

# RaSim: A Range-aware High-fidelity RGB-D Data Simulation Pipeline for Real-world Applications

Xingyu Liu\*, Chenyangguang Zhang\*, Gu Wang, Ruida Zhang, and Xiangyang Ji\*\*

**Abstract**—In robotic vision, a de-facto paradigm is to learn in simulated environments and then transfer to real-world applications, which poses an essential challenge in bridging the sim-to-real domain gap. While mainstream works tackle this problem in the RGB domain, we focus on depth data synthesis and develop a Range-aware RGB-D data Simulation pipeline (*RaSim*). In particular, high-fidelity depth data is generated by imitating the imaging principle of real-world sensors. A range-aware rendering strategy is further introduced to enrich data diversity. Extensive experiments show that models trained with RaSim can be directly applied to real-world scenarios without any finetuning and excel at downstream RGB-D perception tasks. Data and code are available at <https://github.com/shanice-l/RaSim>.

## I. INTRODUCTION

With the advent of deep learning, neural networks have emerged as dominance for numerous 3D vision tasks, including 3D semantic segmentation [1], [2], [3], [4], object pose estimation [5], [6], [7], and depth completion [8], [9], [10]. However, Convolutional Neural Networks (CNNs) and Transformers are extremely data-driven, requiring vast amounts of high-fidelity RGB-D data during the training process. Moreover, obtaining large-scale 3D datasets and annotating their precise labels are extremely time-consuming and labor-intensive.

As a result, numerous approaches have been proposed to address the lack of real RGB-D data and labels. One of the most effective strategies is to simulate large-scale synthetic training data using tools like Blender [11] or OpenGL. Oftentimes, domain randomization is also employed to ensure the diversity of the data [12], [13]. However, rendered images still exhibit the drawbacks of low quality and lack of physical plausibility. Therefore, recent works have shifted their focus towards employing physically-based rendering techniques [14], [15], [16] to enhance image quality. While substantial efforts have been invested in enhancing the fidelity of synthetic RGB data, the sim-to-real domain gap *w.r.t.* the depth modality is still obvious. This is because synthetic depth data is typically flawless, whereas real-world depth data is incomplete, along with blur and artifacts.

To alleviate this problem, we introduce a Range-aware RGB-D data Simulation pipeline named **RaSim** to produce

Xingyu Liu, Chenyangguang Zhang, Ruida Zhang, and Xiangyang Ji are with the Department of Automation, Tsinghua University, Beijing, 100084, China, and also with BNRist, Beijing, 100084, China. E-mail: {liuxy21, zcyg22, zhangrd23}@mails.tsinghua.edu.cn, xyji@tsinghua.edu.cn.

Gu Wang is with the Lab for High Technology, Tsinghua University. E-mail: guwang12@gmail.com.

\*: Xingyu Liu and Chenyangguang Zhang have equally contributed.

\*\*:. Corresponding author.

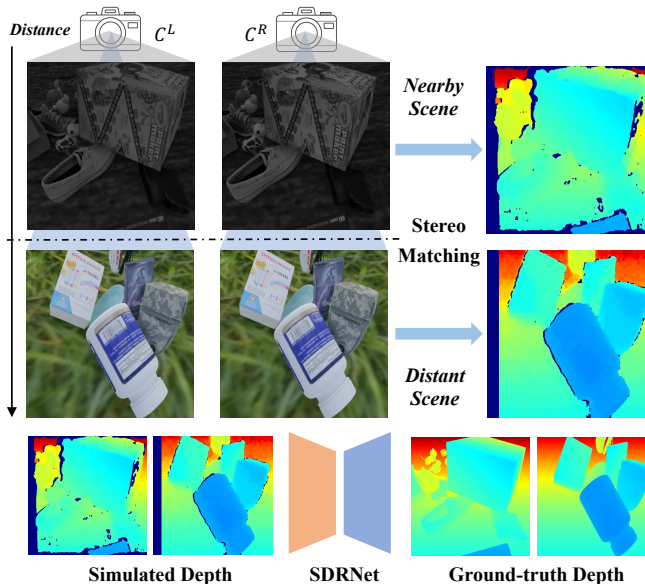


Fig. 1: Illustration of the core idea. We first generate high-fidelity simulated depth maps by imitating the imaging principle of the stereo camera, and further design a range-aware rendering strategy that renders binocular IR or RGB images according to distance to enrich data diversity. Then an SDRNet is devised to restore the ground-truth depth from simulated depth.

high-fidelity simulated 3D data. As shown in Fig. 1, our simulation system is grounded on imitating the imaging principle of the stereo camera based on the RealSense D400 series, as they have broad applications in both industrial and academic scenarios. Implemented with Kubric [14], we first generate large corpora of virtual scenes with photo-realistic object models, diverse backgrounds, global illuminations, and physical simulations. Then simulated depth maps are obtained by performing the semi-global stereo-matching algorithm using binocular images. We further devise a range-aware rendering strategy to enrich data diversity. Specifically, the type of matching images for depth simulation varies between IR and RGB depending on the distance between the scene and the camera. This strategy allows us to simulate nearby and distant scenes, enabling the pipeline to adapt to a wider range of application scenarios.

Supported by RaSim which randomizes over lighting and textures, we create a large-scale domain-randomized dataset that includes simulated and ground-truth depth maps, pixel-level semantic annotations, and millions of instances

labeled with poses, categories, and 3D object coordinates. To verify whether RaSim can assist in real-world applications, we train networks with the proposed RaSim dataset for two RGB-D-based perception tasks: depth completion and depth pre-training. Firstly, a **Simulated Depth Restoration Network (SDRNet)** is trained to repair the incomplete and noisy simulated depth map by decoding hierarchical RGB and depth features extracted with Swin Transformer [17]. Subsequently, inspired by the idea of *masked language modeling* in natural language processing [18], we consider depth restoration as a pre-training task for RGB-D-based Transformer. Specifically, weights pre-trained on RaSim are used to initialize the depth branch of the Transformer for facilitating various downstream tasks. Note that we opt for Transformer over CNNs, in that the scarcity of data in the Transformer architecture is a more prominent concern.

