucey, S.: Learning efficient point cloud generation for dense 3d object reconstruction. In: AAAI Conference on Artificial Intelligence (AAAI) (2018)
22. Liu, F., Shen, C., Lin, G.: Deep convolutional neural fields for depth estimation from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5162–5170 (2015)
23. Liu, M., Salzmann, M., He, X.: Discrete-continuous depth estimation from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 716–723. IEEE (2014)
24. Martin, W.N., Aggarwal, J.K.: Volumetric descriptions of objects from multiple views. IEEE transactions on pattern analysis and machine intelligence (2), 150–158 (1983)
25. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: Dtam: Dense tracking and mapping in real-time. In: Computer Vision (ICCV), 2011 IEEE International Conference on. pp. 2320–2327. IEEE (2011)
26. Prados, E., Faugeras, O.: Shape from shading. In: Handbook of mathematical models in computer vision, pp. 375–388. Springer (2006)
27. Roberts, L.G.: Machine perception of three-dimensional solids. Ph.D. thesis, Massachusetts Institute of Technology (1963)
28. Saxena, A., Sun, M., Ng, A.Y.: Make3d: Depth perception from a single still image. In: AAAI. pp. 1571–1576 (2008)
29. Sinha, A., Unmesh, A., Huang, Q., Ramani, K.: Surfnet: Generating 3d shape surfaces using deep residual networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
30. Su, H., Qi, C.R., Li, Y., Guibas, L.J.: Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2686–2694 (2015)
31. Sun, *et al*: Pix3d: Dataset and methods for single-image 3d shape modeling. In: CVPR (2018)
32. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In: IEEE International Conference on Computer Vision (ICCV) (2017), http://lmb.informatik.uni-freiburg.de/Publications/2017/TDB17b
33. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Multi-view 3d models from single images with a convolutional network. In: European Conference on Computer Vision. pp. 322–337. Springer (2016)
34. Tulsiani, S., Zhou, T., Efros, A.A., Malik, J.: Multi-view supervision for single-view reconstruction via differentiable ray consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. vol. 1, p. 3 (2017)
35. Wu, J., Wang, Y., Xue, T., Sun, X., Freeman, B., Tenenbaum, J.: Marrnet: 3d shape reconstruction via 2.5 d sketches. In: Advances in neural information processing systems. pp. 540–550 (2017)

36. Wu, J., Zhang, C., Xue, T., Freeman, B., Tenenbaum, J.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In: Advances in Neural Information Processing Systems. pp. 82–90 (2016)
37. Yan, X., Yang, J., Yumer, E., Guo, Y., Lee, H.: Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In: Advances in Neural Information Processing Systems. pp. 1696–1704 (2016)
38. Zhan, H., Garg, R., Weerasekera, C.S., Li, K., Agarwal, H., Reid, I.: Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 340–349 (2018)