Single View 3D Point Cloud Reconstruction using Novel View Synthesis and Self-Supervised Depth Estimation

Adrian Johnston
Australian Institute for Machine Learning
School of Computer Science, University of Adelaide
Adelaide, Australia
adrian.johnston@adelaide.edu.au

Gustavo Carneiro
Australian Institute for Machine Learning
School of Computer Science, University of Adelaide
Adelaide, Australia
gustavo.carneiro@adelaide.edu.au

***Abstract*—Capturing large amounts of accurate and diverse 3D data for training is often time consuming and expensive, either requiring many hours of artist time to model each object, or to scan from real world objects using depth sensors or structure from motion techniques. To address this problem, we present a method for reconstructing 3D textured point clouds from single input images without any 3D ground truth training data. We recast the problem of 3D point cloud estimation as that of performing two separate processes, a novel view synthesis and a depth/shape estimation from the novel view images. To train our models we leverage the recent advances in deep generative modelling and self-supervised learning. We show that our method outperforms recent supervised methods, and achieves state of the art results when compared with another recently proposed unsupervised method. Furthermore, we show that our method is capable of recovering textural information which is often missing from many previous approaches that rely on supervision.**

***Index Terms*—Deep Learning, 3D Reconstruction, Deep Generative Modelling, Self-Supervised Learning, Depth Estimation**[1]

I. Introduction

Reconstruction of the 3D world from images has been one of the most studied problems in computer vision [1]. Early works which focused on part-based reconstruction using simple geometric shapes [2], multi-view reconstruction using space carving [3], or 3D shape recovery from shading [1], have led to reasonable quality 3-D reconstructions. In more recent years, with the development and standardisation of deep learning [4] in computer vision, researchers and practitioners have focused on applying these techniques to perform single-view [5]–[9] and multi-view [8], [10], [11] reconstruction. These methods often employ 3D volumetric representations as they are easily adapted from existing 2D convolutional neural networks (CNN), due to the inherent similarities between 2D images and 3D volumes. Many of the architectures and methods used on 2D images can be "lifted" into 3D by replacing the 2D convolutions with 3D convolutions. However, using volumetric representations in the deep learning framework tend to be limited in terms of quality due to computational inefficiencies. The volumetric representation is information sparse, where 3D shapes are represented by a binary occupancy grid or a signed distance field. This representation contains a substantial amount of redundancy, with most of the information concentrated at the surface voxels. Many follow-up papers have focused on improving 3D CNNs by exploiting the fact that most of the information is concentrated at the surface voxels – this idea has lead to improvements in training time and volumetric resolution [6], [12]. Newer papers [7], [13] focus on using a point cloud representation, which allow more precise reconstruction with less memory usage. Furthermore, many of the existing systems only try to recover the geometry of the 3D shape, while completely ignoring the textural information. With the large increase in access to data and improved computational resources, "learning to reconstruct" has become the standard method for single-view 3D reconstruction [5], [7], [10], [13]. However, 3D object data sets are still limited and have varying quality due to being hand modelled by artists. Unlike 2D image data, capturing real world accurate and varied ground truth 3D data is difficult, time consuming and error prone. Our goal is to simultaneously recover both the 3D shape and texture information for a specific object without any 3D supervision. We aim to use only a data set of 2D images to perform our single-view 3D reconstruction. This is achieved by using advances in self-supervised depth estimation [14], [15] and deep generative modelling [16]–[18]. Our contributions are as follows:

1) We develop a novel framework for single view self-supervised 3D point cloud reconstruction using an image based shape representation;
2) Our method is capable of generating both shape and textural information from a single-view by leveraging advances in deep generative modelling and self-

supervised learning; and

3) Our combined novel view synthesis and self-supervised depth estimators are capable of outperforming previous state of the art fully supervised methods on the ShapeNet [19] car dataset.

II. Related Work

Deep learning based shape priors [5], [13] take one of the following three shape representations: volumetric, point cloud or polygonal mesh. We will discuss in detail the most relevant methods to our own, which are based on volumetric and point cloud representations. Volumetric shape representation represents a 3D shape as either a binary occupancy grid or as a signed distance field, where grid cell values represent volume occupancy indicators [5] or the distances to the zero level set [1] respectively. Point clouds are made up of a set of 3D coordinates that represent point samples along either the surface of the shape or within the convex hull of the shape. Polygonal meshes, which are typically used in computer graphics, represent a shape with a collection of vertices, edges and faces, where each face consists of triangles or quadrilaterals. When learning shape priors, each of these representations have pros and cons. Volumetric shapes are easy to represent in the deep learning framework as they are analogous in many ways to 2D images, but they are costly in terms of memory usage [6], [12] and training/inference time [6]. Many recent works have focused on trying to improve the efficiency of volumetric representations, through Octrees [12], frequency domain compression via the Discrete Cosine Transform [6] and sparse convolutions [20]. Point cloud representations are easy to work with in the geometric deep learning framework as projections/unprojections and transformations can be implemented with simple matrix multiplications. However, they usually only have a small fixed number of points to represent the shape, meaning that the 3D shapes have limited quality. This is due to the fact that a set of points has to be represented by a fully connected layer [21] or a recurrent neural network [13], where each point is predicted based on previously predicted points in an auto-regressive manner.