To verify the effectiveness of RaSim, we conduct extensive experiments on two real-world datasets, *i.e.*, ClearGrasp [8] for depth completion and YCB-V [5] for depth pre-training. To sum up, our contributions are threefold:

- By imitating the imaging principle of the stereo camera, we propose a RaSim pipeline to produce high-fidelity simulated depth and photo-realistic RGB-D images. A range-aware scene rendering strategy is further introduced to enrich the diversity of depth data.
- Supported by the RaSim pipeline, we create a large-scale synthetic RGB-D dataset that comprises more than 206K images across 9,835 diverse scenes. This dataset is equipped with physical simulations, comprehensive annotations, and the integration of domain randomization techniques.
- We conduct extensive experiments on two RGB-D-based perception tasks, *i.e.*, depth completion and depth pre-training, to demonstrate the applicability of RaSim in real-world scenarios.

## II. RELATED WORK

This work relates to two major strands of research: synthetic RGB-D dataset generation and learning from simulated environments.

### A. Synthetic RGB-D Dataset Generation

High-quality synthetic data generation plays a crucial role in 3D vision tasks since it is error-prone and labor-intensive to collect, calibrate, and annotate realistic RGB-D data. There are various synthetic 3D dataset generation pipelines like BlenderProc [11], Omnidata [19], OpenRooms [20], and Kubric [14]. However, the depth maps directly generated from these pipelines are too idealistic to adjust to real-world scenarios, since the depth collected from the real world could be noisy and incomplete.

More recently, Dai *et al.* [21] proposed a pipeline called DREDS to generate simulated depth by imitating the RealSense D415 camera following [22]. However, DREDS faces limitations in terms of physics simulation, which results in semantic ambiguity. Additionally, the diversity of depth data is restrained by the ideal range (0.5 – 2 meters) of their

system. Moreover, since DREDS is tailored for category-level pose estimation, the variety *w.r.t.* object categories are relatively scarce. In contrast, our work focuses on generating a large-scale, photo-realistic RGB-D synthetic dataset featuring rich annotations, physical simulations, a diverse array of objects and scenes, and an extensive depth range.

### B. Learning from Simulated Environments

In robotic vision, a widely adopted strategy involves training the network in simulated environments and subsequently transferring to real-world applications, such as robotic grasping [23], pose estimation [24], [25], [26], depth completion [8], [21], and scene understanding [27], [28], [29]. Driven by this strategy, sim-to-real approaches like domain randomization and domain adaptation play a pivotal role in the learning process. Specifically, domain randomization diversifies training data to adapt to various testing scenes [30], [31], [32], while domain adaptation leverages transfer learning techniques to align the simulated environment with the real world [33], [34], [35]. In this work, we address the sim-to-real challenge from both perspectives. On one hand, we introduce randomization in object categories, indoor and outdoor scenes, illuminations, and camera poses. On the other hand, we simulate high-fidelity depth maps by imitating real-world sensors to adapt to real domains.

## III. RANGE-AWARE RGB-D DATA SIMULATION

### A. Overview

Synthetic depth generated by the traditional pipelines is accurate, complete, and noise-free, while the depth collected from the real world is of low quality, along with blur and artifacts. To bridge the sim-to-real domain gap, we choose to simulate active stereo depth sensors, *i.e.*, Intel RealSense D400 Series, as they are relatively cheap and have broad applications in both industrial and academic scenarios. RealSense D400 imaging system includes an infrared (IR) projector, stereo IR cameras with a baseline distance  $C_b$  and a unified focal length  $C_f$ , as well as a central RGB camera. After the projector emits infrared light, the stereo cameras fetch left and right IR images respectively. Given binocular images, we could calculate the disparity value  $\mathcal{D}_p$  with the semi-global matching algorithm [36].

Finally, the depth  $z_{sim}$  is obtained as follows

$$z_{sim} = \frac{C_b \cdot C_f}{\mathcal{D}_p + \epsilon}, \quad (1)$$

where we set  $\epsilon$  to  $10^{-6}$  to avoid dividing by zero.

### B. Range-aware Scene Rendering

Following [21], [22], we render stereo IR images by having all ambient lights emit rays in the IR spectrum with reduced intensity. Additionally, a weak light value is added to simulate radiance from the environment. Finally, the rendered IR images are generated in grayscale. Despite the above pipeline generating high-fidelity depth from stereo IR images, a significant flaw is that the depth quality declines sharply when the camera is far from the scene ( $\geq 2m$ ). It

