Domain Randomization-Enhanced Depth Simulation and Restoration for Perceiving and Grasping Specular and Transparent Objects

Qiyu Dai[1,*], Jiyao Zhang[2,*], Qiwei Li[1], Tianhao Wu[1], Hao Dong[1], Ziyuan Liu[3], Ping Tan[3,4], and He Wang[1,†]

[1] Peking University [2] Xi'an Jiaotong University
[3] Alibaba XR Lab [4] Simon Fraser University
{qiyudai,lqw,hao.dong,hewang}@pku.edu.cn,
zhangjiyao@stu.xjtu.edu.cn, thwu@stu.pku.edu.cn,
ziyuan-liu@outlook.com, pingtan@sfu.ca

Abstract. Commercial depth sensors usually generate noisy and missing depths, especially on specular and transparent objects, which poses critical issues to downstream depth or point cloud-based tasks. To mitigate this problem, we propose a powerful RGBD fusion network, SwinDRNet, for depth restoration. We further propose Domain Randomization-Enhanced Depth Simulation (DREDS) approach to simulate an active stereo depth system using physically based rendering and generate a large-scale synthetic dataset that contains 130K photorealistic RGB images along with their simulated depths carrying realistic sensor noises. To evaluate depth restoration methods, we also curate a real-world dataset, namely STD, that captures 30 cluttered scenes composed of 50 objects with different materials from specular, transparent, to diffuse. Experiments demonstrate that the proposed DREDS dataset bridges the sim-to-real domain gap such that, trained on DREDS, our SwinDRNet can seamlessly generalize to other real depth datasets, e.g. ClearGrasp, and outperform the competing methods on depth restoration. We further show that our depth restoration effectively boosts the performance of downstream tasks, including category-level pose estimation and grasping tasks. Our data and code are available at https://github.com/PKU-EPIC/DREDS.

Keywords: Depth sensor simulation, specular and transparent objects, domain randomization, pose estimation, grasping

1 Introduction

With the emerging depth-sensing technologies, depth sensors and 3D point cloud data become more and more accessible, rendering many applications in VR/AR and robotics. Compared with RGB images, depth images or point clouds contain the true 3D information of the underlying scene geometry, thus depth cameras

*: equal contributions, †: corresponding author

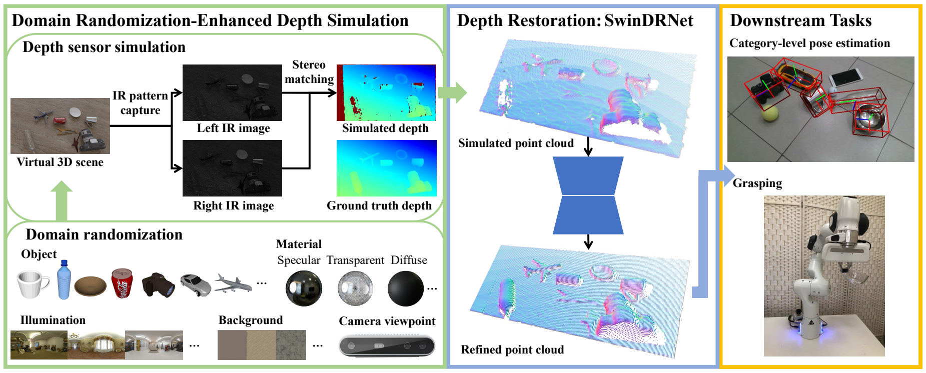

Fig. 1. Framework overview. From the left to right: we leverage domain randomization-enhanced depth simulation to generate paired data, on which we can train our depth restoration network SwinDRNet, and the restored depths will be fed to downstream tasks and improve estimating category-level pose and grasping for specular and transparent objects.

have been widely deployed in many robotic systems, *e.g.* for object grasping [3,11] and manipulation [32,18,17], that care about the accurate scene geometry. However, an apparent disadvantage of accessible depth cameras is that they may carry non-ignorable sensor noises more significant than usual noises in colored images captured by commercial RGB cameras. A more drastic failure case of depth sensing would be on objects that are either transparent or their surfaces are highly specular, where the captured depths would be highly erroneous and even missing around the specular or transparent region. It should be noted that specular and transparent objects are indeed ubiquitous in our daily life, given most of the metallic surfaces are specular and many man-made objects are made of glasses and plastics which can be transparent. The existence of so many specular and transparent objects in our real-world scenes thus poses severe challenges to depth-based vision systems and limits their application scenarios to well-controlled scenes and objects made of diffuse materials.

In this work, we devise a two-stream Swin Transformer [15] based RGB-D fusion network, SwinDRNet, for learning to perform depth restoration. However, it is a lack of real data composed of paired sensor depths and perfect depths to train such a network. Previous works on depth completion for transparent objects, like ClearGrasp [26] and LIDF [38], leverage synthetic perfect depth image for network training. They simply remove the transparent area in the perfect depth and their methods then learn to complete the missing depths in a feed-forward way or further combines with depth optimization. We argue that both the methods can only access incomplete depth images during training and never see a depth with realistic sensor noises, leading to suboptimality when directly deployed on real sensor depths. Also, these two works only consider a small number of similar objects with little shape variations and all being transparent and hence fail to demonstrate their usefulness when adopted in scenes with completely novel object instances. Given material specularity or transparency forms a continuous spectrum, it is further questionable whether their methods can handle objects of intermediate transparency or specularity.