A. Single View 3D Reconstruction

Recovering a 3D shape from a 2D image has been a long-standing goal in the field of computer vision [1]. While many traditional methods, such as structure from motion [1] and multi-view stereo [1] rely on many different views to recover a 3D shape or scene, deep learning based methods aim to "learn to reconstruct" by using large collections of corresponding 2D images and 3D shapes [19]. This is typically done by learning an encoder-decoder model [10], where the encoder is a 2D CNN and the decoder is a 3D deconvolutional neural network. The encoder first produces a representation of the 3D shape, which the decoder then uses to conditionally generate a 3D volume. Using 3D convolutions to recover the shape has many drawbacks. For instance, the volumetric representation limits the reconstructions in terms of shape resolution because of the computational cost of scaling up the representation. Therefore, it is difficult to train models with a high resolution volumetric representation. Furthermore, volumetric representations are inherently sparse – several works have focused on exploiting this fact to improve the performance of volumetric reconstruction methods. Riegler *et al.* [12] redefine the typical 3D convolution and deconvolution operations by a sparse convolution, which uses an Octree to reduce the dimensionality and thereby improve performance. Johnston *et al.* [6] replace the the 3D deconvolution layers by an inverse discrete cosine transform layer that allows the network to learn the coefficients of the underlying compressed 3D volume, resulting in an order of magnitude improvement in training time, resolution and memory efficiency. To deal with the limited resolution and computation cost of the volumetric representation, Fan *et al.* [13] proposed to instead use a point cloud representation. A deep 2D encoder similar to [10], is used to encode the image and a combination of 2D deconvolution and fully connected layers are merged to predict a fixed number of 3D points. Another drawback of the naive 3D volumetric representation is the inability to reason about the underlying geometry of the object. Recently, Xinchen *et al.* [8] presented a method that uses a perspective transformation to project the underlying volume back into a silhouette. They then use a loss function to penalize the projected voxels that are inconsistent with the silhouettes of adjacent views. However, this formulation has difficulty representing concave surfaces as only the visual hull of the object will be projected when computing the loss. To improve upon this method, Tulsiani *et al.* [11] show a method for adding geometric reasoning, by incorporating a differentiable ray consistency operation, which relaxes the problem and treats the voxel occupancy and projection as a probabilistic grid. This allows the model to handle more complex shapes.

Insafutdinov *et al.* [9] further refine this idea by proposing a CNN that predicts a fixed size point cloud, which is then converted to a probabilistic voxel occupancy grid [11]. Instead of regressing directly for the 3D points, the authors propose to use an unsupervised/self-supervised loss function, where the points are projected back to an image, via a differentiable projection function at random views. This allows the model to use a point cloud re-projection loss to train the network in a self-supervised manner. Furthermore, Insafutdinov *et al.* [9] extend their method by jointly learning an ensemble of pose estimators, such that their method can be trained on images from any pose. However, their model performs significantly worse when relying on such estimated poses for the re-projection loss.

As point clouds are represented as an un-ordered set of 3D points, fully connected layers or recurrent neural networks can be used as the output layer for predicting the points [13], [21]. However, in practice, this limits the density of the point cloud as the number of parameters in the output layer increases linearly with the number of points. Rather than predicting an un-ordered set of points or a 3D volume, Lin *et al.* [7] propose to supervise for depth at a set of given fixed views. Rather than directly regressing for depth, they propose to apply a

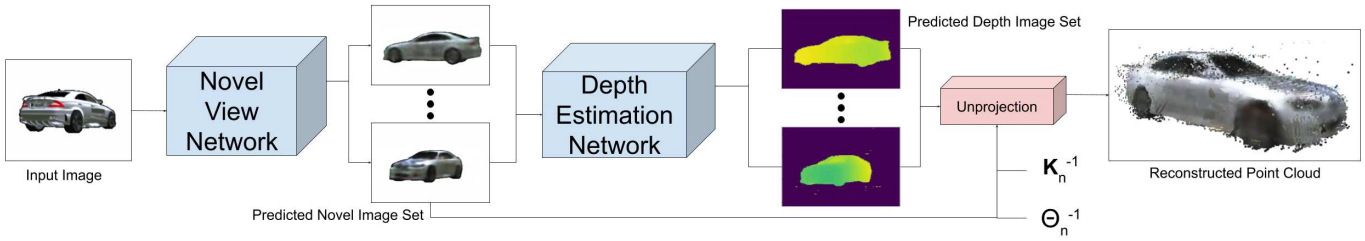

Fig. 1. To recover a 3D point cloud from a single image, the Novel View prediction model predicts a set of novel view images at a fixed view points. These novel views are then passed through a Depth Estimation model, where depths are estimated for each of the images in the set. The predicted RGB-D images are then unprojected using the inverse camera intrinsics matrix $\mathbf{K}_n^{-1}$ and the inverse object pose Θ_n^{-1}, for each novel view point n

"pseudo-rendering" function, where the predicted depth maps are first un-projected into a point cloud, then the point cloud is re-projected back into another set of depth maps at random different viewpoints. The supervised loss is then computed against these new depth maps, which forces the model to learn to create a consistent 3D shape. While generating excellent dense point clouds, this method requires the capturing of large amounts of multi-view depth images, which in practice for real data would be prohibitive. Furthermore, this method is unable to recover the underlying 3D texture of the object. Unlike [9], we do not represent our point cloud as a fixed number of points. Instead, we represent our 3D shape using a fixed number of views of an object. We train two separate networks, the first of which predicts N fixed novel views of an object given a single input image. The second network is trained to predict the depth for a given input image. Similarly to [7], we use a depth representation for our 3D shapes. However, we do not supervise for the depth maps, rather we use a photometric image warping loss to train our depth network in a self-supervised manner.