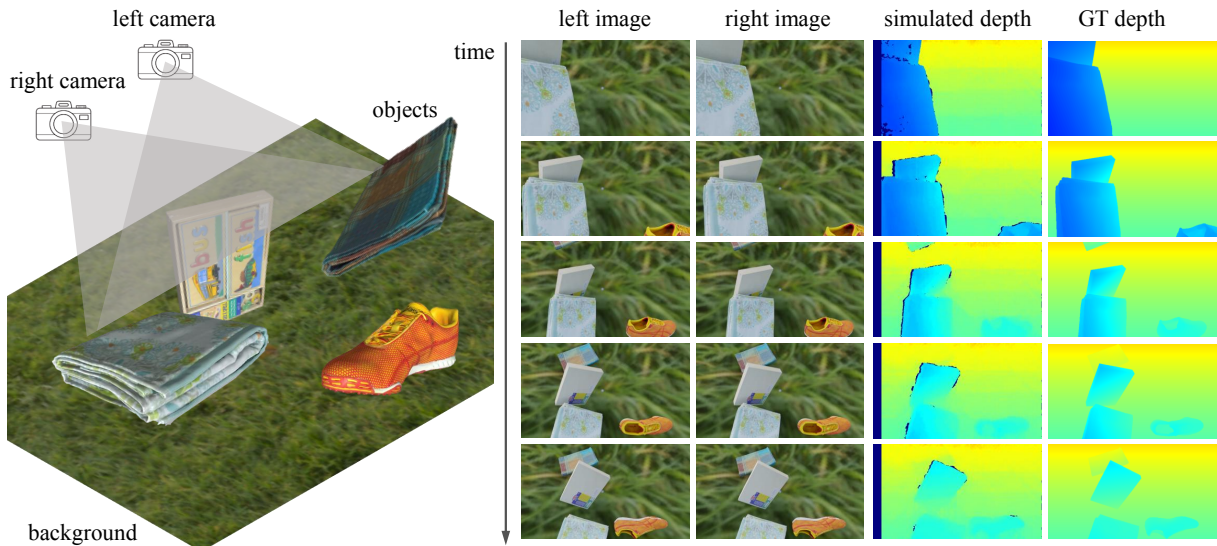


Fig. 2: The pipeline of RaSim. Given the virtual scene constructed by objects, background, and global illumination, the left and right cameras take videos under chronological physical simulation. Subsequently, the simulated depth maps are generated by the semi-global stereo-matching algorithm from binocular images.

is due to the discrepancy between the left and right IR images becoming inconspicuous, along with the reduced environmental illumination, making the stereo-matching procedure error-prone. To alleviate this problem, we propose a range-aware rendering strategy. Recapping Fig. 1, for nearby scenes where the camera and objects are close, we perform stereo matching with IR images. While for distant scenes, the matching is based on binocular RGB images with richer texture information and brighter light illumination. Given rectified stereo images, RaSim first applies the center-symmetric census transform [37], followed by a semi-global stereo-matching [38] algorithm for disparity estimation. Subsequently, the disparity is further refined by median filtering and consistency checks before the conversion into depth. This range-aware rendering strategy enriches the diversity of the dataset, yielding improvements in the versatility of the RaSim pipeline.

As shown in Fig. 2, we denote  $\mathcal{C}^{\{L,R\}}$  as the left and right stereo cameras,  $\mathcal{S}$  as the virtual scene,  $T$  as the total frames per rendering,  $\mathbf{Z}_{gt}$  as the ground-truth depth, and  $\mathbf{I}^{\{L,R\}}$  as the corresponding rendered left and right images in the format of IR or RGB. The rendering procedure can be formulated as follows

$$\begin{aligned} \mathbf{Z}_{gt} &= \{\mathbf{Z}_t \mid \mathbf{Z}_t = \text{Render}(\mathcal{C}^L, \mathcal{S}_t)\}_{t=1}^T, \\ \mathbf{I}^{\{L,R\}} &= \{\mathbf{I}_t \mid \mathbf{I}_t = \text{Render}(\mathcal{C}^{\{L,R\}}, \mathcal{S}_t)\}_{t=1}^T, \\ \mathcal{S}_t &= \mathcal{P}_t \circ (\mathcal{O}, \mathcal{B}, \mathcal{L}). \end{aligned} \quad (2)$$

Thereby,  $\mathcal{O}$  is the objects selected from the GSO [39] dataset,  $\mathcal{B}$  is an indoor or outdoor background selected from either rooms textured by the CC0textures Library or scenes in Poly Haven<sup>1</sup>,  $\mathcal{L}$  is the global illumination varying with the environment, and  $\mathcal{P}_t$  stands for physical simulation working on all the assets at frame  $t$ .

<sup>1</sup><https://polyhaven.com/hdris>

The pipeline is implemented with Kubric [14], a dataset generator interfacing with Blender [11] and PyBullet [40].

### C. RaSim Dataset

Driven by RaSim, we create a large-scale synthetic RGB-D dataset with domain randomization and physically-based rendering techniques. It comprises more than 206K images distributed across 9,835 diverse scenes. Each image is annotated with pixel-level semantic information, alongside both simulated and ground-truth depth maps, and other meta information *w.r.t.* scene generation. Moreover, one million instances featured with CAD models, poses, categories, and 3D coordinates are also included. Thanks to rich annotations, the dataset can be applied to numerous 3D vision tasks including object manipulation [13], unseen pose estimation [41], [42], and 3D semantic segmentation [1], [43].

## IV. DOWNSTREAM TASKS

In this section, we introduce two downstream tasks, *i.e.*, depth completion and depth pre-training, where our RaSim dataset effectively addresses the data scarcity issue and assists in real-world applications.

### A. Depth Completion

As depicted in Fig. 3, taking an RGB image  $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$  and the repeated simulated depth  $\mathbf{Z}_{sim} \in \mathbb{R}^{H \times W \times 3}$  as input, our SDRNet first respectively extracts hierarchical color and depth features with two Swin Transformer [17] backbones named SwinC and SwinD. The features are then concatenated and fed into two UPerNet [44] based decoders predicting a coarse depth map  $\mathbf{Z}_{est}^c$  and a confidence map  $\mathbf{C}$  as used in [21]. The final depth prediction is composed of the input and predicted depth as

$$\mathbf{Z}_{est}^f = (1 - \mathbf{C}) \otimes \mathbf{Z}_{sim} + \mathbf{C} \otimes \mathbf{Z}_{est}^c, \quad (3)$$

