Inferring Point Clouds from Single Monocular Images by Depth Intermediation

Wei Zeng\textsuperscript{1}, Sezer Karaoglu\textsuperscript{1,2}, Theo Gevers\textsuperscript{1,2}

\textsuperscript{1}Computer Vision Lab, University of Amsterdam
\textsuperscript{2}3DUniversum
{\{w.zeng, s.karaoglu, th.gevers\}@uva.nl}

Abstract

In this paper, we propose a framework for generating 3D point cloud of an object from a single-view RGB image. Most previous work predict the 3D point coordinates from single RGB images directly. We decompose this problem into depth estimation from single images and point completion from partial point clouds.

Our method sequentially predicts the depth maps and then infers the complete 3D object point clouds based on the predicted partial point clouds. We explicitly impose the camera model geometrical constraint in our pipeline and enforce the alignment of the generated point clouds and estimated depth maps.

Experimental results for the single image 3D object reconstruction task show that the proposed method outperforms state-of-the-art methods. Both the qualitative and quantitative results demonstrate the generality and suitability of our method.

1. Introduction

Inferring the 3D shape from a single image is a fundamental task in computer vision with different applications in robotics, CAD systems, virtual and augmented reality. Recently, increasing attention has focused on deep 3D shape generation from single RGB images [4, 6] with the creation of large-scale datasets [3], and the development of deep learning techniques.

A number of the previous methods represent the estimated 3D shape as a voxelized 3D occupancy grid [4, 9, 31]. Albeit easy to integrate into deep neural networks, these voxel-based methods may suffer from data sparsity because most of the information needed to compute the 3D structure is given by the surface voxels. In fact, the part of the shape representation lies on the surface of the 3D object, which makes up only a small fraction of all voxels in the occupancy grid. This makes the use of 3D CNNs computational expensive yielding considerable amount of overhead during training and inference. To overcome these issues, recent methods focus more on designing neural network architectures and loss functions to process and predict point clouds (3D). These point clouds consist of points which are uniformly sampled over the object surfaces. For example, Fan et al. [6] introduce a framework and loss functions designed to generate unordered point clouds directly from 2D images. Other work extends this pipeline by adding geometrically driven loss functions for training [11]. However, the inference procedure does not explicitly impose any geometrical constraint.

In this paper, we address this problem and propose an efficient framework to sequentially predict the depth map to infer the full 3D object shape, see Figure 1. The transformation of the depth map into the partial point cloud is based on the camera model. In this way, the camera model is ex-
Figure 2. Overview of our framework. Our proposed network receives a RGB image as input. It predicts the depth map of the input image, and calculates the partial point cloud based on camera geometry. Then, a point cloud auto-encoder is applied to the predicted partial point cloud to abstract the representative code vector. The full point cloud is generated from the code vector in a sparse-to-dense fashion. Finally, the 3D-2D refinement module ensures the alignment between the generated full 3D point cloud and the estimated depth map.

Our method is composed of three components, namely, depth intermediation, point cloud completion and 3D-2D refinement, see Figure 2 for a detailed overview of our framework.

First, given a single RGB image of an object, the depth intermediation module predicts the depth map, and then computes the point cloud of the visible part of the object contained in the RGB image. We refer to this single-view point cloud as the partial point cloud. The computation of the partial point cloud is based on the camera model geometry. In this way, we explicitly impose the camera model as a geometrical constraint in our transformation to regulate the 2D-3D domain transfer.

The point cloud completion module infers the full point cloud using the partial point cloud as input. An encoder-decoder network is used to convert the partial point cloud to the full point cloud [33]. The encoder is an auto-encoder that takes the predicted partial point cloud as input and learns to reproduce it. We use the low-dimensional representation, i.e. code vector, as the representative feature vector of the point cloud. The decoder takes this feature vector to produce the full point cloud.

Finally, the 3D-2D refinement process enforces the alignment between the generated full point cloud and the depth map prediction. The refinement module imposes a 2D projection criterion on the generated point cloud together with the 3D supervision on the depth estimation. This self-supervised mechanism enables our network to jointly optimize both the depth intermediation and the point cloud completion modules.

In summary, our contributions in this work are as follows:

- A novel neural network pipeline to generate 3D shapes from single monocular RGB images by depth intermediation.
- Incorporating the camera model as a geometrical constraint to regulate the 2D-3D domain transfer.
- A 3D-2D refinement module to jointly optimize both depth estimation and point cloud generation.
- Outperforming the state-of-the-art methods on the task of 3D single view reconstruction on the ShapeNet dataset.

