RELATIVE VIEWPOINT ESTIMATION BASED ON STRUCTURED 3D REPRESENTATION ALIGNMENT

Kohei Matsuzaki and Kei Kawamura

KDDI Research, Inc., Japan

ABSTRACT

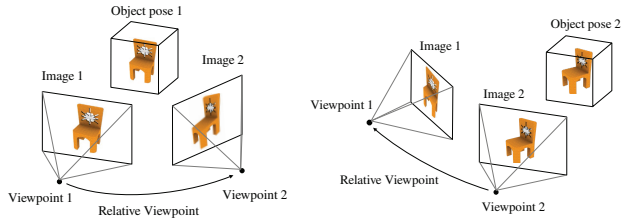
Relative viewpoint estimation is a fundamental problem in various image processing applications. Traditional estimation approaches can fail if sufficient appearance overlap is not observed between two images. Recent advances in 3D representation learning from images have made it possible to exploit the underlying 3D structure. In this paper, we propose a relative viewpoint estimation method using an end-to-end trainable network that learns structured 3D representations. In the proposed method, an independent coordinate system is set for each image in order to construct a structured 3D representation. This makes it possible to estimate the relative viewpoint by aligning those representations through coordinate transformations. Experimental results on the ShapeNet, Pix3D, and Thingi10K datasets demonstrated that the proposed method achieves accurate estimation even if there is not sufficient observable appearance overlap between the images.

***Index Terms*—** Viewpoint estimation, appearance overlap, structured 3D representation, end-to-end training

1. INTRODUCTION

Relative viewpoint estimation is a process to estimate the transformation between viewpoints for an object in given images. It is a fundamental problem and plays a crucial role in various image processing applications such as 3D reconstruction, visual localization, and augmented reality. It remains a daunting problem for machines in spite of much recent progress in this research area.

Traditional approaches estimate the relative viewpoint based on a set of keypoint correspondences obtained by local feature description [1] and random sample consensus (RANSAC) [2] in the context of multiple view geometry [3]. With the advance of deep learning techniques, some approaches have been proposed to reformulate local feature description and the RANSAC algorithm using deep neural networks [4, 5, 6]. However, they can fail when there is not sufficient observable overlap in appearance between two images. A major limitation of keypoint correspondence-based approaches is that they rely on appearance overlap between the images. To overcome this limitation, the approaches should be able to recognize the invisible parts.



(a) Coordinate system 1 (b) Coordinate system 2

Fig. 1: Illustration of independent coordinate systems in the proposed method. In coordinate system 1, viewpoint 1 is set up in front of the object and viewpoint 2 corresponds to a relative viewpoint with respect to viewpoint 1. In coordinate system 2, viewpoint 2 is set up in front of the object.

Several recent studies [7, 8, 9] have succeeded in learning structured 3D representations from images to exploit the underlying 3D structure. These learn 3D representations that can control the object poses by mapping image features to a physically meaningful coordinate system using viewpoints. Subsequent work [10] proposed an approach to estimate a relative viewpoint using a structured 3D representation. This approach trains two networks. The first network learns the structured 3D representation from image pairs. The second network is a discriminator, which predicts the magnitude of relative viewpoint error from the structured 3D representation. Therefore, it is a multi-stage approach and involves multiple networks that are not end-to-end trainable.

In this paper, we propose a relative viewpoint estimation method with an end-to-end trainable network. This method constructs structured 3D representations from an image pair and estimates the relative viewpoint by their alignment. As shown in Fig. 1, we assume different poses of the same object in independent coordinate systems for each image. This allows the structured 3D representations to be aligned directly by coordinate transformation. To compare relative viewpoints, we calculate the ray termination probabilities using a differentiable ray consistency [11]. In inference, we find a relative viewpoint based on the similarity of the probabilities. This achieves an estimation with reduced dependence on appearance overlap between images. In our experiments using the ShapeNet [12], Pix3D [13], and Thingi10K [14] datasets, we demonstrate the effectiveness of the proposed method.