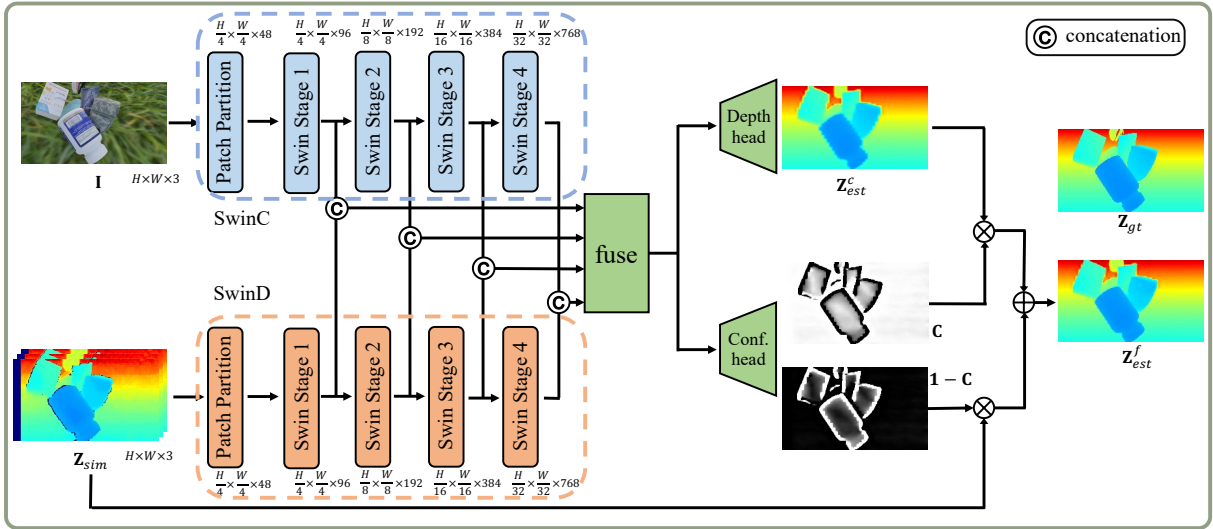


Fig. 3: The architecture of SDRNet.

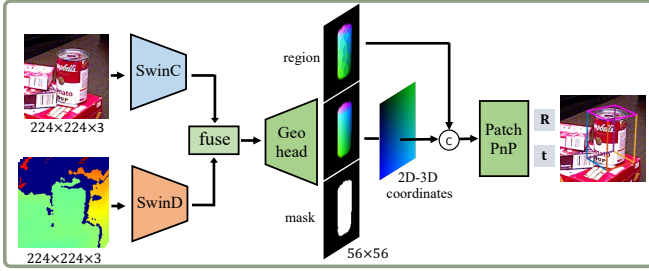


Fig. 4: The architecture for object pose estimation.

where  $\otimes$  denotes element-wise production.

Apart from the ground-truth depth, surface normal and gradient are also derived to supervise the coarse  $Z_{est}^c$  and fine  $Z_{est}^f$  depth estimation. The loss function is written as

$$\begin{aligned} \mathcal{L} &= \mathcal{L}^f + w_c \mathcal{L}^c, \\ \mathcal{L}^{\{c,f\}} &= \mathcal{L}_Z^{\{c,f\}} + w_n \mathcal{L}_N^{\{c,f\}} + w_g \mathcal{L}_G^{\{c,f\}}, \end{aligned} \quad (4)$$

where  $\mathcal{L}_Z, \mathcal{L}_N, \mathcal{L}_G$  denote the  $\mathcal{L}_1$  losses of ground-truth and estimated depth, surface normal and gradient, and  $w_c, w_n, w_g$  are the loss factors. This optimization target enables  $Z_{est}^c$  to target the easily predictable area like the background while  $Z_{est}^f$  could focus on the challenging area like the edge of objects.

Notably, the network is trained with pure synthetic data and tested with the ClearGrasp dataset [8] collected from the real world.

### B. Depth Pre-training

To alleviate the lack of data, one de-facto paradigm is to pre-train neural networks on large-scale datasets and finetune on downstream task-specific datasets. Inspired by the idea of *masked language modeling*, i.e., masking part of the data and then predicting the invisible content according to context, we introduce the simulated depth restoration as a pre-text task for depth-based Transformer pre-training. Specifically, we first pre-train an SDRNet with the proposed RaSim

dataset. After pre-training, two homogeneous Swin backbones, i.e., SwinC to encode color information and SwinD to encode depth information, are initialized by ImageNet-21K [45] and our RaSim dataset pre-trained weights separately. In this way, the pre-trained SwinD backbone gains prior knowledge of 3D geometric structures, thus benefiting various 3D tasks.

We choose object pose estimation as a verification task, for which collecting real-world annotations is oftentimes very expensive. The objective of pose estimation is solving the 6DoF object pose, i.e., 3DoF rotation and 3DoF translation, in the camera coordinate system. As shown in Fig. 4, the features extracted from zoomed-in RGB-D images are first aggregated and then sent to a geometric head to decode surface region, 3D coordinate map, and object mask. Afterwards, the surface region along with a 2D-3D dense correspondence map is fed into a Patch-PnP module to solve allocentric continuous rotation  $\mathbf{R}$  and scale-invariant translation  $\mathbf{t}$  as used in [7], [46], [47].

## V. EXPERIMENTS

### A. Depth Completion

**Implementation Details.** The model is trained on our RaSim dataset with the backbone of Swin-tiny (Swin-T). To adapt the input scale of the backbone, we resize the depth map to  $224 \times 224$  or  $512 \times 512$  with nearest-neighbor interpolation. We employ the Ranger optimizer [48], [49], [50] with a learning rate of  $1 \times 10^{-4}$  and a weight decay of 0.01. The training epoch is set to 10 with a batch size of 32.

**Dataset.** We evaluate SDRNet with the ClearGrasp [8] test split, which contains 286 real-world RGB-D images of transparent objects along with their corresponding ground-truth depth maps.

**Evaluation Metrics.** We follow the evaluation protocol of [8], [9]. The predicted and ground-truth depth maps are first resized to  $144 \times 256$ , and we use four evaluation metrics: (1) root mean squared error (RMSE), (2) absolute relative difference (REL), (3) mean absolute error (MAE), and (4)

TABLE I: Comparison with state-of-the-art methods on ClearGrasp.  $\downarrow$  means lower is better,  $\uparrow$  means higher is better. RGBD-FCN and [51] are implemented by [9].