2. Related Work

Depth Estimation Single-view, or monocular, depth estimation refers to the problem where only a single image is available at test time. Eigen et al. [5] show that it is possible to produce pixel depth estimations using a two scale deep network trained on images and their corresponding depth values. Several methods extend this approach by introducing new components such as CRFs to increase the accuracy [15], changing the loss from regression to classification [2], using other more robust loss functions [13], and by incorporating strong scene priors [28]. Recently, there are a number of methods to estimate depth in an unsupervised way. Garg et al. [7] introduce an unsupervised method by using an image alignment loss. Godard et al. [8] propose an unsupervised deep learning framework by employing loss functions which impose consistency between predicted depth maps which are obtained from different camera viewpoints. Kuznietsov et al. [12] adopt a semi-supervised deep method to predict depths from single images. As opposed to existing methods, in our work, we use supervised depth estimation to produce depth maps to enable the inference of 3D shapes. Our 3D-2D refinement module uses the generated full point cloud as a 3D supervision algorithm to steer the depth estimation.

Point Cloud Feature Learning Point cloud feature extraction is a challenging problem because points of 3D point clouds lie in a non-regular space and cannot be processed
easily by standard CNNs. Qi et al. [22] propose PointNet to extract unordered point representations by using multi-layer perceptrons and global pooling. As a follow-up work, PointNet++ abstracts local patterns by sampling representative points and recursively applying PointNet as learning blocks to obtain the final representation. Wei et al. [34] introduce 3DContextNet that exploits both local and global contextual cues imposed by the k-d tree to learn point cloud features hierarchically. Yang et al. [32] propose a folding-based decoder that deforms a canonical 2D grid onto the underlying 3D object surface of a point cloud. In our work, we leverage the PointNet layers and folding operations to build our point completion module. The PointNet layer is used as the basic learning block to build our network. The folding operation is used as the last step of our point completion module to transform the sparse full point cloud to a dense full point cloud.

3D Shape Completion Shape completion is an essential task in geometry and shape processing and has wide applications. The aim of existing methods is to complete shapes using local surface primitives, or to formulate it as an optimization problem [21, 24]. Recently, there is a growing number of methods that exploit shape structures and regularities [20, 26], and methods using strong database priors [1, 16]. These methods, however, often require that the datasets contain the exact parts of the shape, and thus are limited in generalization power. With the advances of large-scale shape repositories like ShapeNet [3], researchers start to develop fully data-driven methods. For example, 3D ShapeNets [30] use a deep belief network to obtain a generative model for a given shape database. Nguyen et al. [25] extend this method for mesh repairing. Most existing learning-based methods represent shapes by voxels. However, volumetric representation are suited for convolutional neural networks. In contrast, our method uses point clouds. Point clouds preserve the full geometric information about the shapes while being memory efficient. Related to our work is PCN [33], which uses an encoder-decoder network to generate full point clouds in a coarse-to-fine fashion. However, our method is not limited to the shape completion task. The aim is to generate the full point cloud of an object from a single RGB image.

Single-image 3D Reconstruction Traditional 3D reconstruction methods [14, 17, 19] require correspondences of multiple views. Recently, increasing attention has focused on data-driven 3D reconstruction from single images [4, 6, 31].

One class of the previous researches focuses on voxel-based representations [4, 9, 31, 29]. Choy et al. [4] propose 3D-R2N2 that takes as an input one or more images of an object taken from different viewpoints. The output is reconstruction of the object in the form of a 3D occupancy grid by means of recurrent neural networks. Follow-up work on this [9] proposes an adversarial constraint to regularize the predictions by a large amount of unlabeled realistic 3D shapes. Tulsiani et al. [27] adopt an unsupervised solution for 3D object reconstruction and jointly learn shape and pose predictions by enforcing consistency between the predictions and available observations. Jiajun et al. [29] also attempt to reconstruct the 3D shapes from 2.5D sketches. They first recover the 2.5D sketches of objects and then treat the predicted 2.5D sketches as intermediate images to regress the 3D shapes. Different from their method, our proposed approach explicitly imposes the camera model into the transformation and infers the partial point clouds from predicted depth maps purely based on 3D geometry.