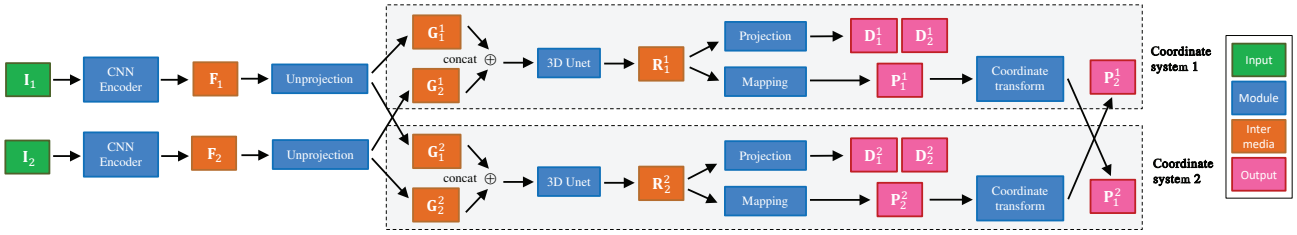


Fig. 2: Overview of our network architecture. The network extracts 2D features $\mathbf{F}_i$ from input images $\mathbf{I}_i$ $(i = 1, 2)$. These features are unprojected to the 3D features $\mathbf{G}_i^j$ $(j = 1, 2)$ in different coordinate systems. These 3D features are refined to $\mathbf{R}_i^j$ through 3D UNet. The network outputs predicted depth maps $\mathbf{D}_i^j$ using different viewpoints from the refined 3D features. The network also outputs the predicted and transformed ray termination probabilities $\mathbf{P}_i^j$. While we draw the same name blocks at the top and bottom parts to distinguish the processing to the two inputs, they represent common modules.

In summary, our main contributions are as follows:

- We propose a method to accurately estimate the relative viewpoint, even if there is little appearance overlap, through aligning the structured 3D representations.
- We introduce a framework to learn structured 3D representations on independent coordinate systems using an end-to-end trainable network.
- We experimentally evaluate the proposed method, showing that it outperforms state-of-the-art methods in the task of relative viewpoint estimation.

2. METHOD

We aim to obtain the relative viewpoint from a pair of images using an end-to-end trainable network. To this end, we propose a method to estimate the relative viewpoint by constructing structured 3D representations in the independent coordinate systems and aligning them. This allows our network to learn more effective representations for viewpoint estimation than previous work that conducts training separately.

Notation. We use subscript numbers such as $\mathbf{I}_1$ to distinguish a pair of input images and their corresponding products. We add superscript numbers such as $\mathbf{G}_1^1$ to represent the index of coordinate systems that are independent for each 3D representation. In addition, since our method predicts the depth map by projection in each coordinate system, superscript numbers are also added to the corresponding depth maps such as $\mathbf{D}_1^1$.

2.1. Network Architecture

An overview of our network architecture is shown in Fig. 2. We train this network that receives an input tuple $(\mathbf{I}_1, \mathbf{I}_2, \theta)$, where $\mathbf{I}_1$ and $\mathbf{I}_2$ are an image pair and θ is the relative viewpoint corresponding to them. The output of this network is predictions of depth maps and ray termination probabilities. Given the image pair $\mathbf{I}_1$ and $\mathbf{I}_2$, feature tensors $\mathbf{F}_1 \in \mathbb{R}^{M\times N\times N}$, $\mathbf{F}_2 \in \mathbb{R}^{M\times N\times N}$ are extracted from each image through a 2D CNN encoder, where M and N represent the number of channels and spatial resolution, respectively.

These tensors are unprojected by the differentiable unprojection layer [15], resulting in structured 3D representations. This operation transforms 2D image feature into a 3D feature grid using virtual viewpoints. We unproject each 2D image feature into two 3D grid coordinate systems. These are the coordinate systems in which each image feature is projected from the front of the 3D grid. The two 3D feature grids $\mathbf{G}_1^1 \in \mathbb{R}^{M\times N\times N\times N}$, $\mathbf{G}_2^2 \in \mathbb{R}^{M\times N\times N\times N}$ are obtained by projecting each image feature onto the 3D grid in front of it. Furthermore, the two 3D feature grids $\mathbf{G}_1^2 \in \mathbb{R}^{M\times N\times N\times N}$, $\mathbf{G}_2^1 \in \mathbb{R}^{M\times N\times N\times N}$ are obtained by projecting one image feature onto the coordinate system corresponding to another image using the input relative viewpoint θ. Here, the features extracted from $\mathbf{I}_1$ are projected onto the coordinate system corresponding to $\mathbf{I}_2$. In the opposite case, the inverse of the input relative viewpoint θ is calculated.