To mitigate the problems in the existing works, we thus propose to synthesize depths with realistic sensor noise patterns by simulating an active stereo depth camera resembling RealSense D415. Our simulator is built on Blender and leverages raytracing to mimic the IR stereo patterns and compute the depths from them. To facilitate generalization, we further adopt domain randomization techniques that randomize the object textures, object materials (from specular, transparent, to diffuse), object layout, floor textures, illuminations along camera poses. This domain randomization-enhanced depth simulation method, or in short DREDS, leads to 130K photorealistic RGB images and their corresponding simulated depths. We further curate a real-world dataset, STD dataset, that contains 50 objects with specular, transparent, and diffuse material. Our extensive experiments demonstrate that our SwinDRNet trained on DREDS dataset can handle depth restoration on object instances from both seen and unseen object categories in STD dataset and can even seamlessly generalize to ClearGrasp dataset and beat the previous state-of-the-art method, LIDF [38] trained on ClearGrasp dataset. Our further experiments on estimating category-level pose and grasping specular and transparent objects prove that our depth restoration is both generalizable and successful.

2 Related Work

2.1 Depth Estimation and Restoration

The increasing popularity of RGBD sensors has encouraged much research on depth estimation and restoration. Many works [7,12,16] directly estimate the depth from a monocular RGB image, but fail to restore accurate geometries of the point cloud because of the few geometric constraints of the color image. Other studies [19,33,25] restore the dense depth map given the RGB image and the sparse depth from LiDAR, but the estimated depth still suffers from low quality due to the limited geometric guidance of the sparse input. Recent research focuses on commercial depth sensors, trying to complete and refine the depth values from the RGB and noisy dense depth images. Sajjan *et al.* [26] proposed a two-stage method for transparent object depth restoration, which firstly estimates surface normals, occlusion boundaries, and segmentations from RGB images, and then calculates the refined depths via global optimization. However, the optimization is time-consuming, and heavily relies on the previous network predictions. Zhu *et al.* [38] proposed an implicit transparent object depth completion model, including the implicit representation learning from ray-voxel pairs and the self-iterating refinement, but voxelization of the 3D space results in heavy geometric discontinuity of the refined point cloud. Our method falls into this category and outperforms those methods, ensuring fast inference time and better geometries to improve the performance of downstream tasks.

2.2 Depth Sensor Simulation

To close the sim-to-real gap, the recent research focuses on generating simulated depth maps with realistic noise distribution. [14] simulated the pattern projection and capture system of Kinect to obtain simulated IR images and perform stereo matching, but could not simulate the sensor noise caused by object materials and scene environments. [22] proposed an end-to-end framework to simulate the mechanism of various types of depth sensors. However, the rasterization method limits the photorealistic rendering and physically correct simulation. [21] presented a new differentiable structure-light depth sensor simulation pipeline, but cannot simulate the transparent material, limited by the renderer. Recently, [37] proposed a physics-grounded active stereovision depth sensor simulator for various sim-to-real applications, but focused on instance-level objects and the robot arm workspace. Our DREDS pipeline generates realistic RGBD images for various materials and scene environments, which can generalize the proposed model to category-level unseen object instances and novel categories.

2.3 Domain Randomization

Domain randomization bridges the sim-to-real gap in the way of data augmentation. Tobin *et al.* [27] first explore transferring to real environments by generating training data through domain randomization. Subsequent works [28,35,23] generate synthetic data with sufficient variation by manually setting randomized features. Other studies [36] perform randomization using the neural networks. These works have verified the effectiveness of domain randomization on the tasks such as robotic manipulation [20], object detection and pose estimation [13], *etc.* In this work, we combine the depth sensor simulation pipeline with domain randomization, which, for the first time, enables direct generalization to unseen diverse real instances on specular and transparent object depth restoration.

3 Domain Randomization-Enhanced Depth Simulation

3.1 Overview

In this work, we propose a simulated RGBD data generation pipeline, namely Domain Randomization Enhanced Depth Simulation (DREDS), for tasks of depth restoration, object perception, and robotic grasping. We build a depth sensor simulator, modeling the mechanism of the active stereo vision depth camera system based on the physically based rendering, along with the domain randomization technique to handle real-world variations.

Leveraging domain randomization and active stereo sensor simulation, we present DREDS, the large-scale simulated RGBD dataset, containing photorealistic RGB images and depth maps with the real-world measurement noise and error, especially for the hand-scale objects with specular and transparent materials. The proposed DREDS dataset bridges the sim-to-real domain gap,

Table 1. Comparisons of specular and transparent depth restoration dataset. S, T, and D refer to specular, transparent, and diffuse materials, respectively. #Objects refers to the number of objects. SN+CG means the objects are selected from ShapeNet and ClearGrasp (the number are not mentioned).

Dataset	Type	#Objects	Type of Material	Size
ClearGrasp-Syn [26]	Syn	9	T	50K
Omniverse [38]	Syn	SN+CG	T+D	60K
ClearGrasp-Real [26]	Real	10	T	286
TODD [34]	Real	6	T	1.5K
DREDS	Sim	1,861	S+T+D	130K
STD	Real	50	S+T+D	27K

and generalizes the RGBD algorithms to unseen objects. DREDS dataset's comparison to the existing specular and transparent depth restoration datasets is summarized in Table 1.

3.2 Depth Sensor Simulation

A classical active stereo depth camera system contains an infrared (IR) projector, left and right IR stereo cameras, and a color camera. To measure the depth, the projector emits an IR pattern with dense dots to the scene. Subsequently, the two stereo cameras capture the left and right IR images, respectively. Finally, the stereo matching algorithm is used to calculate per-pixel depth values based on the discrepancy between the stereo images, to get the final depth scan. Our depth sensor simulator follows this mechanism, containing light pattern projection, capture, and stereo matching. The simulator is mainly built upon Blender [1].