B. Self-Supervised Depth Estimation

In the standard supervised setting, a convolutional neural network is used to estimate depth by supervising against ground truth depth maps captured from any form of depth sensor e.g. Microsoft Kinect, Stereo Depth Maps, LIDAR etc. However, each of these sensors have limitations with regards to range and operating compatibility (e.g. weather or lighting conditions). Furthermore, ground truth RGB-D data is still limited in variety and size when compared with RGB image data sets. Recent works have shown that it is possible to self-supervise neural networks such that they can implicitly solve the task of interest. This is achieved by using a proxy loss function that solves a closely related problem. This allows networks to be trained from scratch on large collections of unlabelled data. In the case of self-supervised depth models, a photometric error based on differentiable image warping [22] and re-projection is used to implicitly train the network to predict depth. Garg *et al.* [14] show the earliest example of self-supervised depth estimation, performed by using synchronized stereo pairs. These results were further improved by Godard *et al.* [15] with the addition of a left-right consistency term. The photometric loss is extended to compute the loss bidirectionally from left to right and right to left for both images in the pair, ensuring consistency between the depths. Further work in this area has relaxed the requirement of needing stereo pairs, by using monocular video. More specifically, instead of using the stereo information for self-supervision, a second neural network simultaneously predicts the camera pose between frames in the input video [23], [24] and image warping is performed between successive frames. As these methods rely on predicted pose values, they are typically worse than the stereo based methods [14], [23], [24].

C. Novel View Synthesis

Novel view synthesis is an image based rendering technique, where instead of using a traditional graphics engine, like those found in many 3D applications (e.g. video games, architectural visualization), a model is used to approximate the rendering function. In recent works, an encoder-decoder CNN is used to approximate the rendering function [25], [26]. Alternatively, Zhou *et al.* [25] formulate the problem as that of regressing the 2D optical flow field that transforms the input image into the selected target image. In *Transformation Grounded Image Generation Network for Novel 3D View Synthesis* (TVSN) [26], this idea is extended to also include a term to predict the visibility of each pixel. Using this visibility map they mask the occluded pixels and then fill in the missing information using a refinement network. This is combined with an adversarial loss [16] and a perceptual feature matching loss [27], which is used to improve training stability [18], [26]. However, TVSN requires that the visibility maps be computed ahead of time when rendering the objects. This limits the technique to only work on synthetic data sets where it is possible to compute accurate 3D visibility maps ahead of time.

III. Method

Our goal is to generate the 3D textured point cloud of a single object given a single input view. Our training set consists of a set of multi-view observations for several instances of objects from the same category, together with their respective pose. We propose to replace the supervised point cloud estimation, by using a set of depth maps automatically predicted from novel views generated by a deep generative model. Rather than supervising for depth prediction [7], we leverage the advances in self-supervised/unsupervised depth estimation [14], [15]. These depths can then be un-projected to recover a partial

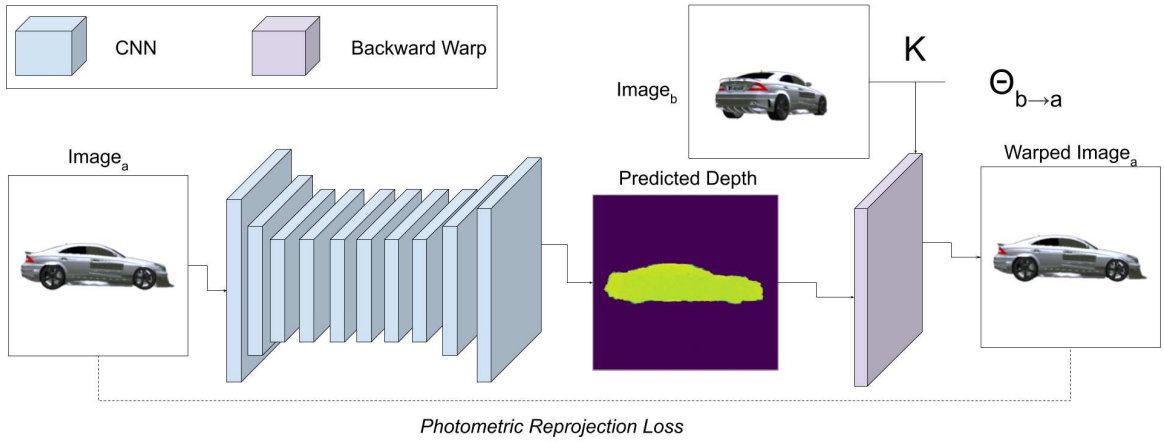

Fig. 2. The self-supervised depth network is trained using a set of images at different view points with known poses. The input image, $Image_a$ is passed through a CNN which predicts a dense depth map for the image. The predicted depth is then passed into a warping function, along with $Image_b$, the relative pose for the image pair $\theta_{b\rightarrow a}$ and the camera intrinsics matrix $\mathbf{K}$. The warping function uses a differentiable image sampler [22] to re-project the $Image_b$ in the pose of $Image_a$. Finally, the network is trained using a photometric consistency loss function (eq. 1), which allows the network to implicitly learn to predict depth, without any ground truth depth images.