We then fuse and refine the 3D feature grid located in the same coordinate system $\{\mathbf{G}_1^1, \mathbf{G}_2^1\}$, $\{\mathbf{G}_1^2, \mathbf{G}_2^2\}$. We concatenate each pair of 3D feature grids and refine them through 3D UNet [16, 17], which learns to inpaint the input features. As a result, the refined tensors $\mathbf{R}_1^1 \in \mathbb{R}^{M\times N\times N\times N}$ and $\mathbf{R}_2^2 \in \mathbb{R}^{M\times N\times N\times N}$ are obtained. After that, the depth maps are predicted by projecting the tensors $\mathbf{R}_1^1$ and $\mathbf{R}_2^2$ onto the 2D plane through the differentiable projection layer [15]. Here, the viewpoints from the front of each tensor and its relative viewpoints are used. The depth maps $\mathbf{D}_1^1$, $\mathbf{D}_2^1$ are predicted from $\mathbf{R}_1^1$ and the depth maps $\mathbf{D}_1^2$, $\mathbf{D}_2^2$ are predicted from $\mathbf{R}_2^2$. Furthermore, the ray termination probabilities $\mathbf{P}_1^1 \in \mathbb{R}^{N\times N\times N}$ and $\mathbf{P}_2^2 \in \mathbb{R}^{N\times N\times N}$ are predicted through a linear layer that maps the per-grid code vectors of tensors $\mathbf{R}_1^1$ and $\mathbf{R}_2^2$ to the ray termination probability.

To align the coordinate systems of two ray termination probabilities $\mathbf{P}_1^1$ and $\mathbf{P}_2^2$, we apply rigid-body transformations to them. In this transformation, the grid values are interpolated according to the trilinear interpolation method. We transform the $\mathbf{P}_1^1$ exploiting the input relative viewpoint θ to obtain $\mathbf{P}_1^2$. This $\mathbf{P}_1^2$ takes a coordinate system aligned with $\mathbf{P}_2^2$. Similarly, we obtain $\mathbf{P}_2^1$ from $\mathbf{P}_2^2$ by exploiting the inverse of the relative viewpoint θ.

2.2. Training and Inference

Our network is trained by minimizing loss functions in an end-to-end fashion. The loss functions consist of three parts, namely, depth map loss $\mathcal{L}_{\text{depth}}$, ray consistency loss $\mathcal{L}_{\text{ray}}$, and transformation loss $\mathcal{L}_{\text{trans}}$. The total loss $\mathcal{L}$ is a weighted sum of these three, and is defined as follows:

$$\mathcal{L} = \mathcal{L}_{\text{depth}} + \lambda_1 \mathcal{L}_{\text{ray}} + \lambda_2 \mathcal{L}_{\text{trans}}. \quad (1)$$

In the following, we briefly explain each loss term.

Depth Loss. We measure a sum of L_1 losses between the ground-truth depth maps $\mathbf{D}_i$ and the predicted depth maps $\hat{\mathbf{D}}_i^j$ as the depth loss:

$$\mathcal{L}_{\text{depth}} = \sum_{i,j} |\mathbf{D}_i - \hat{\mathbf{D}}_i^j|, \quad (2)$$

where i and j represent the index of the corresponding input image and coordinate system, respectively. This makes it possible to learn 3D representations from multiple 2D images based on a multi-view supervision approach.

Ray Consistency Loss. In order to train the network to predict ray termination probabilities, we measure the following ray consistency loss [11]:

$$\mathcal{L}_{\text{ray}} = \sum_{i,j,k} q_i^j(k)\psi_i^j(k), \quad (3)$$

where i and j represent the index of the corresponding input image and coordinate system, respectively. $q_i^j(k)$ and $\psi_i^j(k)$ indicate the event probability and event cost for k-th grid cell in the path of the camera ray. For the prediction, we introduce a linear layer that maps 3D representations to ray termination probabilities such as $\mathbf{P}_1^1$. Then we calculate the corresponding $q_i^j(k)$ and $\psi_i^j(k)$ using these ray termination probabilities.