Light Pattern Capture via physically based rendering. For real-world specular and transparent objects, the IR light from the projector may not be received by the stereo cameras, due to the reflection on the surface or the refraction through the transparent objects, resulting in inaccurate and missing depths. To simulate the physically correct IR pattern emission and capture process, we thus adopt physically based ray tracing, a technique that mimics the real light transportation process, and supports various surface materials especially specular and transparent materials.

Specifically, the textured spotlight projects a binary pattern image into the virtual scene. Sequentially, the binocular IR images are rendered from the stereo cameras. We manage to simulate IR images via visible light rendering, where both the light pattern and the reduced environment illumination contribute to the IR rendering. From the perspective of physics, the difference between IR and visible light is the reflectivity and refractive index of the object. We note that the wavelength (850 nm) of IR light used in depth sensors, *e.g.* RealSense D415, is close to the visible light (400-800 nm). So the resulting effects have already been well-covered by the randomization in object reflectivity and refractive index used in DREDS, which constructs a superset of real IR images. To mimic the portion of IR in environmental light, we reduce its intensity. Finally, all RGB values are converted to intensity, which is our final IR image.

Stereo Matching. We perform stereo matching to obtain the disparity map, which can be transferred to the depth map leveraging the intrinsic parameters of the depth sensor. In detail, we compute a matching cost volume over the left and right IR images along the epipolar line and find the matching results with minimum matching cost. Then we perform sub-pixel detection to generate a more accurate disparity map using the quadratic curve fitting method. To generate a more realistic depth map, we perform post-processing, including left/right consistency check, uniqueness constraint, median filtering, *etc.*

3.3 Simulated Data Generation with Domain Randomization

Based on the proposed depth sensor simulator, we formulate the simulated RGBD data generation pipeline as $D = Sim(\mathcal{S}, \mathcal{C})$, where $\mathcal{S} = \{\mathcal{O}, \mathcal{M}, \mathcal{L}, \mathcal{B}\}$ denotes scene-related simulation parameters in the virtual environment, including $\mathcal{O}$ the setting of the objects with random categories, poses, arrangements, and scales, $\mathcal{M}$ the setting of random object materials from specular, transparent, to diffuse, $\mathcal{L}$ the setting of environment lighting from varying scenes with different intensities, $\mathcal{B}$ the setting of background floor with diverse materials. $\mathcal{C}$ is the cameras' statue parameters, consisting of intrinsic and extrinsic parameters, the pattern image, baseline distance, *etc.* Taking these settings as input, the proposed simulator Sim generates the realistic RGB and depth images D.

To construct scenes with sufficient variations so that the proposed method can generalize to the real, we adopt domain randomization to enhance the generation, considering all these aspects. See supplementary materials for more details.

3.4 Simulated Dataset: DREDS

Making use of domain randomization and depth simulation, we construct the large-scale simulated dataset, DREDS. In total, DREDS dataset consists of two subsets: 1) **DREDS-CatKnown**: 100,200 training and 19,380 testing RGBD images made of 1,801 objects spanning 7 categories from ShapeNetCore [5], with randomized specular, transparent, and diffuse materials, 2) **DREDS-CatNovel**: 11,520 images of 60 category-novel objects, which is transformed from GraspNet-1Billion [8] that contains CAD models and annotates poses, by changing their object materials to specular or transparent, to verify the ability of our method to generalize to new object categories. Examples of paired simulated RGBD images of DREDS-Catknown and DREDS-CatNovel datasets are shown in Figure 2.

4 STD Dataset

4.1 Real-world Dataset: STD

To further examine the proposed method in real scenes, we curate a real-world dataset, composed of Specular, Transparent, and Diffuse objects, which we call

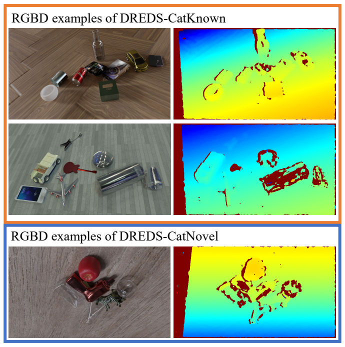
RGBD examples of DREDS-CatKnown
RGBD examples of DREDS-CatNovel

Fig. 2. RGBD examples of DREDS dataset.

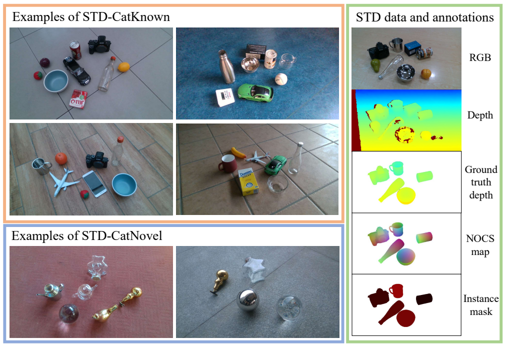

Fig. 3. Scene examples and annotations of STD dataset.

it STD dataset. Similar to DREDS dataset, STD dataset contains 1) **STD-CatKnown**: the subset with category-level objects, for the evaluation of depth restoration and category-level pose estimation tasks, and 2) **STD-CatNovel**: the subset with category-novel objects for evaluating the generalization ability of the proposed SwinDRNet method. Figure 3 shows the scene examples and annotations of STD dataset.