point cloud for each generated novel view image. As the model predicts the novel fixed views, we can estimate the depth, un-project using the known camera intrinsics and then combine the N partial point clouds into a single 3D point cloud using the inverse of the object pose. In our experiments, we set $N = 5$ and the camera poses are set to fixed 80 degree intervals, such that they have overlapping fields of view. These output viewpoints are independent from the object pose in the input image.

A. Self-Supervised Depth Estimation

To estimate the 3D point cloud for a set of images, we train a self-supervised monocular depth estimator. First, a convolutional neural network is used predict the depth for a given input image $I_a : \Omega \rightarrow \mathbb{R}^3$, where Ω denotes image lattice. Then, using the vectorized homogeneous depth points[2] $\mathbf{Z}_a \in \mathbb{R}^{N\times 4}$ (4^{th} dimension represents the homogeneous coordinate), known camera intrinsics $\mathbf{K} \in \mathbb{R}^{4\times 4}$ and relative camera pose $\mathbf{T}_{b\rightarrow a} \in \mathbb{R}^{4\times 4}$, the next image in the set I_b is un-projected and transformed into a matrix of homogeneous points $\mathbf{P}_b \in \mathbb{R}^{N\times 4}$. The un-projected points are transformed back into the source target frame and then re-projected. This process is defined as follows:

$$\mathbf{P}_b = \mathbf{K}\mathbf{T}^{-1}(\mathbf{K}^{-1}\mathbf{Z}_a^T). \tag{1}$$

The point set $\mathbf{P}_b$ in (1) is then sampled using a differentiable image sampler [22], such that the pixels in the source image are warped into the original image

$$\hat{I}_a = \phi(I_b, \mathbf{P}_b), \tag{2}$$

where $\phi(.)$ denotes the differential sampler defined by [22].

The photometric re-projection error [14], [15] is then computed between the warped image and the original input image. This forces the model to implicitly learn to predict depth for the input image. At test time, only a single image is needed to predict the depth output. The photometric re-projection loss function is computed as follows:

$$\mathcal{L}_{pe} = \sum_{x,y} \|I_a(x,y) - \hat{I}_a(x,y)\|_1. \tag{3}$$

The photometric re-projection loss can be any image reconstruction loss function computed in pixel space. In our case, we find that using a mean absolute error (i.e., L1 distance) is sufficient and provides sharper depth estimates than L2 distance. An overview of this process can be found in Fig. 2

B. Novel View Synthesis

While it is possible to only use a novel view synthesis model with a simple regression loss, this often leads to blurry and inaccurate images (see Table II for a comparison). As our method requires chaining together image synthesis and a depth estimation, we aim to generate accurate novel views such that the point cloud can be as accurate as possible. Therefore, to improve the image quality we train our novel view model as a generative adversarial network [16] (GAN). The object of the GAN framework is to train two networks, the generator network $G(.)$ which attempts to generate samples that are real enough to fool the discriminator network $D(.)$. These networks are then trained in an alternating fashion[3]. Empirically, we find that the standard adversarial loss [16] is unstable and fails to give satisfactory results. Therefore, we opt to use the least squares generative adversarial loss (LSGAN) [28] formulation, which shows more stable results during training. The LSGAN loss functions for the discriminator network $D(.)$ is defined by:

$$\mathcal{L}_{dis}(G,D) = \frac{1}{2}\mathbb{E}[(D(I_y)-1)^2] + \frac{1}{2}\mathbb{E}[(D(G(I_x))^2], \tag{4}$$

[2]The x and y coordinates are uniformly sampled from a 2D grid between $[-1, 1]$ for each spatial location in I_a.

[3]This process can be thought of as a zero-sum game where the objective is to find a Nash Equilibrium between the two networks.

where I_x is the input image and I_y is the ground truth images for each of the novel viewpoints associated with I_x. The loss function for the generator network $G(.)$ is defined as:

$$\mathcal{L}_{gan}(G, D) = \mathbb{E}[(D(G(I_x)) - 1)^2]. \quad (5)$$

We wish to generate the highest possible resolution point cloud, we therefore need to synthesize high resolution novel views. As GANs often struggle with generating images with a resolution greater than 128x128, Iizuka *et al.* [29] and Wang *et al.* [18] suggest that using multiple discriminators at different image scales improves with both the local and global consistency of synthesized images at high resolution. Each discriminator is trained at a different scale improving training stability. Similarly to Wang *et al.* [18], we use three scales, represented by $k \in \{1, 2, 3\}$, and optimizing the generator adversarial loss (5) as the sum of the multiple scale discriminator outputs:

$$\min_G \max_{D1,D2,D3} \sum_{k=1,2,3} \mathcal{L}_{gan}(G, D_k), \quad (6)$$

where D_k represents the Discriminator network for each scale k. We also use an adversarial feature matching loss [18], [26] to improve training stability. The feature matching loss extracts multiple feature maps from the different scale intermediate layers of the Discriminator network. The $L1$ error is then computed between the feature representations for both the real images samples and the synthesized images. This feature matching loss is computed for each of the multiple scale discriminators D_k, as follows:

$$\mathcal{L}_{feat}(G, D_k) = \sum_{i=0}^{T} \frac{1}{N_i} \|D_k^{(i)}(I_y) - D_k^{(i)}(G(I_x))\|_1, \quad (7)$$

where T represents the number of intermediate layers. The error between the feature maps is then weighted by the size of each feature map N_i at each intermediate feature scale i. As we are training a conditional GAN we also use a reconstruction term to encourage the network to create exact reconstructions:

$$\mathcal{L}_{recon}(G) = \|I_y - G(I_x)\|_1. \quad (8)$$

where the reconstruction loss (8) is computed between the synthesised images $G(I_x)$ and the corresponding ground truth images I_y. The final loss function for training the refinement/novel-view network is then computed as the weighted sum of the previous equations:

$$\min_G \mathcal{L}(G, D) = \mathcal{L}_{recon} * \lambda_1 + \mathcal{L}_{feat} * \lambda_2 + \mathcal{L}_{gan} * \lambda_3. \quad (9)$$

As the $G(.)$ and $D(.)$ networks are trained in an alternating fashion, the final objective for training the multi-scale discriminator networks is to minimize the sum of the discriminative loss in (4) for each of the different scales k:

$$\min_{D1,D2,D3} \sum_{k=1,2,3} \mathcal{L}_{dis}(G, D_k). \quad (10)$$

C. Unprojection and Masking

Finally, it is possible to estimate the 3D point cloud for a set of novel images generated by the novel view network, by passing the novel views through the trained depth estimator. These depths can then be un-projected to form the final point cloud $P \in \mathbb{R}^{N\times 4}$ by performing (11) but stopping before re-projection. Given the vectorized depth points $\mathbf{Z}_n \in \mathbb{R}^{N\times 4}$ for each of the novel view points n, known camera intrinsics $\mathbf{K} \in \mathbb{R}^{4\times 4}$ and relative camera poses $\mathbf{T}_n \in \mathbb{R}^{4\times 4}$ we can un-project the point cloud for each viewpoint by:

$$P = \mathbf{T_n^{-1}}(\mathbf{K^{-1}}\mathbf{Z}_n^T) \quad \forall n, \quad (11)$$

where n denotes the index of the novel view. As the self-supervised depth estimation model is trained implicitly via the image warping function, the model will still attempt to estimate depth for undefined regions. If we were using real images it would be possible to remove background and outlying points based solely on the depth value. However, as the rendered images from ShapeNet data set have no background, we decide to predict the object mask along side the RGB channels in the novel view synthesis. When performing the un-projection, we can mask the background depth image values using this predicted mask.

IV. Experiments

We evaluate the efficacy of our system for single view 3D point cloud reconstruction using the *car* category of the ShapeNet data set [19]. The images in this data set are taken at uniformly sampled poses and have 256×256 pixels. We select the car class due to its large number of varied instances with high textural detail. We use an instance-wise split of 80%/20% for training and testing, exactly as defined in [6], [7], [9], [10], [13]. A UNet network [30] is used for both the depth prediction network and the novel view network. Both networks use convolutional encoder blocks consisting of a strided convolution, batch normalization [4] and leaky ReLU [4]. The convolutional decoder differs between the two architectures. The depth prediction network uses UpConv blocks (bilinear upsample + convolution), as we found that using a transposed convolution results in unacceptable artifacts. In the novel view network, we found that the transposed convolution layers were necessary to stabilize the GAN training [17]. All up-sampling blocks make use of batch normalization and leaky ReLU. We train our novel view network and discriminator using the Adam optimizer with learning rate 0.0002 and 0.0004 respectively. Furthermore, we set the hyper-parameters that control the momentum in the Adam optimizer to $\beta_1 = 0.0$ and $\beta_2 = 0.999$ for both $G(.)$ and $D(.)$. We set the loss function weights (Eq. 9) as $\lambda_1 = 100$, $\lambda_2 = 1$ and $\lambda_3 = 1$. To train the depth estimator, we use the Adam optimizer with learning rate 0.001 with default momentum. For each batch, the input/target $\mathbf{I}_a$ view and source view point $\mathbf{I}_b$ are randomly selected from the data set to be corresponding views that are rotated 20° from one another. Training hyperparameters were selected via manual search.