Methods	RMSE $\downarrow$	REL $\downarrow$	MAE $\downarrow$	$\delta_{1.05}\uparrow$	$\delta_{1.10}\uparrow$	$\delta_{1.25}\uparrow$
ClearGrasp Real-known						
RGBD-FCN	0.054	0.087	0.048	36.32	67.11	96.26
NLSPN [51]	0.149	0.228	0.127	14.04	26.67	54.32
CG [8]	0.039	0.051	0.029	72.62	86.96	95.58
LIF [9]	0.028	0.033	0.020	82.37	92.98	98.63
DREDS [21]	0.022	<b>0.017</b>	0.012	91.46	<b>97.47</b>	<b>99.86</b>
Ours	<b>0.021</b>	<b>0.017</b>	<b>0.011</b>	<b>94.14</b>	<b>97.47</b>	99.58
ClearGrasp Real-novel						
RGBD-FCN	0.042	0.070	0.037	42.45	75.68	99.02
NLSPN [51]	0.145	0.240	0.123	13.77	25.81	51.59
CG [8]	0.034	0.045	0.025	76.72	91.00	97.63
LIF [9]	0.025	0.036	0.020	76.21	94.01	99.35
DREDS [21]	0.016	<b>0.008</b>	<b>0.005</b>	<b>96.73</b>	<b>98.83</b>	99.78
Ours	<b>0.014</b>	0.010	<b>0.005</b>	95.74	98.26	<b>99.87</b>

the threshold  $\delta$  which satisfies  $\delta > \max(\frac{\tilde{d}_i}{d_i}, \frac{d_i}{\tilde{d}_i})$ , where  $\delta \in \{1.05, 1.10, 1.25\}$ ,  $d_i$  and  $\tilde{d}_i$  denote ground-truth and predicted depths.

**Comparison with State of the Arts.** We compare our method with several top-performing methods in Table I. Our SDRNet exceeds previous state-of-the-art methods [8], [9] by a large margin and achieves comparable results with [21]. Note that [8], [9] are trained with data from ClearGrasp, while our network is trained exclusively on the synthetic RaSim dataset and demonstrates superior performance when transferred to real-world scenes. These results confirm the high quality of the RaSim dataset and its effectiveness in bridging the sim-to-real domain gap.

### B. Depth Pre-training for Object Pose Estimation

**Implementation Details.** The experiments are implemented with PyTorch [52]. Pose estimation models are trained for 12 epochs using the Ranger optimizer with a batch size of 24 and a learning rate of  $1 \times 10^{-4}$ , annealing at 50% of the training phase leveraging a cosine schedule [53]. The objects of interest are obtained using the detection results of YOLOX [54]. In all experiments, one model is trained for all objects.

**Dataset.** The experiments are conducted on the widely used YCB-V [5] dataset. It comprises more than 110K images in 92 RGB-D videos spanning 21 selected objects from the YCB object set. The dataset is challenging due to severe occlusions, symmetric objects, variable lighting conditions, and noisy depth. Additionally, we also use the publicly available PBR images [55], [15] to aid training.

**Evaluation Metrics.** We use the most common metric ADD and its variants for evaluation. The error of ADD metric [56], [57] calculates the average distance of the object vertices transformed by the ground-truth pose  $[\mathbf{R}|\mathbf{t}]$  and the estimated pose  $[\tilde{\mathbf{R}}|\tilde{\mathbf{t}}]$

$$e_{\text{ADD}} = \frac{1}{N_v} \sum_{i=1}^{N_v} \|(\mathbf{R}\mathbf{x}_i + \mathbf{t}) - (\tilde{\mathbf{R}}\mathbf{x}_i + \tilde{\mathbf{t}})\|. \quad (5)$$

It is considered correct if  $e_{\text{ADD}}$  is below 10% of the object diameter. For symmetric objects, the  $e_{\text{ADD-S}}$  is employed

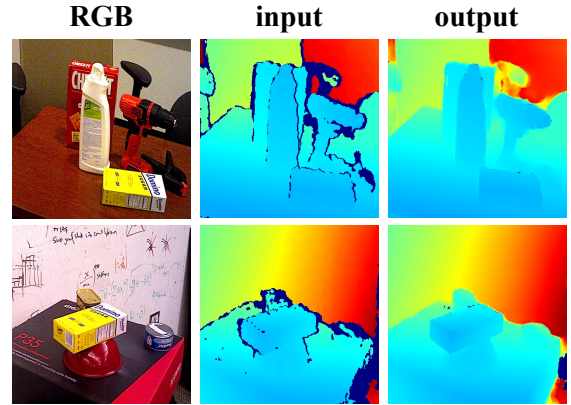


Fig. 5: Visualization results of depth restoration on YCB-V.

based on the distance to the closest model point

$$e_{\text{ADD-S}} = \frac{1}{N_v} \sum_{i=1}^{N_v} \min_{\mathbf{x}_j \in \mathbb{V}} \|(\mathbf{R}\mathbf{x}_i + \mathbf{t}) - (\tilde{\mathbf{R}}\mathbf{x}_j + \tilde{\mathbf{t}})\|. \quad (6)$$

We also report the area under curve (AUC) of ADD and ADD-S metrics by varying the distance threshold from 0cm to 10cm following [5].

**Zero-shot Depth Restoration on YCB-V.** To appraise whether the pre-trained model has a transfer ability on datasets in different domains, we perform depth restoration experiments on the YCB-V dataset. Although the lack of ground-truth depth constraints the calculation of quantitative results, the qualitative results still reveal that the SDRNet trained from the synthetic dataset could generalize well in real-world scenarios, as shown in Fig. 5. It further proves the broad usage of the proposed RaSim dataset.

**Comparison with State of the Arts.** Table II presents our quantitative results with other state-of-the-art methods [58], [59], [60], [61] on YCB-V. Our pre-trained model achieves 98.2% on AUC of ADD-S and 93.8% on AUC of ADD(-S), surpassing all the compared methods without any time-consuming refinement procedure. Note that [58], [60] focus on designing complex fusion strategies for color and depth features, while [61] directly concatenates the RGB and depth map and feeds them into the network. However, our strategy is moderate yet more reasonable: leveraging the homogeneous Transformer backbones to extract multimodal features while initializing them with heterogeneous pre-trained weights. This simplifies the network architecture yet maintains high accuracy and efficiency.