4.2 Data Collection

We collect an object set, covering specular, transparent, and diffuse materials. Specifically, for STD-CatKnown dataset, we collect 42 instances from 7 known ShapeNetCore [5] categories, and several category-unseen objects from the YCB dataset [4] and our own as the distractors. For STD-CatNovel dataset, we pick 8 specular and transparent objects from unseen categories. For each object except the distractors, we utilize the photogrammetry-based reconstruction tool, Object Capture API [2], to obtain its clean and accurate 3D mesh for ground truth poses annotation, so that we can yield ground truth depth and object masks.

We capture data from 30 different scenes (25 for STD-CatKnown, 5 for STD-CatNovel) with various backgrounds and illuminations, using RealSense D415. In each scene, over 4 objects with random arrangements are placed in a cluttered way. The sensor moves around the objects in an arbitrary trajectory. In total, we take 22,500 RGBD frames for STD-CatKnown, and 4,500 for STD-CatNovel.

Overall, the proposed real-world STD dataset consists of 27K RGBD frames, 30 diverse scenes, and 50 category-level and category-novel objects, making it facilitate the further generalizable object perception and grasping research.

5 Method

In this section, we introduce our network for depth restoration in section 5.1 and then introduce the methods we used for downstream tasks, *i.e.* category-level 6D object pose estimation and robotic grasping, in section 5.2.

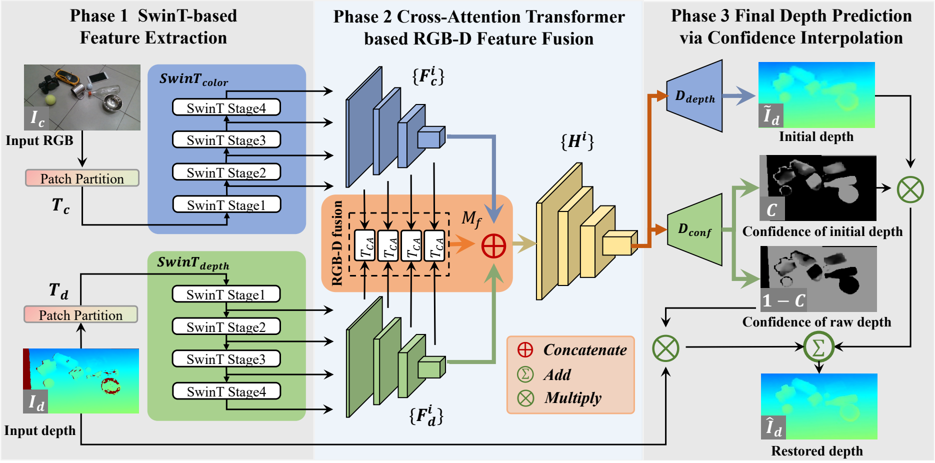

Fig. 4. Overview of our proposed depth restoration network SwinDRNet. We first extract the multi-scale features of RGB and depth image in phase 1, respectively. Next, in phase 2, our network fuse features of different modalities. Finally, we generate the initial depth map and confidence maps via two decoders, respectively, and fuse the raw depth and initial depth using the predicted confidence map.

5.1 SwinDRNet for Depth Restoration

Overview. To restore the noisy and incomplete depth, we propose a SwinTransformer [15] based depth restoration network, namely **SwinDRNet**.

SwinDRNet takes as input a RGB image $\mathcal{I}_c \in \mathbb{R}^{H\times W\times 3}$ along with its aligned depth image $\mathcal{I}_d \in \mathbb{R}^{H\times W}$ and outputs a refined depth $\hat{\mathcal{I}}_d \in \mathbb{R}^{H\times W}$ that restores the error area of the depth image and completes the invalid area, where H and W are the input image sizes.

We notice that prior works, *e.g.* PVN3D [9], usually leverage a heterogeneous architecture that extracts CNN features from RGB and extracts PointNet++ [24] features from depth. We, for the first time, devise a homogeneous and mirrored architecture that only leverages SwinT to extract and hierarchically fuse the RGB and depth features.

As shown in Figure 4, the architecture of SwinDRNet is a two-stream fused encoder-decoder and can be further divided into three phases: in the first phase of feature extraction, we leverage two separate SwinT backbones to extract hierarchical features $\{\mathcal{F}_c^i\}$ and $\{\mathcal{F}_d^i\}$ from the input RGB image $\mathcal{I}_c$ and depth $\mathcal{I}_d$, respectively; In the second stage of RGBD feature fusion, we propose a fusion module M_f that utilizes cross-attention transformers to combine the features from the two streams and generate fused hierarchical features $\{\mathcal{H}^i\}$; and finally in the third phase, we propose two decoder modules, the depth decoder module D_{depth} decodes the fused feature into a raw depth and the confidence decoder module D_{conf} outputs a confidence map of the predicted raw depth, and from the outputs we can compute the final restored depth by using the confidence map to select accurate depth predictions at noisy and invalid areas of the input depth while keeping the originally correct area as much as possible.