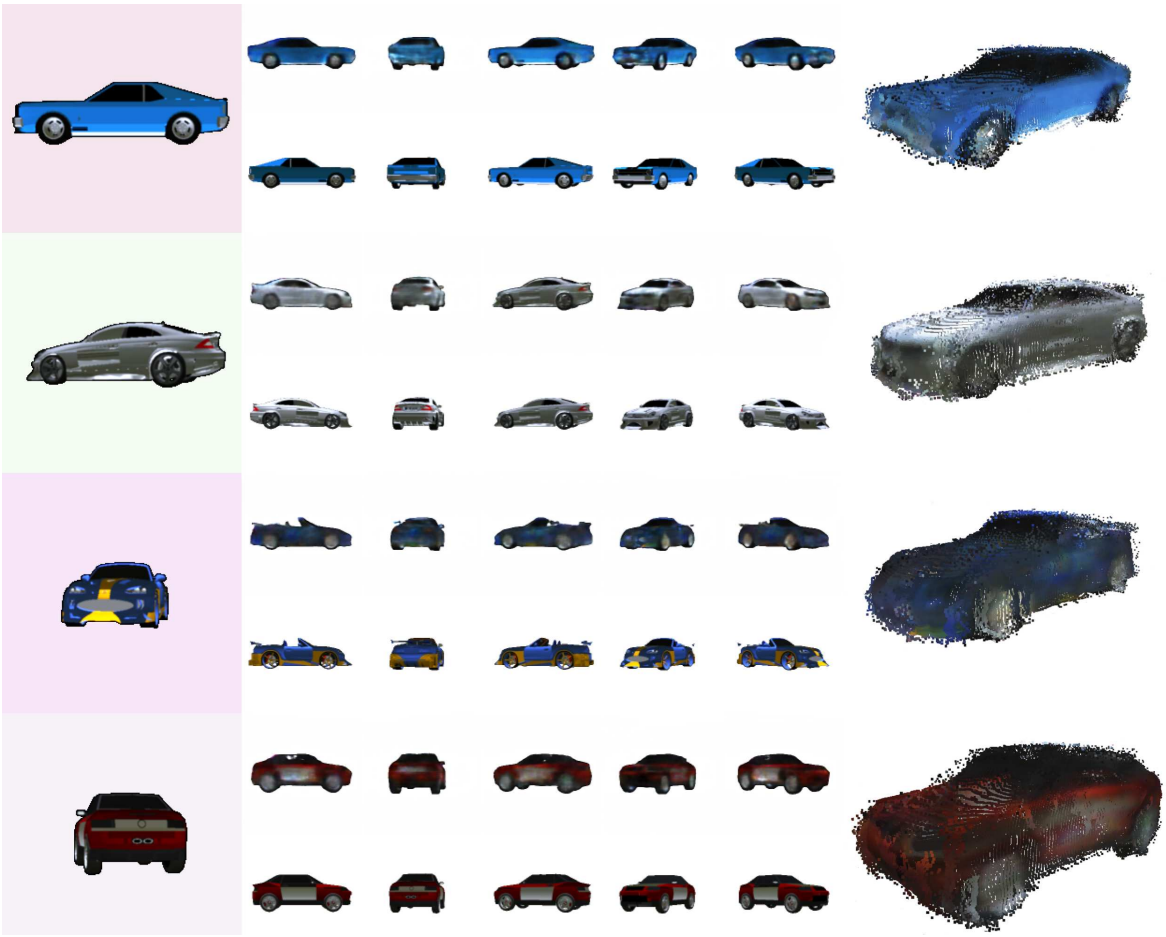

Fig. 3. **Qualitative results** for the cars category on the ShapeNet test set. Our method is capable of synthesizing coherent and accurate 3D point clouds, with textural (colour) information, using only a single image as input. *Left:* Input image, *Middle-Top:* predicted novel views, *Middle-Bottom:* ground truth test images, *Right:* 3D point clouds, un-projected using the depth and novel view networks.

A. Evaluation Metrics

1) Shape Metric: To quantitatively evaluate our 3D point clouds, we opt to use the Chamfer distance metric [7], [13] as it has been shown to be highly correlated with human judgment of 3D shape similarity. Given a ground truth point cloud $\mathbf{P}_{gt}$ and a predicted point cloud $\mathbf{P}_{pr}$, the distance is defined as follows:

$$d_{Chamf}\left(\mathbf{P}_{gt}, \mathbf{P}_{pr}\right) = \min \left\|\mathbf{P}_{pr} - \mathbf{P}_{gt}\right\|_2 + \min \left\|\mathbf{P}_{gt} - \mathbf{P}_{pr}\right\|_2 \tag{12}$$

The Chamfer distance is defined by a sum of two components. The left-hand component measures the precision, or how similar the predicted point cloud is to the ground truth. While the right-hand side is the coverage of the predicted point cloud, which measures how well the points cover the surface of the object.

2) Image Metric: To measure the image generation quality we use the structured similarity image metric (SSIM) [31] The SSIM metric is often used as a perceptual measure of the quality of an image and has been show to have a strong correlation with human perception of image quality [31]. The SSIM measure is computed between the two sets of image patches of size $W \times W$ extracted from the predicted image $\hat{x}$ and ground truth image x:

$$SSIM(\hat{x}, x) = \frac{(2\mu_{\hat{x}} 2\mu_x + c_1)(2\sigma_{\hat{x}x} + c_2)}{(\mu_{\hat{x}}^2 + \mu_x^2 + c_1)(\sigma_{\hat{x}}^2 + \sigma_x^2 + c_2)}, \tag{13}$$

where $\mu_{\hat{x}}$ and μ_x are the means for each window, and $\sigma_{\hat{x}}$ and σ_x are the variance for each window. While $\sigma_{\hat{x}x}$ is the covariance between the windows $\hat{x}$ and x, the constants are set to the default values of $c_1 = 0.01^2$ and $c_2 = 0.03^2$, and the window size is set to the default value of $W = 11$. The measure returns a value in the range $[0.0, 1.0]$ with 1.0 being perfect recreation of the original image.