**Ablation studies.** Table III illustrates several ablations *w.r.t.* depth pre-training strategies. We can observe that initializing SwinD with ImageNet pre-trained weights brings slight enhancement over PyTorch’s default random initialization (Table III A1 *v.s.* A0). Nevertheless, our depth pre-training shows more distinct superiority, achieving an enhancement of 3.7% on ADD(-S) metric and 2.9% on the AUC of ADD(-S) metric (Table III B0 *v.s.* A0).

Aside from pre-training on RaSim, a simpler approach is applying randomization after rendering depth (Ren & Ran), like adding Gaussian noise or randomly dropping

TABLE II: **Comparison with state of the arts on YCB-V.** Here ADD(-S) uses the symmetric metric only for symmetric objects (denoted with \*), while ADD-S uses the symmetric metric for all objects.

Object	DenseFusion [58]		PVN3D [59]		FFB6D [60]		Uni6D [61]		Ours	
	AUC of ADD-S	AUC of ADD(-S)	AUC of ADD-S	AUC of ADD(-S)	AUC of ADD-S	AUC of ADD(-S)	AUC of ADD-S	AUC of ADD(-S)	AUC of ADD-S	AUC of ADD(-S)
002_master_chef_can	95.3	70.7	96.0	80.5	96.3	<b>80.6</b>	95.4	70.2	<b>100.0</b>	80.2
003_cracker_box	92.5	86.9	96.1	<b>94.8</b>	<b>96.3</b>	94.6	91.8	85.2	92.1	71.9
004_sugar_box	95.1	90.8	97.4	96.3	97.6	96.6	96.4	94.5	<b>99.9</b>	<b>99.6</b>
005_tomato_soup_can	93.8	84.7	96.2	88.5	95.6	89.6	95.8	85.4	<b>98.7</b>	<b>96.7</b>
006_mustard_bottle	95.8	90.9	97.5	96.2	97.8	97.0	95.4	91.7	<b>100.0</b>	<b>100.0</b>
007_tuna_fish_can	95.7	79.6	96.0	89.3	96.8	88.9	95.2	79.0	<b>100.0</b>	<b>98.8</b>
008_pudding_box	94.3	89.3	97.1	<b>95.7</b>	97.1	94.6	94.1	89.8	<b>99.2</b>	92.7
009_gelatin_box	97.2	95.8	97.7	96.1	98.1	96.9	97.4	96.2	<b>100.0</b>	<b>99.9</b>
010_potted_meat_can	89.3	79.6	93.3	88.6	94.7	88.1	93.0	89.6	<b>95.9</b>	<b>90.6</b>
011_banana	90.0	76.7	96.6	93.7	97.2	94.9	96.4	93.0	<b>100.0</b>	<b>99.2</b>
019_pitcher_base	93.6	87.1	97.4	96.5	97.6	96.9	96.2	94.2	<b>100.0</b>	<b>99.8</b>
021_bleach_cleanser	94.4	87.5	96.0	93.2	96.8	94.8	95.2	91.1	<b>99.1</b>	<b>95.8</b>
024_bowl*	86.0	86.0	90.2	90.2	<b>96.3</b>	<b>96.3</b>	95.5	95.5	94.0	94.0
025_mug	95.3	83.8	<b>97.6</b>	<b>95.4</b>	97.3	94.2	96.6	93.0	96.1	95.0
035_power_drill	92.1	83.7	96.7	95.1	97.2	95.9	94.7	91.1	<b>99.9</b>	<b>96.4</b>
036_wood_block*	89.5	89.5	90.4	90.4	92.6	92.6	94.3	94.3	<b>96.9</b>	<b>96.9</b>
037_scissors	90.1	77.4	96.7	92.7	<b>97.7</b>	<b>95.7</b>	87.6	79.6	94.6	70.6
040_large_marker	95.1	89.1	96.7	91.8	96.6	89.1	96.7	92.8	<b>99.8</b>	<b>94.3</b>
051_large_clamp*	71.5	71.5	93.6	93.6	96.8	96.8	95.9	95.9	<b>99.2</b>	<b>99.2</b>
052_extra_large_clamp*	70.2	70.2	88.4	88.4	96.0	96.0	95.8	95.8	<b>97.6</b>	<b>97.6</b>
061_foam_brick*	92.2	92.2	96.8	96.8	97.3	97.3	96.1	96.1	<b>99.9</b>	<b>99.9</b>
Avg	91.2	82.9	95.5	91.8	96.6	92.7	95.2	88.8	<b>98.2</b>	<b>93.8</b>

TABLE III: **Ablations on depth pre-training strategies.**

We report the results of ADD(-S), and AUC of ADD-S and ADD(-S) metrics on the YCB-V dataset. Ren & Ran is short for randomization after rendering depth.

Row	SwinC	SwinD	ADD(-S)	AUC of ADD(-S)	AUC of ADD-S
A0	ImageNet-21K	Random	81.8	90.9	97.6
A1	ImageNet-21K	ImageNet-21K	82.5	92.8	97.8
B0	ImageNet-21K	RaSim	<b>85.5</b>	<b>93.8</b>	<b>98.2</b>
B1	ImageNet-21K	Ren & Ran	84.6	93.3	97.6
C0	ImageNet-21K	Stereo IR Split	83.9	93.4	98.0
C1	ImageNet-21K	Stereo RGB Split	83.7	93.6	97.9

depth values. We observe this straightforward strategy is also effective (Table III B1 v.s. A0), but worse than the performance of RaSim initialization (Table III B1 v.s. B0). This result reveals that the RaSim pipeline effectively shrinks the sim-to-real domain gap.

As mentioned in Sec. III-B, the range-aware rendering strategy broadens the depth range and enriches data diversity. When the network is pre-trained solely on the stereo IR split (Table III C0) or RGB split (Table III C1), the performance drop distinctly on the ADD(-S) metric.

We illustrate the results of the baseline and our pre-training v.s. iterations in Fig. 6. As is depicted, pre-training with the RaSim significantly boosts the performance, especially in the early stage of training. This indicates that pre-training on the RaSim dataset effectively equips the Transformer-based backbone with prior 3D geometric knowledge.