SwinT-based Feature Extraction. To accurately restore the noisy and incomplete depth, we need to leverage visual cues from the RGB image that helps depth completion as well as geometric cues from the depth that may save efforts at areas with correct input depths. To extract rich features, we propose to utilize SwinT [15] as our backbone, since it is a very powerful and efficient network that can produce hierarchical feature representations at different resolutions and has linear computational complexity with respect to input image size. Given our inputs contain two modalities – RGB and depth, we deploy two seperate SwinT networks, $SwinT_{\text{color}}$ and $SwinT_{\text{depth}}$, to extract features from $\mathcal{I}_c$ and $\mathcal{I}_d$, respectively. For each one of them, we basically follow the design of SwinT. Taking the $SwinT_{\text{color}}$ as an example: we first divide the input RGB image $\mathcal{I}_c \in \mathbb{R}^{H\times W\times 3}$ into non-overlapping patches, which is also called tokens, $\mathcal{T}_c \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times 48}$; we then pass $\mathcal{T}_c$ through the four stages of SwinT to generate the multi-scale features $\{\mathcal{F}_c^i\}$, which are especially useful for dense depth prediction thanks to the hierarchical structure. The encoder process can be formulated as:

$$\{\mathcal{F}_c^i\}_{i=1,2,3,4} = SwinT_{\text{color}}(\mathcal{T}_c), \tag{1}$$

$$\{\mathcal{F}_d^i\}_{i=1,2,3,4} = SwinT_{\text{depth}}(\mathcal{T}_d). \tag{2}$$

where $\mathcal{F}^i \in \mathbb{R}^{\frac{H}{4i}\times\frac{W}{4i}\times iC}$ and C is the output feature dimension of the linear embedding layer in the first stage of SwinT.

Cross-Attention Transformer based RGB-D Feature Fusion. Given the hierarchical features $\{\mathcal{F}_c^i\}$ and $\{\mathcal{F}_d^i\}$ from the two-stream SwinT backbone, our RGB-D fusion module M_f leverages cross-attention transformers to fuse the corresponding $\mathcal{F}_c^i$ and $\mathcal{F}_d^i$ into $\mathcal{H}^i$. For attending feature $\mathcal{F}_{\mathcal{A}}$ to $\mathcal{F}_{\mathcal{B}}$, a common cross-attention transformer T_{CA} first calculates the query vector Q_A from $\mathcal{F}_A$ and the key K_B and value V_B vectors from feature $\mathcal{F}_B$:

$$Q_A = \mathcal{F}_A \cdot W_q, \quad K_B = \mathcal{F}_B \cdot W_k, \quad V_B = \mathcal{F}_B \cdot W_v, \tag{3}$$

where Ws are the learnable parameters, and then computes the cross-attention feature $\mathcal{H}_{\mathcal{F}_A \to \mathcal{F}_B}$ from $\mathcal{F}_A$ to $\mathcal{F}_B$:

$$\mathcal{H}_{\mathcal{F}_A \to \mathcal{F}_B} = T_{CA}(\mathcal{F}_A, \mathcal{F}_B) = \text{softmax}\left(\frac{Q_A \cdot K_B^T}{\sqrt{d_K}}\right) \cdot V_B, \tag{4}$$

where d_K is the dimension of Q and K.

In our module M_f, we leverage bidirectional cross-attention by deploying two cross-attention transformers to obtained the cross-attention features from both directions, and then concatenates them with the original features to form the fused hierarchical features $\{\mathcal{H}^i\}$, as shown below:

$$\mathcal{H}^i = \mathcal{H}_{\mathcal{F}_c^i \to \mathcal{F}_d^i} \bigoplus \mathcal{H}_{\mathcal{F}_d^i \to \mathcal{F}_c^i} \bigoplus \mathcal{F}_c^i \bigoplus \mathcal{F}_d^i, \tag{5}$$

where $\bigoplus$ represents concatenation along the channel axis.

Final Depth Prediction via Confidence Interpolation. The credible area of the input depth map (*e.g.*, the edges of specular or transparent objects in contact with background or diffusive objects) plays a critical role in providing information about spatial arrangement. Inspired by the previous works [30,10], we make use of a confidence map between the raw and predicted depth maps. However, unlike [30,10] predicting the confidence map between the multi-modality, we focus on preserving the correct original value to generate more realistic depth maps with less distortion. The final depth map can be formulated as:

$$\hat{\mathcal{I}}_d = C \bigotimes \tilde{\mathcal{I}}_d + (1 - C) \bigotimes \mathcal{I}_d \tag{6}$$

where $\bigotimes$ represents elementwise multiplication, and $\hat{\mathcal{I}}_d$ and $\tilde{\mathcal{I}}_d$ denote the final restored depth and the output of depth decoder head, respectively.

Loss Functions For SwinDRNet training, we supervise both the final restored depth $\hat{\mathcal{I}}_d$ and the output of depth decoder head $\tilde{\mathcal{I}}_d$, which is formulated as:

$$\mathcal{L} = \omega_{\tilde{\mathcal{I}}_d}\mathcal{L}_{\tilde{\mathcal{I}}_d} + \omega_{\hat{\mathcal{I}}_d}\mathcal{L}_{\hat{\mathcal{I}}_d}, \tag{7}$$

where $\mathcal{L}_{\hat{\mathcal{I}}_d}$ and $\mathcal{L}_{\tilde{\mathcal{I}}_d}$ are the losses of $\hat{\mathcal{I}}_d$ and $\tilde{\mathcal{I}}_d$, respectively. $\omega_{\hat{\mathcal{I}}_d}$ and $\omega_{\tilde{\mathcal{I}}_d}$ are weighting factors. Each of the two loss can be formulated as:

$$\mathcal{L}_i = \omega_n\mathcal{L}_n + \omega_d\mathcal{L}_d + \omega_g\mathcal{L}_g, \tag{8}$$

where $\mathcal{L}_n$, $\mathcal{L}_d$ and $\mathcal{L}_g$ are the L1 losses between the predicted and ground truth surface normal, depth and the gradient map of depth image, respectively. ω_n, ω_d and ω_g are the weights for different losses. We further add higher weight to the loss within the foreground region, to push the network to concentrate more on the objects.