Transformation Loss. We measure a sum of L_2 losses between ray termination probabilities on the same coordinate system as the transformation loss. This encourages maximization of the similarity between the probabilities in the same coordinate system through the coordinate transformations. We convert tensors $\{\mathbf{P}_1^i, \mathbf{P}_2^i\}$ to vector format $\{\mathbf{p}_1^i, \mathbf{p}_2^i\}$ by performing a flattening operation, where i represents the index of the corresponding coordinate system. We sum L_2 losses between the corresponding vectors as follows:

$$\mathcal{L}_{\text{trans}} = \sum_i \|\mathbf{p}_1^i - \mathbf{p}_2^i\|. \quad (4)$$

At inference time, we estimate a relative viewpoint of the input images $\mathbf{I}_1$ and $\mathbf{I}_2$ based on structured 3D representation alignment. We assume scene crops as in [18] as the input images. Unlike during training, our network uses relative viewpoint candidates to predict ray termination probabilities. We generate these candidates by uniformly sampling Euler angles. We search for a candidate that maximizes the cosine similarity between converted ray termination probabilities $\{\mathbf{p}_1^2, \mathbf{p}_2^2\}$. As a result, we obtain a relative viewpoint that best aligns the structured 3D representations.

3. EXPERIMENTS

3.1. Experimental Setups

Datasets. We use the ShapeNet [12], Pix3D [13], and Thingi10K [14] datasets for evaluation. We render an image from the 3D model using a viewpoint randomly chosen from azimuth in the range from -180 to 180 degrees and elevation in the range from -90 to 90 degrees. We sample 20 viewpoints for ShapeNet and Thingi10K, and 200 viewpoints for Pix3D to mitigate the difference in the number of models. In addition, we randomly select and blend an image from the SUN dataset [19] as the background as done in [20]. Then we construct image pairs from the images. In many cases, sufficient appearance overlap is not observed. We split the data into 80%/10%/10% for training, validating, and testing.

Evaluation Metrics. To evaluate the relative viewpoint estimation performance, we measure the geodesic distance [21] between the ground-truth and the estimated rotation as an angle error. We use $\text{Acc}_{\frac{\pi}{6}}$ and MedErr [22] as evaluation metrics. $\text{Acc}_{\frac{\pi}{6}}$ indicates a fraction of instances with error less than 30 degrees, and MedErr indicates a median error.

Implementation Details. We implemented our network using the PyTorch library. To train our model, we used the Rectified Adam optimizer [23] with a learning rate of 10^{-3}, weight decay of 10^{-6}, and momentum of 0.9. We set $\lambda_1 = 10^{-2}$ and $\lambda_2 = 10^{-3}$ in Eq. (1). We train each model with a Nvidia Quadro GV100 GPU and select the model that achieves the best performance on the validation dataset. The training is conducted for each dataset, regardless of object category. We set the resolution of the input image and depth map to 256×256. In all experiments, the number of feature channels and the spatial resolution of the feature grid are set to $M = 32$ and $N = 16$, respectively. At the inference, we uniformly sampled each Euler angle at 20-degree intervals to generate relative viewpoint candidates.

3.2. Comparison with State-of-the-art

We assume two state-of-the-art baselines in regard to relative viewpoint estimation. One is an RPNet [24] that directly regresses the relative viewpoint from an image pair. Its architecture consists of a Siamese network that uses GoogLeNet as the backbone encoder and a regressor as the inference module. The RPNet is a state-of-the-art viewpoint estimation method with an end-to-end trainable network. The other is a novel object viewpoint estimation (NOVE) [10], which searches for a relative viewpoint that minimizes the magnitude of predicted error. In this method, the first network constructs a structured 3D representation from an image pair, and the second network then predicts the magnitude of error from the representation. The first and second networks are trained separately. Since the full source code of NOVE is not publicly available, we re-implement it ourselves.

Table 1: Relative viewpoint estimation results.