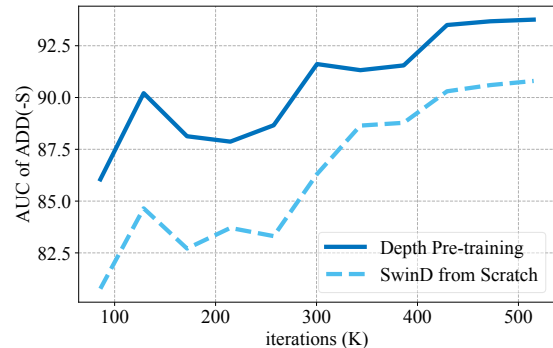


Fig. 6: AUC of ADD(-S) v.s. iterations on YCB-V.

## VI. CONCLUSION

This work has introduced RaSim, a range-aware RGB-D data simulation pipeline that excels in producing high-fidelity RGB-D data. By imitating the imaging principle of real-world depth sensors, we effectively bridge the sim-to-real domain gap concerning depth maps. Notably, we incorporate a range-aware rendering strategy to enrich data diversity, making RaSim generalizable to a broader range of real-world application scenarios. Experiments on 3D perception tasks demonstrate that models trained with RaSim can be directly applied to real-world datasets like ClearGrasp and YCB-V without the need for finetuning. In the future, we aim to explore the simulation of more types of depth sensors and expand RaSim to more diverse applications.

**Acknowledgments.** This work was supported by the National Key R&D Program of China under Grant 2018AAA0102801.

## REFERENCES

- [1] H. Liu, J. Zhang, K. Yang, X. Hu, and R. Stiefelhagen, "CMX: Cross-Modal Fusion for RGB-X Semantic Segmentation with Transformers," *arXiv preprint arXiv:2203.04838*, 2022.
- [2] L.-Z. Chen, Z. Lin, Z. Wang, Y.-L. Yang, and M.-M. Cheng, "Spatial information guided convolution for real-time rgbd semantic segmentation," *IEEE Transactions on Image Processing*, vol. 30, pp. 2313–2324, 2021.
- [3] X. Chen, K.-Y. Lin, J. Wang, W. Wu, C. Qian, H. Li, and G. Zeng, "Bi-directional cross-modality feature propagation with separation-and-aggregation gate for rgb-d semantic segmentation," in *European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 561–577.
- [4] M. Sodano, F. Magistri, T. Guadagnino, J. Behley, and C. Stachniss, "Robust double-encoder network for rgb-d panoptic segmentation," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4953–4959.
- [5] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," *Robotics: Science and Systems Conference (RSS)*, 2018.
- [6] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "CosyPose: Consistent multi-view multi-object 6D pose estimation," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 574–591.
- [7] G. Wang, F. Manhardt, F. Tombari, and X. Ji, "GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 16611–16621.
- [8] S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song, "Clear grasp: 3d shape estimation of transparent objects for manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3634–3642.
- [9] L. Zhu, A. Mousavian, Y. Xiang, H. Mazhar, J. van Eenbergen, S. Debnath, and D. Fox, "Rgb-d local implicit function for depth completion of transparent objects," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 4649–4658.
- [10] M. Hu, S. Wang, B. Li, S. Ning, L. Fan, and X. Gong, "Penet: Towards precise and efficient image guided depth completion," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13656–13662.
- [11] Blender Online Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Blender Institute, Amsterdam, 2021. [Online]. Available: <http://www.blender.org>
- [12] F. Hagelskjær and A. G. Buch, "Bridging the reality gap for pose estimation networks using sensor-based domain randomization," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 935–944.
- [13] T. Kollar, M. Laskey, K. Stone, B. Thananjeyan, and M. Tjersland, "Simnet: Enabling robust unknown object manipulation from pure synthetic data via stereo," in *Conference on Robot Learning (CoRL)*. PMLR, 2022, pp. 938–948.
- [14] K. Greff, F. Belletti, B. Beyer, C. Doersch, Y. Du, D. Duckworth, D. J. Fleet, D. Gnanaprasgam, F. Golemo, C. Herrmann, T. Kipf, A. Kundu, D. Lagun, I. Laradji, H.-T. D. Liu, H. Meyer, Y. Miao, D. Nowrouzezahrai, C. Ozireli, E. Pot, N. Radwan, D. Rebain, S. Sabour, M. S. M. Sajjadi, M. Sela, V. Sitzmann, A. Stone, D. Sun, S. Vora, Z. Wang, T. Wu, K. M. Yi, F. Zhong, and A. Tagliasacchi, "Kubric: a scalable dataset generator," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [15] T. Hodaň, V. Vineet, R. Gal, E. Shalev, J. Hanzelka, T. Connell, P. Urbina, S. Sinha, and B. Guenter, "Photorealistic Image Synthesis for Object Instance Detection," *IEEE International Conference on Image Processing (ICIP)*, 2019.
- [16] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," in *Conference on Robot Learning (CoRL)*, 2018, pp. 306–316.
- [17] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 10012–10022.
- [18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [19] A. Eftekhar, A. Sax, J. Malik, and A. Zamir, "Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 10786–10796.
- [20] Z. Li, T.-W. Yu, S. Sang, S. Wang, M. Song, Y. Liu, Y.-Y. Yeh, R. Zhu, N. Gundavarapu, J. Shi *et al.*, "Openrooms: An open framework for photorealistic indoor scene datasets," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 7190–7199.
- [21] Q. Dai, J. Zhang, Q. Li, T. Wu, H. Dong, Z. Liu, P. Tan, and H. Wang, "Domain randomization-enhanced depth simulation and restoration for perceiving and grasping specular and transparent objects," in *European Conference on Computer Vision (ECCV)*, 2022.
- [22] X. Zhang, R. Chen, A. Li, F. Xiang, Y. Qin, J. Gu, Z. Ling, M. Liu, P. Zeng, S. Han *et al.*, "Close the optical sensing domain gap by physics-grounded active stereo sensor simulation," *IEEE Transactions on Robotics (T-RO)*, 2023.
- [23] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12627–12637.
- [24] M. Rad, M. Oberweger, and V. Lepetit, "Feature mapping for learning fast and accurate 3d pose inference from synthetic images," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [25] X. Deng, S. Yang, Y. Zhang, P. Tan, L. Chang, and H. Wang, "Hand3d: Hand pose estimation using 3d neural network," *arXiv preprint arXiv:1704.02224*, 2017.
- [26] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6D object pose and size estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2642–2651.
- [27] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1746–1754.
- [28] M. Roberts, J. Ramapuram, A. Ranjan, A. Kumar, M. A. Bautista, N. Paczan, R. Webb, and J. M. Susskind, "Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 10912–10922.
- [29] A. Szot, A. Clegg, E. Undersander, E. Wijnmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets *et al.*, "Habitat 2.0: Training home assistants to rearrange their habitat," *Conference on Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 251–266, 2021.
- [30] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *IEEE/RIS International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 23–30.
- [31] X. Ren, J. Luo, E. Solowjow, J. A. Ojea, A. Gupta, A. Tamar, and P. Abbeel, "Domain randomization for active pose estimation," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7228–7234.
- [32] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 969–977.
- [33] M. Jaritz, T.-H. Vu, R. de Charette, E. Wirbel, and P. Pérez, "xMUDA: Cross-modal unsupervised domain adaptation for 3D semantic segmentation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [34] J. Yang, S. Shi, Z. Wang, H. Li, and X. Qi, "St3d: Self-training for unsupervised domain adaptation on 3d object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [35] X. Zhou, A. Karpur, C. Gan, L. Luo, and Q. Huang, "Unsupervised domain adaptation for 3d keypoint estimation via view consistency," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 137–153.
- [36] H. Hirschmüller, "Semi-global matching-motivation, developments and applications," *Photogrammetric Week 11*, pp. 173–184, 2011.