5.2 Downstream Tasks

Category-level 6D Object Pose Estimation. Inspired by [31], we use the same backbone with SwinDRNet, and add two decoder heads to predict coordinates of the NOCS map and semantic segmentation mask. Then we follow the method [31], perform pose fitting between the restored object point clouds in the world coordinate space and the predicted object point clouds in the normalized object coordinate space, and perform pose fitting to get the 6D object pose.

Robotic Grasping. By combining SwinDRNet to the object grasping task, we can analyze the performance of depth restoration on the robotic manipulation. We adopt the end-to-end network, GraspNet-baseline [8], to predict the 6-DoF grasping poses directly from the scene point cloud. Given the restored depth map from SwinDRNet, the scene point cloud is transformed and sent to GraspNet-baseline. Then the model predicts the grasp candidates. Finally, the gripper of the parallel-jaw robot arm executes the target rotation and position selected from those candidates.

6 Tasks, Benchmarks and Results

In this section, we train our SwinDRNet on the train split of DREDS-CatKnown dataset and deploy it on the tasks including category-level 6D object pose estimation and robotic grasping.

6.1 Depth Restoration

Evaluation Metrics. We follow the metrics of transparent objects depth completion in [38]: 1) **RMSE**: the root mean squared error, 2) **REL**: the mean absolute relative difference, 3) **MAE**: the mean absolute error, 4) the percentage of d_i satisfying $max(\frac{d_i}{d_i^*}, \frac{d_i^*}{d_i}) < \boldsymbol{\delta}$, where d_i denotes the predicted depth, d_i^* is GT and $\delta \in \{1.05, 1.10, 1.25\}$. We resize the prediction and GT to 126×224 resolution for fair comparisons, and evaluate in all objects area and challenging area (specular and transparent objects), respectively.

Baselines. We compare our method with several state-of-the-art methods, including LIDF [38], the SOTA method for depth completion of transparent objects, and NLSPN [19], the SOTA method for depth completion on NYUv2 [29] dataset. All baselines are trained on the train split of DREDS-CatKnown and evaluated on four types of testing data: 1) the test split of DREDS-CatKnown: simulated images of category-known objects. 2) DREDS-CatNovel: simulated images of category-novel objects. 3) STD-CatKnown: real images of category-known objects; 4) STD-CatNovel. real images of category-novel objects.

Results. The quantitative results reported in Table 2 show that we achieve the best performance compared to other methods on DREDS and STD datasets, and have a powerful generalization ability to transfer to not only novel category objects in the simulation environment but also in the real world. Moreover, SwinDRNet (30 FPS) is significantly faster than LIDF (13 FPS) and the two-branch baseline that uses PointNet++ on depth (6 FPS). Although it is a little slower than NLSPN (35 FPS) because the code still has room for further optimization and acceleration, SwinDRNet is real-time for downstream tasks. The methods are all evaluated on an NVIDIA RTX 3090 GPU.

Sim-to-Real and Domain Transfer. We perform sim-to-real and domain transfer experiments to verify the generalization ability of the DREDS dataset. For sim-to-real experiments, SwinDRNet is trained on DREDS-CatKnown, but takes different depth images as input of training (one follow [38] and takes the cropped synthetic depth image as input, and another takes the simulated depth image). The results evaluated on STD in Table 3 reveal the powerful potential of our depth simulation pipeline, which can significantly close the sim-to-real gap and generalize to the new categories. For domain transfer experiments, we train SwinDRNet on the train split of DREDS-CatKnown dataset and evaluate on Cleargrasp dataset. The results reported in Table 4 testify that model only trained on DREDS-CatKnown can easily generalize to the new domain ClaerGrasp and outperform the previous results directly trained on ClearGrasp and Omniverse [38] (LIDF train on Omniverse and ClearGrasp), which verifies the generalization ability of our dataset.

Table 2. Quantitative comparison to state-of-the-art methods on DREDS and STD. ↓ means lower is better, ↑ means higher is better. The left of '/' shows the results evaluated on all objects, and the right of '/' shows the results evaluated on specular and transparent objects. Note that only one result is reported on STD-CatNovel, because all the objects are specular or transparent.

Methods	RMSE↓	REL↓	MAE↓	$\delta_{1.05}$ ↑	$\delta_{1.10}$ ↑	$\delta_{1.25}$ ↑
	DREDS-CatKnown (Sim)					
NLSPN	**0.010**/0.011	0.009/0.011	0.006/0.007	97.48/96.41	99.51/99.12	99.97/99.74
LIDF	0.016/0.015	0.018/0.017	0.011/0.011	93.60/94.45	98.71/98.79	99.92/99.90
Ours	**0.010/0.010**	**0.008/0.009**	**0.005/0.006**	**98.04/97.76**	**99.62/99.57**	**99.98/99.97**
	DREDS-CatNovel (Sim)					
NLSPN	0.026/0.031	0.039/0.054	0.015/0.021	78.90/69.16	89.02/83.55	97.86/96.84
LIDF	0.082/0.082	0.183/0.184	0.069/0.069	23.70/23.69	42.77/42.88	75.44/75.54
Ours	**0.022/0.025**	**0.034/0.044**	**0.013/0.017**	**81.90/75.27**	**92.18/89.15**	**98.39/97.81**
	STD-CatKnown (Real)					
NLSPN	0.114/0.047	0.027/0.031	0.015/0.018	94.83/89.47	98.37/97.48	99.38/99.32
LIDF	0.019/0.022	0.019/0.023	0.013/0.015	93.08/90.32	98.39/97.38	99.83/99.62
Ours	**0.015/0.018**	**0.013/0.016**	**0.008/0.011**	**96.66/94.97**	**99.03/98.79**	**99.92/99.85**
	STD-CatNovel (Real)					
NLSPN	0.087	0.050	0.025	**81.95**	90.36	96.06
LIDF	0.041	0.060	0.031	53.69	79.80	99.63
Ours	**0.025**	**0.033**	**0.017**	81.55	**93.10**	**99.84**

Table 3. Quantitative results for Sim-to-Real. *Synthetic* means taking the cropped synthetic depth images for training, and *Simulated* means taking the simulated depth images from the train split of DREDS-CatKnown for training.