Voxel-based representations are computationally expensive and are only suitable for coarse 3D voxel resolutions. To address this issue, Fan et al. [6] introduce point cloud based representations for 3D reconstruction. They propose an end-to-end framework to directly regress the point location from a single image. Different from [6], our approach sequentially predicts the depth map, infers the partial point cloud based on the camera model, and generates the full point cloud of the 3D shape. In addition, we also explicitly enforce the alignment between the generated point cloud and the estimated depth map to jointly optimize both of the components.
3. Method

We propose a method that generates point clouds from \( RGB \) images using depth intermediation. To recover a 3D point cloud from a single view \( RGB \) image, our network uses three modules: (1) a depth intermediation module is used to predict depth maps and calculate the partial point clouds based on the camera model geometry; (2) a point cloud completion module is proposed to infer full 3D point clouds from predicted partial point clouds; (3) a 3D-2D refinement mechanism is proposed to enforce the alignment between the generated point clouds and the estimated depth maps. Our whole pipeline can be trained in an end-to-end fashion and enables to jointly optimize both depth estimation and point cloud generation.

3.1. Depth Intermediation

The first component of our network takes a 2D \( RGB \) image of an object as input. It predicts the depth map of the object and calculates the visible point cloud based on the camera model. The aim of the depth intermediation module is to regulate the 2D-3D domain transfer and to constrain the structure of the learned manifold. Most of the previous methods directly generate the 3D shape from a single 2D image. Although they use geometrically inspired loss functions during training, the inference procedure does not explicitly impose any geometrical constraint. In contrast, our method uses the predicted depth map to compute the partial point cloud. In this way, during inference, geometrical constraints are explicitly incorporated by means of depth estimation and the camera model.

An encoder-decoder network architecture is used for our depth estimation. The encoder is a VGG-16 architecture up to layer conv5_3 encoding a \( 224 \times 224 \) \( RGB \) image into 512 feature maps of size \( 7 \times 7 \). The decoder contains five sets of \( 3 \times 3 \) deconvolutional layers, followed by four \( 1 \times 1 \) convolutional layers. Skip connections link the related layers between the encoder and decoder. The output is the corresponding depth map with the same resolution as the 2D \( RGB \) input image.

Then, the partial point cloud is computed using a camera model. For a perspective camera model, the correspondence between a 3D point \((X, Y, Z)\) and its projected pixel location \((u, v)\) on an image plane is given by:

\[
Z[u, v, 1]^T = K(R[X, Y, Z]^T + t)
\]

where \( K \) is the camera intrinsic matrix, \( R \) and \( t \) denote the rotation matrix and the translation vector. In our paper, we assume that the principal points coincide with the image center, and that the focal lengths are known. Note that when the exact focal length is not available, an estimation (approximation) may still suffice. When the object is reasonably distant from the camera, larger focal lengths will choose between perspective and weak-perspective models.

In general, object-level depth estimation is coarse. Hence, the corresponding partial point cloud may suffer from noise (e.g. flying dots) on the boundaries along the frustum. The aim of our 3D-2D refinement is to enforce the partial point cloud to be consistent with the full point cloud. The aim is to reduce the estimation errors at the boundaries. For example, consider Figure 3, where depth maps and their corresponding partial point clouds are shown. The predicted partial point cloud without refinement (second row) suffers from errors (i.e. the flying dots). This type of estimation errors are largely reduced by our 3D-2D refinement process (third row).

3.2. Point Cloud Completion

The full point cloud is inferred by learning a mapping from the space of partial observations to the space of complete shapes. The point cloud completion module consists of two parts: an extraction and a generation stage. The aim of the extraction stage is to concisely represent the geometric information of the partial point cloud by a code vector \( v \). A point cloud auto-encoder is proposed to compute the (lower-dimensional) code vector. The encoder part is based on graph max-pooling [23] and DenseNet [10]. Specifically, the encoder is composed of PointNet layers and graph-based max-pooling layers. The graph is the K-nearest neighbor graph constructed by considering each point in the input point set as a vertex, with edges connecting only to nearby points. The max-pooling operation is only applied to the local neighborhood of each point to aggregate (local) data signatures. In the experiments, we choose \( K = 16 \). We use one PointNet layer followed by one graph-based max-pooling layer as one graph layer. We also connect the output of each graph layer to every other graph layer in a feed-forward fashion. The reason is to regulate the flow of information and gradients throughout the network. The output of the encoder is passed to a feature-wise global max-pooling component to produce a \( k \)-dimensional vector. This vector is the basis of our latent space. The decoder transforms the latent vector using \( 3 \) fully connected layers to produce the same input. We use a \( k \)-dimensional representation \( k = 1024 \) in our paper, i.e. code vector, as the input for the generation of the full point cloud.