B. Single-view Reconstruction

The quantitative results for the single view object reconstruction task are reported in Table I. When comparing the Chamfer distance (12) of our system with several supervised methods [7], [10], [13], we observe that we outperform all other reported methods. We likely outperform the simpler point cloud and volumetric methods [10], [13] due to the denser representation afforded by using a depth map representation. Note that as 3D-R2N2 [10] uses a 3D volumetric

TABLE I

QUANTITATIVE RESULTS OF OUR METHOD IN SINGLE VIEW 3D RECONSTRUCTION COMPARED AGAINST SEVERAL SUPERVISED (ABOVE LINE) AND SELF-SUPERVISED (BELOW LINE) SYSTEMS. NUMBERS REPORTED ARE POINT CLOUD PRECISION/COVERAGE/CHAMFER DISTANCE (12). THE BEST NUMBERS FOR EACH CATEGORY ARE IN BOLD FONT (LOWER IS BETTER).

Method	Car
3D-R2N2 (1 view) [10]	1.808 / 3.238 / 5.046
3D-R2N2 (3 view) [10]	1.685 / 3.151 / 4.836
3D-R2N2 (5 view) [10]	1.664 / 3.146 / 4.810
Fan *et al.* [13]	1.800 / 2.053 / 3.853
Lin *et al.* [7]	1.446 / **1.061** / 2.507
Insafutdinov and Dosovitskiy [9]	- / - / 2.42
Proposed (1 view)	**1.208** / 1.208 / **2.416**

TABLE II

ABLATION STUDY RESULTS OF THE SHAPENET CAR CATEGORY. IMAGE QUALITY RESULTS ARE EVALUATED USING THE STRUCTURED SIMILARITY METRIC IN (13) (HIGHER IS BETTER) AND THE POINT CLOUD CHAMFER DISTANCE (12) (LOWER IS BETTER). MS: MULTI-SCALE DISCRIMINATOR. FM: DISCRIMINATOR FEATURE MATCHING LOSS. THE BEST NUMBERS FOR EACH CATEGORY ARE IN BOLD FONT.

Method	SSIM	Chamfer
Deep Convolutional VAE	0.8410	2.623
Encoder-Decoder (No GAN)	0.8475	2.692
Encoder-Decoder (GAN)	0.8493	2.476
Encoder-Decoder (GAN + MS + FM)	0.8550	2.43
UNet (GAN + MS + FM)	**0.8756**	**2.416**

representation, the shapes are converted to a point cloud via uniform sampling along the boundary of the volume, severely limiting the final resolution of the point cloud representation. Furthermore, we outperform the method in [7], which also uses a depth map based representation, however, unlike us their method is supervised for depth and is unable to recover textural information. We argue that the improvement over the supervised depth estimator [7] is due to the use of the geometric loss function to train the depth network. While our model under-performs in terms of coverage metric in (12), when compared with Lin *et al.* [7], we believe this is due to the simplifying setup that we rely on, consisting of novel views images with zero-degree elevation. As the images have zero elevation, points that are partially self-occluded (e.g. on the bonnet or roof of the car) will be sparser than points in direct view. In future, this could be addressed by using multiple elevations in the novel view network. We also compare with the current state of the art method [9], which also uses self-supervised learning to estimate the 3D point cloud. Our method slightly outperforms with respect to Chamfer distance, but the exact numbers for the method in [9] regarding precision and coverage are unavailable for a more detailed analysis. It is clear from the qualitative results shown in Fig. 3, that our method fails to preserve high frequency information like racing stripes or decals, even when utilizing a GAN. However, the general shape and colour are consistent, with some fine details being recovered.

C. Ablation Study

As our novel view network has a complex training process, we also performed an ablation study to show the efficacy of the GAN method and the proposed architecture. The results presented in Table II use the same training setup as used in Section IV-B. We evaluate the use of a simple Encoder-Decoder model, which contains no skip connections, but otherwise is architecturally the same as the UNet. We also show results without the GAN loss function, trained only with $L1$ loss for the Encoder-Decoder model. Furthermore, we also evaluate the use of the Multi-Scale and Discriminator Feature Matching losses in (8) for both the Encoder-Decoder model and the UNet model. Finally, we also tested our novel view model using a Variational Autoencoder (VAE) [4], another type of deep generative model. We also present the results for the UNet without the multi-scale and feature matching discriminators [18], [26]. The comparison is based on the SSIM result over the test set for each of the different methods – see Table II. It is clear from the results that each of the architectural and extra losses, such as multi-scale and feature matching discriminators, are required to achieve a state of the art result with our method. Counter-intuitively, we found that the skip connections provide a significant improvement in SSIM, Chamfer distance and overall 3D reconstruction quality. Normally, skip connections are used to pass high level structural details for observable details in the input image e.g. object edges and boundaries. Therefore, there should be limited improvement by adding in skip connections, as there will be limited overlap between observable features and predicted novel views. We believe there are two reasons for the improvement when using skip connections. The first is that the encoder-decoder cannot easily recover the finer details of the texture as there is limited capacity in the hidden layer for representing the 256×256-pixel images. Furthermore, it is challenging for the encoder-decoder to correctly estimate the overall colour of an object resulting in blurry and patchy texturing, as can be seen in Fig. 4. Secondly, we empirically found that the UNet model is more stable when training the GAN. We believe this is because when one of the target views and the input view are very similar, the network has an easier task of predicting that novel view as it can simply "copy" many of the pixels to the output. The result of this is that the discriminator cannot overpower the generator network as easily. Furthermore, as objects like cars have many textural symmetries, the skip connections can provide important cues to the model about the shape and symmetries of the objects we are trying to reconstruct.