Trainset	RMSE↓	REL↓	MAE↓	$\delta_{1.05}$ ↑	$\delta_{1.10}$ ↑	$\delta_{1.25}$ ↑
	STD-CatKnown (Real)					
Synthetic	0.0467/0.056	0.0586/0.070	0.0377/0.047	49.12/39.42	86.50/79.85	98.98/97.66
Simulated	**0.015/0.018**	**0.013/0.016**	**0.008/0.011**	**96.66/94.97**	**99.03/98.79**	**99.92/99.85**
	STD-CatNovel (Real)					
Synthetic	0.065	0.101	0.053	21.04	55.87	96.96
Simulated	**0.025**	**0.033**	**0.017**	**81.55**	**93.10**	**99.84**

Table 4. Quantitative results for domain transfer. *The previous best results* means that the best previous method is trained on ClearGrasp and Omniverse, and evaluated on ClearGrasp. *Domain transfer* means that SwinDRNet is trained on DREDS-CatKnown and evaluated on ClearGrasp.

Model	RMSE↓	REL↓	MAE↓	$\delta_{1.05}$ ↑	$\delta_{1.10}$ ↑	$\delta_{1.25}$ ↑
	ClearGrasp real-known					
The previous best results	0.028	0.033	0.020	82.37	92.98	98.63
Domain transfer	**0.022**	**0.017**	**0.012**	**91.46**	**97.47**	**99.86**
	ClearGrasp real-novel					
The previous best results	0.025	0.036	0.020	79.5	94.01	99.35
Domain transfer	**0.016**	**0.008**	**0.005**	**96.73**	**98.83**	**99.78**

6.2 Category-level Pose Estimation

Evaluation Metrics. We use two aspects of metrics to evaluate: 1) **3D IoU.** It computes the intersection over union of ground truth and predicted 3D bounding boxes. We choose the threshold of 25% (IoU25), 50%(IoU50) and 75%(IoU75) for this metric. 2) **Rotation and translation errors.** It computes the rotation and translation errors between the ground truth pose and predicted pose. We choose 5°2cm, 5°5cm, 10°2cm, 10°5cm, 10°10cm for this metric.

Baselines. We choose two models as baselines to show the usefulness of the restored depth for category-level pose estimation and the effectiveness of SwinDRNet+NOCSHead: 1) **NOCS** [31]. It takes a RGB image as input to predict the per-pixel normalized coordinate map and obtain the pose with the help of the depth map. 2) **SGPA** [6]. The state-of-the-art method. It leverages one object and its corresponding category prior to dynamically adapting the prior to the observed object. Then the prior adaptation is used to reconstruct the 3D canonical model of the specific object for pose fitting.

Results. To verify the usefulness of the restored depth, we report the results of three methods using raw or restored (output of SwinDRNet) depth in Table 5. *-only* means using raw depth in the whole experiment, *Refined depth+* means using restored depth for pose fitting in NOCS and SwinDRNet+NOCSHead. Due to the fact that SGPA deforms the point cloud to get the results which are sensitive to depth, we use restored depth for both training and inference. We observe that restored depth improves the performance of three methods by large margins under all the metrics on both dataset. These performance gains suggest that depth restoration is truly useful for category-level pose estimation. Moreover, SwinDRNet+NOCSHead outperforms NOCS and SGPA under all the metrics.

Table 5. Quantitative results for category-level pose estimation. *only* means using raw depth in the whole experiment,*Refined* means using restored depth for training and inference in SGPA and for pose fitting in NOCS and our method.

Methods	IoU25	IoU50	IoU75	5°2cm	5°5cm	10°2cm	10°5cm	10°10cm
	DREDS-CatKnown (Sim)							
NOCS-only	85.7	66.0	23.0	21.3	25.4	40.0	47.9	49.0
SGPA-only	79.5	66.7	49.1	29.5	32.5	48.7	54.7	55.7
Refined depth + NOCS	86.7	73.2	40.7	30.4	31.8	54.1	57.5	57.6
Refined depth + SGPA	82.3	72.0	60.5	45.9	46.8	66.4	68.4	68.5
Ours-only	94.3	82.5	57.9	34.5	37.6	55.7	62.6	63.2
Refined depth + Ours	**94.7**	**84.8**	**68.0**	**49.1**	**50.1**	**69.8**	**72.4**	**72.5**
	STD-CatKnown (Real)							
NOCS-only	83.2	66.9	16.9	20.4	26.0	37.9	52.5	53.5
SGPA-only	77.6	67.1	46.6	30.0	32.3	47.7	53.3	53.9
Refined depth + NOCS	82.6	72.6	35.6	28.5	30.0	54.4	57.6	57.7
Refined depth + SGPA	78.8	71.6	62.8	49.3	49.7	70.5	71.5	71.6
Ours-only	**92.4**	87.4	61.7	37.9	42.6	57.8	70.6	71.0
Refined depth + Ours	**92.4**	**88.0**	**75.9**	**52.9**	**53.8**	**77.1**	**79.1**	**79.1**

6.3 Robotic Grasping

Experiments Setting. We conduct real robot experiments to evaluate the depth restoration performance on robotic grasping tasks. In our physical setup, we use a 7-DoF Panda robot arm from Franka Emika with a parallel-jaw gripper. RealSense D415 depth sensor is mounted on the tripod in front of the arm. We set 6 rounds of table clearing experiments. For each round, 4 to 5 specular and transparent objects are randomly picked from STD objects to construct a cluttered scene. For each trial, the robot arm executes the grasping pose with the highest score, and removes the grasped object until the workspace is cleared, or 10 attempts are reached.