In the generation stage, the network architecture is similar to the decoder of PCN [33]. The code vector is taken as input. It produces a sparse output point cloud by a fully-connected decoder [6]. Then, a detailed output point cloud is obtained by a folding-based decoder [32]. The fully-connected decoder predicts a sparse set of points representing the global geometry of an object. The folding-based decoder approximates a smooth surface representing the local geometry of a shape. In this paper, \( n = 256 \) sparse point clouds are generated by the fully-connected decoder and used as key point sets. Then, for each key point \( p_i \), a
patch of \( t = u^2 \) points (\( u = 2 \) in our experiments) is generated in local coordinates which are centered at \( p_i \) via the folding-based decoder. Eventually, \( N = 1024 \) complete point cloud are generated as output of the network.

### 3.3. 3D-2D Refinement

In this section, the aim is to align the predicted point clouds and the corresponding estimated depth maps and to jointly optimize both the depth intermediation and the point completion module.

For the depth intermediation network, flying dots may occur in the inferred partial point cloud near the object boundaries along the frustum, as shown in Figure 3. The cause of this is the lack of contextual information for object-level depth estimation. Therefore, the aim of the 3D-2D refinement is to reduce these estimation errors (i.e. depth noise reduction).

To reduce the depth estimation errors, the generated point cloud is used as a 3D self-supervision component. A point-wise 3D Euclidean distance is used between the partial point cloud and the full point cloud, which is defined as:

\[
L_d(P_d, P_p) = \sum_{p_i \in P_d} \min_{p_j \in P_p} \| p_i - p_j \|^2_2 \tag{2}
\]

where \( P_d \) and \( P_p \) are the predicted partial point cloud and the predicted full point cloud, respectively. This regularizes the partial point cloud to be consistent with the full point cloud with the aim to reduce noise (i.e. flying points).

To constrain the generated point cloud using the 2D projection supervision, we penalize points in the projected image \( I_p \) which are outside the silhouette \( I_s \):

\[
L_p = \sum_{q_i \in Q_p} I((I_p(q_i) - I_s(q_i)) > 0) \min_{q_j \in Q_s} \| q_i - q_j \|^2_2 \tag{3}
\]

where \( Q_p \) and \( Q_s \) represent the pixel coordinates of the projected image and the silhouette, respectively. \( I(\cdot) \) is an indicator function set to 1 when a projected point is outside the silhouette. The purpose of this constraint is to recover the details of the 3D shape.

### 4. Experiments

**Dataset:** We train and evaluate the proposed networks using the ShapeNet dataset [3], which contains a large collection of categorized 3D CAD models. For fair comparison, we use the same training/testing split as in Choy et al. [4].

**Training Details:** Our networks are optimized using the Adam optimizer. During our experiments, we found that it is crucial to initialize the network properly. Therefore, we follow a two-stage training procedure: the depth estimation network and the point completion network are first pretrained to predict the depth map and the complete point cloud, separately. The depth estimation network is trained with the L2 loss. The point completion network is trained using the ground truth of the partial point clouds as input with the Chamfer/Earth Mover’s distance loss. Then, the entire network is jointly trained end-to-end with the 3D-2D refinement as a complementary constraint.

**Evaluation Metric:** We evaluate the generated point clouds of the different methods using three metrics: point cloud based Chamfer Distance (CD), point cloud based Earth Mover’s Distance (EMD) and voxel based Intersection over Union (IoU).

The Chamfer Distance loss measures the distance between the predicted point cloud \( P_p \) and the ground truth point cloud \( P_{gt} \). This loss is defined by:

\[
L_{CD}(P_p, P_{gt}) = \sum_{x \in P_p} \min_{y \in P_{gt}} \| x - y \|^2_2 + \sum_{y \in P_{gt}} \min_{x \in P_p} \| x - y \|^2_2 \tag{4}
\]

The Earth Mover’s Distance requires \( P_p, P_{gt} \in \mathbb{R}^3 \) to have equal size \( s = |P_p| = |P_{gt}| \). The EMD distance is defined as:

\[
L_{EMD}(P_p, P_{gt}) = \min_{\phi: P_p \rightarrow P_{gt}} \sum_{x \in P_p} \| x - \phi(x) \|^2_2 \tag{5}
\]

where \( \phi: P_p \rightarrow P_{gt} \) is a bijection. A lower CD/EMD value represents better reconstruction results.