	Acc $\frac{\pi}{6}$			MedErr		
	RPNet	NOVE	Ours	RPNet	NOVE	Ours
ShapeNet	54.74	50.18	**56.83**	25.94	29.38	**24.56**
Pix3D	62.38	59.83	**63.73**	21.92	23.02	**21.41**
Thingi10K	18.27	42.91	**44.28**	75.25	40.85	**38.54**
ShapeNet/Pix3D	58.23	61.18	**63.25**	23.81	22.35	**21.52**
ShapeNet/Thingi10K	21.38	37.71	**41.39**	71.37	51.01	**43.35**
Pix3D/Thingi10K	12.84	31.82	**33.34**	88.25	65.27	**61.71**

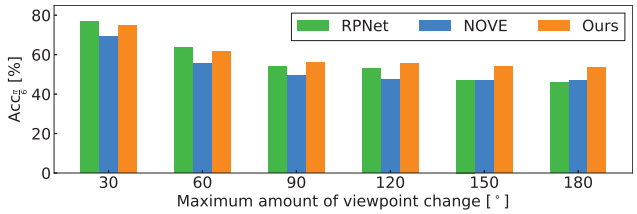


Fig. 3: Distributions of accuracy.

We summarize the relative viewpoint estimation results in Table 1. The first three rows show the results when training and testing are performed on subsets split from the same dataset. We can see that the proposed method outperforms the state-of-the-art baselines under both metrics, Acc $\frac{\pi}{6}$ and MedErr. It indicates the effectiveness of the proposed alignment of the structured 3D representations with regard to the relative viewpoint estimation. The performance of RPNet becomes low for the Thingi10K dataset, which has high variability in appearance. Following [10], we also evaluate the generalization ability of the model across the datasets. The last three rows show the results when training with one dataset and testing with another dataset, where training/testing is performed using the dataset of each row heading. In this case, it can be seen that the proposed method achieves comparable performance when using subsets of the same dataset. Therefore, we can expect that our model is generalized to unseen datasets.

We investigate the distribution of estimation accuracy with respect to the amount of viewpoint change between input images. We classify the estimation results according to the amount of viewpoint change using six ranges divided into 30-degree segments from 0 to 180 degrees. Then we measure the Acc $\frac{\pi}{6}$ independently for each range, as shown in Fig. 3. Although the Acc $\frac{\pi}{6}$ tends to decrease as the viewpoint changes become larger, we can observe that the proposed method maintains higher accuracy than the baseline methods. In other words, the proposed method achieves the most robust estimation against to the amount of viewpoint change.

3.3. Internal Analysis

We demonstrate the depth prediction results in the proposed method in Fig. 4. Each column represents a pair of input images, ground-truth depth maps, and predicted depth maps. It

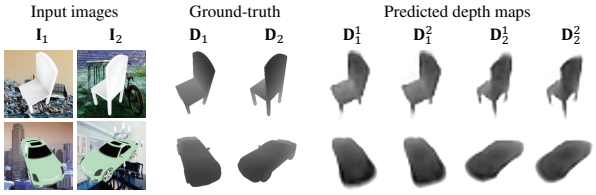


Fig. 4: Depth prediction results.

Table 2: Ablation study.

$\mathcal{L}_{\text{ray}}$	$\mathcal{L}_{\text{trans}}$	Acc $\frac{\pi}{6}$	MedErr
✓	✓	56.83	24.56
✓	✗	51.47	28.43
✗	✓	21.23	84.57
✗	✗	18.71	88.54

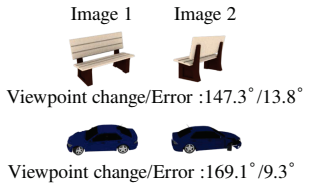


Fig. 5: Appearance examples.

is shown that these predicted depth maps capture the shape of the object accurately and that the network learns to separate the background naturally. Also, the depth maps predicted from different coordinate systems show similar results, indicating that our network learns similar 3D representations in independent coordinate systems.

We provide an ablation study to validate our model design with regard to the relative viewpoint estimation performance. Table 2 shows how the loss functions $\mathcal{L}_{\text{ray}}$ and $\mathcal{L}_{\text{trans}}$ affect the metrics Acc $\frac{\pi}{6}$ and MedErr on the ShapeNet dataset. The results show that introducing these functions provides better performance on the metrics. This suggests that both of these functions improve the relative viewpoint estimation performance since they facilitate learning more effective 3D representations for the proposed alignment.