- [37] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *European Conference on Computer Vision (ECCV)*. Springer, 1994, pp. 151–158.
- [38] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 30, no. 2, pp. 328–341, 2007.
- [39] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke, "Google scanned objects: A high-quality dataset of 3d scanned household items," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2553–2560.
- [40] E. Coumans and Y. Bai, "PyBullet, a python module for physics simulation for games, robotics and machine learning," <http://pybullet.org>, 2016–2020.
- [41] Y. Labbé, L. Manuelli, A. Mousavian, S. Tyree, S. Birchfield, J. Tremblay, J. Carpentier, M. Aubry, D. Fox, and J. Sivic, "Megapose: 6d pose estimation of novel objects via render & compare," in *Conference on Robot Learning (CoRL)*. PMLR, 2023, pp. 715–725.
- [42] Y. Liu, Y. Wen, S. Peng, C. Lin, X. Long, T. Komura, and W. Wang, "Gen6d: Generalizable model-free 6-dof object pose estimation from rgb images," in *European Conference on Computer Vision (ECCV)*. Springer, 2022, pp. 298–315.
- [43] Y. Wang, X. Chen, L. Cao, W. Huang, F. Sun, and Y. Wang, "Multimodal token fusion for vision transformers," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12 186–12 195.
- [44] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 418–434.
- [45] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "Imagenet-21k pretraining for the masses," *arXiv preprint arXiv:2104.10972*, 2021.
- [46] Y. Di, F. Manhardt, G. Wang, X. Ji, N. Navab, and F. Tombari, "SO-Pose: Exploiting self-occlusion for direct 6D pose estimation," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 12 396–12 405.
- [47] D. Gao, Y. Li, P. Ruhkamp, I. Skobleva, M. Wysocki, H. Jung, P. Wang, A. Guridi, and B. Busam, "Polarimetric pose prediction," in *European Conference on Computer Vision (ECCV)*. Springer, 2022, pp. 735–752.
- [48] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," in *International Conference on Learning Representations (ICLR)*, 2019.
- [49] M. Zhang, J. Lucas, J. Ba, and G. E. Hinton, "Lookahead optimizer: k steps forward, 1 step back," in *Conference on Neural Information Processing Systems (NeurIPS)*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.
- [50] H. Yong, J. Huang, X. Hua, and L. Zhang, "Gradient centralization: A new optimization technique for deep neural networks," in *European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 635–652.
- [51] J. Park, K. Joo, Z. Hu, C.-K. Liu, and I. S. Kweon, "Non-local spatial propagation network for depth completion," in *European Conference on Computer Vision (ECCV)*, 2020.
- [52] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Conference on Neural Information Processing Systems (NeurIPS)*, 2019, pp. 8026–8037.
- [53] F. H. Ilya Loshchilov, "SGDR: stochastic gradient descent with warm restarts," in *International Conference on Learning Representations (ICLR)*, 2017.
- [54] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.
- [55] T. Hodan, M. Sundermeyer, B. Drost, Y. Labbe, E. Brachmann, F. Michel, C. Rother, and J. Matas, "BOP Challenge 2020 on 6D object localization," in *European Conference on Computer Vision Workshops (ECCVW)*, A. Bartoli and A. Fusiello, Eds., 2020, pp. 577–594.
- [56] S. Hinterstoisser, V. Lepetit, S. Ilic, K. Konolige, K. Konolige, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Asian Conference on Computer Vision (ACCV)*, 2012, pp. 548–562.
- [57] T. Hodaň, J. Matas, and Š. Obdržálek, "On evaluation of 6d object pose estimation," *European Conference on Computer Vision Workshops (ECCVW)*, pp. 606–619, 2016.
- [58] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3343–3352.
- [59] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "PVN3D: A deep point-wise 3d keypoints voting network for 6dof pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 632–11 641.
- [60] Y. He, H. Huang, H. Fan, Q. Chen, and J. Sun, "FFB6D: A full flow bidirectional fusion network for 6d pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 3003–3013.
- [61] X. Jiang, D. Li, H. Chen, Y. Zheng, R. Zhao, and L. Wu, "Uni6d: A unified cnn framework without projection breakdown for 6d pose estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11 174–11 184.