To compute IoU of the predicted and ground truth point clouds, we follow the setting of GAL [11]. Each point set is voxelized by distributing points into \( 32 \times 32 \times 32 \) grids. The point grid for each point is defined as a \( 1 \times 1 \times 1 \) grid centered at this point. For each voxel, the maximum intersecting volume ratio of each point grid and this voxel is calculated as the occupancy probability. IoU is defined as follows:

\[
IoU = \frac{\sum_i I[V_{gt}(i) > 0] I[V_p(i) > 0]}{\sum_i I[V_{gt}(i) + V_p(i) > 0]} \tag{6}
\]

where \( V_{gt} \) and \( V_p \) are the voxelized ground-truth and prediction, respectively. \( i \) is the index of the voxels. \( I(\cdot) \) is an indicator function. A higher IoU value indicates more precise point cloud prediction.

Table 1 shows the quantitative comparison between 3D-R2N2 [4], PSGN [6], 3D-LMNet [18] and our proposed method. 3D-R2N2 takes as an input one or more images of an object taken from different viewpoints. It outputs a reconstruction of the object in the form of a 3D occupancy grid by using recurrent neural networks. PSGN utilizes fully-connect layers and deconvolutional layers to predict 3D points directly from 2D images. 3D-LMNet is a latent-embedding matching method to learn the prior over 3D point clouds. Our method outperforms existing methods for most of the categories and achieves a lower overall...
Table 1. CD and EMD metric on ShapeNet. Our proposed method outperforms the state-of-the-art on most categories and achieves a lower overall mean error in both Chamfer and Earth Mover’s Distance metrics. All numbers are scaled by a factor of 100.

Table 2. IoU of the 3D reconstruction results. It can be derived that our proposed method achieves higher IoU for most of the categories.

Table 3. Depth estimation results before and after 3D-2D refinement. The depth estimation network strongly benefits from the 3D self-supervision approach in the 3D-2D refinement module. All numbers are scaled by a factor of 10.

mean score. Note that 3D-LMNet applies the iterative closest point algorithm (ICP) as a post-processing step. However, our proposed method still outperforms 3D-LMNet for 9 out of 13 using the Chamfer metric.

A number of qualitative results are presented in Figure 4. In the first row, both PSGN and our method perform well to generate the full point cloud. In the second to fifth row, our method provides accurate structures, while PSGN confuses some details of the 3D shapes (backrest of the bench in the second row, the open back of the Pick-up in the third row, and the table legs in the fourth row). Our method also generates better pose estimation. This is shown for viewpoint v2 in the fourth row. Further, the result of the proposed method is more aligned with the ground truth than PSGN. A failure case is shown in the last row. Both methods, PSGN and ours, are not able to capture the correct structure of the chair leg.

Table 2 shows the IoU value for each category. It is shown that our method achieves higher IoU for most of the categories. 3R-R2N2 is able to predict 3D shapes from more than one views. For many of the categories, our method even outperforms the 3D-R2N2s prediction using
Figure 4. Qualitative results for the ShapeNet dataset. We demonstrate the reconstruction results from two representative viewpoints $v_1$ and $v_2$. Compared to PSGN, the proposed method is better in capturing the overall shape and in generating finer details.

5 views.

Table 3 shows that the depth estimation network benefits from the 3D self-supervision strategy of the generated point cloud. As shown in Figure 3, the depth estimation with only 2D supervision may suffer from the estimation error near the boundaries along the frustum. With our 3D-2D refinement, the generated full point cloud is utilized as 3D self-supervision to reduce the estimation error.

**Results on Real-World Images:** We also test the generalizability of our approach for real-world images. We use the model trained from the ShapeNet dataset directly and run it on real images without fine-tuning. Results are shown in Figure 5. It can be (visually) derived that our model trained on synthetic data generalizes well to the real-world images.

5. Conclusion

In this paper, we propose an efficient framework to generate 3D point clouds from single RGB images by sequentially predicting the depth maps and inferring the complete 3D object shapes. Depth estimation and camera model are explicitly incorporated into our pipeline as geometrical constraints during both training and inference. We also enforce the alignment between the predicted full 3D point clouds and the corresponding estimated depth maps to jointly optimize both depth intermediation and the point completion module.

Both qualitative and quantitative results on ShapeNet show that our method outperforms existing methods. Furthermore, it also generates precise point clouds from the
Figure 5. Qualitative results on real-world images. Our proposed method generalizes well to the real-world images.

real-world images. In the future, we plan to extend our framework to scene-level point cloud generation from images.

References

[1] A. Brock, T. Lim, J. M. Ritchie, and N. Weston. Generative and discriminative voxel modeling with convolutional neural networks. arXiv preprint arXiv:1608.04236, 2016.