We also visually confirmed that the proposed method can achieve accurate estimation even under large viewpoint changes, such as when estimation is performed from almost the opposite direction. Fig. 5 shows appearance examples when the estimation error is less than 30 degrees. For clarity, image backgrounds have been removed. From the results, it is found that the proposed method can achieve accurate estimation even if there is not sufficient observable overlap in appearance between the images.

4. CONCLUSIONS

In this paper, we proposed a relative viewpoint estimation method that can perform accurate estimation even if there is little appearance overlap between two images. This method estimates the relative viewpoint from the images by aligning structured 3D representations, which are constructed using an end-to-end trainable network. We conducted experiments to evaluate the effectiveness of the proposed method on the ShapeNet, Pix3D, and Thingi10K datasets. The results demonstrated that the performance of the proposed method is superior to state-of-the-art methods.

5. REFERENCES

[1] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[2] Martin A. Fischler and Robert C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[3] Alex M. Andrew, "Multiple view geometry in computer vision," *Kybernetes*, 2001.

[4] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua, "LIFT: Learned invariant feature transform," in *Proc. of ECCV*. Springer, 2016, pp. 467–483.

[5] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother, "DSAC-differentiable RANSAC for camera localization," in *Proc. of CVPR*, 2017, pp. 6684–6692.

[6] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua, "Learning to find good correspondences," in *Proc. of CVPR*, 2018, pp. 2666–2674.

[7] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer, "DeepVoxels: Learning persistent 3D feature embeddings," in *Proc. of CVPR*, 2019, pp. 2437–2446.

[8] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang, "HoloGAN: Unsupervised learning of 3D representations from natural images," in *Proc. of ICCV*, 2019, pp. 7588–7597.

[9] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein, "Scene representation networks: Continuous 3D-structure-aware neural scene representations," in *Proc. of NeurIPS*, 2019, pp. 1121–1132.

[10] Mohamed El Banani, Jason J. Corso, and David F. Fouhey, "Novel object viewpoint estimation through reconstruction alignment," in *Proc. of CVPR*, 2020, pp. 3113–3122.

[11] Shubham Tulsiani, Tinghui Zhou, Alexei A. Efros, and Jitendra Malik, "Multi-view supervision for single-view reconstruction via differentiable ray consistency," in *Proc. of CVPR*, 2017, pp. 2626–2634.

[12] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al., "ShapeNet: An information-rich 3D model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[13] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B. Tenenbaum, and William T Freeman, "Pix3D: Dataset and methods for single-image 3D shape modeling," in *Proc. of CVPR*, 2018, pp. 2974–2983.

[14] Qingnan Zhou and Alec Jacobson, "Thingi10K: A dataset of 10,000 3D-printing models," *arXiv preprint arXiv:1605.04797*, 2016.

[15] Abhishek Kar, Christian Häne, and Jitendra Malik, "Learning a multi-view stereo machine," in *Proc. of NIPS*, 2017, pp. 365–376.

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. of MICCAI*, 2015, pp. 234–241.

[17] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger, "3D U-Net: Learning dense volumetric segmentation from sparse annotation," in *Proc. of MICCAI*, 2016, pp. 424–432.

[18] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel, "Implicit 3D orientation learning for 6D object detection from rgb images," in *Proc. of ECCV*. Springer, 2018, pp. 699–715.

[19] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *Proc. of CVPR*, 2010, pp. 3485–3492.

[20] Hao Su, Charles R. Qi, Yangyan Li, and Leonidas J. Guibas, "Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views," in *Proc. of ICCV*, 2015, pp. 2686–2694.

[21] Du Q. Huynh, "Metrics for 3D rotations: Comparison and analysis," *Journal of Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 155–164, 2009.

[22] Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik, "Multi-view consistency as supervisory signal for learning shape and pose prediction," in *Proc. of CVPR*, 2018, pp. 2897–2905.

[23] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han, "On the variance of the adaptive learning rate and beyond," *arXiv preprint arXiv:1908.03265*, 2019.

[24] Sovann En, Alexis Lechervy, and Frédéric Jurie, "RPNet: An end-to-end network for relative camera pose estimation," in *Proc. of ECCVW*. Springer, 2018.