V. CONCLUSIONS AND FUTURE WORK

In this work, we have presented a method for reconstructing textured 3D point clouds from single images. We achieved this result by leveraging the advances in both deep generative modelling and self-supervised depth estimation. We have shown state of the art results for 3D point cloud reconstruction for the car category in the ShapeNet data set [19]. Future work will focus on extending this method to work

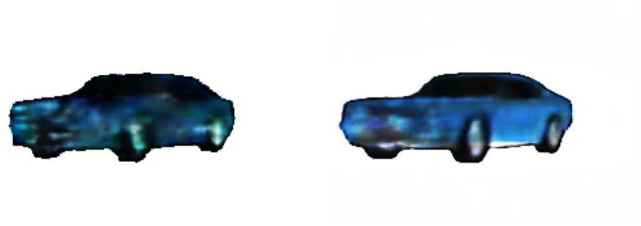

Fig. 4. Comparison of our method without using UNet skip connections (left) and when using skip connections (right).

for multiple categories, by making use of new improvements in deep generative modelling [32]. To allow for training on images with limited textural detail (Chairs and Airplanes), our method could be further improved by unifying the depth and novel view networks with the incorporation of a differentiable projection function, similar to that presented in [9]. The depth network could be further improved by using both multi-scale [24] and structural dissimilarity loss functions [23] and the Novel View model could be extended to include both geometric reasoning and refinement [26]. Additionally, our method assumes known ground truth poses for performing the reconstruction. By incorporating a 3D pose estimator [23], [24], it would be possible to remove this limitation and train a fully unsupervised 3D reconstruction model.

VI. Acknowledgment

This research was in part supported by the Data to Decisions Cooperative Research Centre (A.J). Supported by Australian Research Council through grants DP180103232, CE140100016. G.C. acknowledges the support by the Alexander von Humboldt-Stiftung for the renewed research stay sponsorship.

References

[1] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[2] B. K. Horn and M. J. Brooks, *Shape from shading*. MIT press, 1989.

[3] K. N. Kutulakos and S. M. Seitz, "A theory of shape by space carving," *International Journal of Computer Vision*, vol. 38, no. 3, pp. 199–218, 2000.

[4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[5] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1912–1920.

[6] A. Johnston, R. Garg, G. Carneiro, I. Reid, and A. van den Hengel, "Scaling cnns for high resolution volumetric reconstruction from a single image," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 939–948.

[7] C.-H. Lin, C. Kong, and S. Lucey, "Learning efficient point cloud generation for dense 3d object reconstruction," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[8] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee, "Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 1696–1704. [Online]. Available: http://papers.nips.cc/paper/6206-perspective-transformer-nets-learning-single-view-3d-object-reconstruction-without-3d-supervision.pdf

[9] E. Insafutdinov and A. Dosovitskiy, "Unsupervised learning of shape and pose with differentiable point clouds," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[10] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[11] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik, "Multi-view supervision for single-view reconstruction via differentiable ray consistency," in *Computer Vision and Pattern Regognition (CVPR)*, 2017.

[12] G. Riegler, A. O. Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," *CoRR*, vol. abs/1611.05009, 2016. [Online]. Available: http://arxiv.org/abs/1611.05009

[13] H. Fan, H. Su, and L. J. Guibas, "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*, 2017, pp. 605–613.

[14] R. Garg, V. K. BG, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *European Conference on Computer Vision*. Springer, 2016, pp. 740–756.

[15] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 270–279.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[17] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[18] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," *arXiv preprint arXiv:1711.11585*, 2017.

[19] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," Stanford University — Princeton University — Toyota Technological Institute at Chicago, Tech. Rep. arXiv:1512.03012 [cs.GR], 2015.

[20] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," *arXiv preprint arXiv:1904.08755*, 2019.

[21] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[22] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025.

[23] C. Godard, O. Mac Aodha, M. Firman, and G. Brostow, "Digging into self-supervised monocular depth estimation," *arXiv preprint arXiv:1806.01260*, 2018.

[24] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "Sfm-net: Learning of structure and motion from video," *arXiv preprint arXiv:1704.07804*, 2017.

[25] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View synthesis by appearance flow," in *European conference on computer vision*. Springer, 2016, pp. 286–301.

[26] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg, "Transformation-grounded image generation network for novel 3d view synthesis," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 702–711.

[27] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*. Springer, 2016, pp. 694–711.

[28] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 2813–2821.

[29] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 107, 2017.

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[31] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
[32] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410.