[2] Y. Cao, Z. Wu, and C. Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. IEEE Transactions on Circuits and Systems for Video Technology, 2017.

[3] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012, 2015.

[4] 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 European conference on computer vision, pages 628–644. Springer, 2016.

[5] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In Advances in neural information processing systems, pages 2366–2374, 2014.

[6] H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3d object reconstruction from a single image. In CVPR, volume 2, page 6, 2017.

[7] 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, pages 740–756. Springer, 2016.

[8] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In CVPR, volume 2, page 7, 2017.

[9] J. Gwak, C. B. Choy, M. Chandraker, A. Garg, and S. Savarese. Weakly supervised 3d reconstruction with adversarial constraint. In 3D Vision (3DV), 2017 International Conference on, pages 263–272. IEEE, 2017.

[10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In CVPR, volume 1, page 3, 2017.

[11] L. Jiang, S. Shi, X. Qi, and J. Jia. Gal: Geometric adversarial loss for single-view 3d-object reconstruction. In European Conference on Computer Vision, pages 820–834. Springer, Cham, 2018.

[12] Y. Kuznietsov, J. Stückler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6647–6655, 2017.

[13] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In 3D Vision (3DV), 2016 Fourth International Conference on, pages 239–248. IEEE, 2016.

[14] A. Laurentini. The visual hull concept for silhouette-based image understanding. IEEE Transactions on pattern analysis and machine intelligence, 16(2):150–162, 1994.

[15] B. Li, C. Shen, Y. Dai, A. Van Den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1119–1127, 2015.

[16] Y. Li, A. Dai, L. Guibas, and M. Nießner. Database-assisted object retrieval for real-time 3d reconstruction. In Computer Graphics Forum, volume 34, pages 435–446. Wiley Online Library, 2015.

[17] S. Liu and D. B. Cooper. Ray markov random fields for image-based 3d modeling: Model and efficient inference. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 1530–1537. IEEE, 2010.

[18] P. Mandikal, N. Murthy, M. Agarwal, and R. V. Babu. 3d-lmnet: Latent embedding matching for accurate and diverse 3d point cloud reconstruction from a single image. arXiv preprint arXiv:1807.07796, 2018.

[19] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan. Image-based visual hulls. In Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pages 369–374. ACM Press/Addison-Wesley Publishing Co., 2000.

[20] N. J. Mitra, L. J. Guibas, and M. Pauly. Partial and approximate symmetry detection for 3d geometry. ACM Transactions on Graphics (TOG), 25(3):560–568, 2006.
[21] A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. Laplacian mesh optimization. In Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia, pages 381–389. ACM, 2006.

[22] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. arXiv preprint arXiv:1612.00593, 2016.

[23] Y. Shen, C. Feng, Y. Yang, and D. Tian. Mining point cloud local structures by kernel correlation and graph pooling. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, volume 4, 2018.

[24] O. Sorkine and D. Cohen-Or. Least-squares meshes. In Shape Modeling Applications, 2004. Proceedings, pages 191–199. IEEE, 2004.

[25] D. Thanh Nguyen, B.-S. Hua, K. Tran, Q.-H. Pham, and S.-K. Yeung. A field model for repairing 3d shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5676–5684, 2016.

[26] S. Thrun and B. Wegbreit. Shape from symmetry. In ICCV, pages 1824–1831. IEEE, 2005.

[27] S. Tulsiani, A. A. Efros, and J. Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. Computer Vision and Pattern Recognition (CVPR), 2018.

[28] X. Wang, D. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 539–547, 2015.

[29] J. Wu, Y. Wang, T. Xue, X. Sun, B. Freeman, and J. Tenenbaum. Marrnet: 3d shape reconstruction via 2.5d sketches. In Advances in neural information processing systems, pages 540–550, 2017.

[30] 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, pages 1912–1920, 2015.

[31] B. Yang, H. Wen, S. Wang, R. Clark, A. Markham, and N. Trigoni. 3d object reconstruction from a single depth view with adversarial learning. arXiv preprint arXiv:1708.07969, 2017.

[32] Y. Yang, C. Feng, Y. Shen, and D. Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), volume 3, 2018.

[33] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert. Pcn: Point completion network. In 2018 International Conference on 3D Vision (3DV), pages 728–737. IEEE, 2018.

[34] W. Zeng and T. Gevers. 3dcontextnet: Kd tree guided hierarchical learning of point clouds using local contextual cues. arXiv preprint arXiv:1711.11379, 2017.