Evaluation Metrics. Real grasping performance is measured using the following metrics: 1) **Success Rate**: the ratio of grasped object number and attempt number, 2) **Completion Rate**: the ratio of successfully removed object number and the original object number in a scene.

Baselines. We follow the 6-DoF grasping pose prediction network GraspNet-baseline, using the released pretrained model. *GraspNet* means GraspNet-baseline directly takes the captured raw depth as input, while *SwinDRNet+GraspNet* means the network receives the refined point cloud from SwinDRNet that is trained only on DREDS-CatKnown dataset.

Table 6. Results of real robot experiments. *#Objects* denotes the sum of grasped object numbers in all rounds. *#Attempts* denotes the sum of robotic grasping attempt numbers in all rounds.

Methods	#Objects	#Attempts	Success Rate	Completion Rate
GraspNet	19	49	38.78%	40%
SwinDRNet+GraspNet	25	26	**96.15%**	**100%**

Results. Table 6 reports the performance of real robot experiments. *SwinDRNet+GraspNet* obtains high success rate and completion rate, while *GraspNet* is lower. Without depth restoration, it is difficult for a robot arm to grasp specular and transparent objects due to the severely incomplete and inaccurate raw depth. The proposed SwinDRNet significantly improves the performance of specular and transparent object grasping.

7 Conclusions

In this work, we propose a powerful RGBD fusion network, SwinDRNet, for depth restoration. Our proposed framework, DREDS, synthesizes a large-scale RGBD dataset with realistic sensor noises, so as to close the sim-to-real gap for specular and transparent objects. Furthermore, we collect a real dataset STD, for real-world performance evaluation. Evaluations on depth restoration, category-level pose estimation, and object grasping tasks demonstrate the effectiveness of our method.

References

1. Blender. https://www.blender.org/
2. Object capture api on macos. https://developer.apple.com/augmented-reality/object-capture/
3. Breyer, M., Chung, J.J., Ott, L., Roland, S., Juan, N.: Volumetric grasping network: Real-time 6 dof grasp detection in clutter. In: Conference on Robot Learning (2020)
4. Calli, B., Singh, A., Bruce, J., Walsman, A., Konolige, K., Srinivasa, S., Abbeel, P., Dollar, A.M.: Yale-cmu-berkeley dataset for robotic manipulation research. The International Journal of Robotics Research **36**(3), 261–268 (2017)
5. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
6. Chen, K., Dou, Q.: Sgpa: Structure-guided prior adaptation for category-level 6d object pose estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2773–2782 (2021)
7. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. Advances in neural information processing systems **27** (2014)
8. Fang, H.S., Wang, C., Gou, M., Lu, C.: Graspnet-1billion: A large-scale benchmark for general object grasping. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11444–11453 (2020)
9. He, Y., Sun, W., Huang, H., Liu, J., Fan, H., Sun, J.: Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11632–11641 (2020)
10. Hu, M., Wang, S., Li, B., Ning, S., Fan, L., Gong, X.: Penet: Towards precise and efficient image guided depth completion. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 13656–13662. IEEE (2021)
11. Jiang, Z., Zhu, Y., Svetlik, M., Fang, K., Zhu, Y.: Synergies between affordance and geometry: 6-dof grasp detection via implicit representations. Robotics: science and systems (2021)
12. Jiao, J., Cao, Y., Song, Y., Lau, R.: Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In: Proceedings of the European conference on computer vision (ECCV). pp. 53–69 (2018)
13. Khirodkar, R., Yoo, D., Kitani, K.: Domain randomization for scene-specific car detection and pose estimation. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1932–1940. IEEE (2019)
14. Landau, M.J., Choo, B.Y., Beling, P.A.: Simulating kinect infrared and depth images. IEEE transactions on cybernetics **46**(12), 3018–3031 (2015)
15. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)
16. Long, X., Lin, C., Liu, L., Li, W., Theobalt, C., Yang, R., Wang, W.: Adaptive surface normal constraint for depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12849–12858 (2021)
17. Mo, K., Guibas, L.J., Mukadam, M., Gupta, A., Tulsiani, S.: Where2act: From pixels to actions for articulated 3d objects. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6813–6823 (2021)

18. Mu, T., Ling, Z., Xiang, F., Yang, D., Li, X., Tao, S., Huang, Z., Jia, Z., Su, H.: ManiSkill: Generalizable Manipulation Skill Benchmark with Large-Scale Demonstrations. In: Annual Conference on Neural Information Processing Systems (NeurIPS) (2021)
19. Park, J., Joo, K., Hu, Z., Liu, C.K., So Kweon, I.: Non-local spatial propagation network for depth completion. In: European Conference on Computer Vision. pp. 120–136. Springer (2020)
20. Peng, X.B., Andrychowicz, M., Zaremba, W., Abbeel, P.: Sim-to-real transfer of robotic control with dynamics randomization. In: 2018 IEEE international conference on robotics and automation (ICRA). pp. 3803–3810. IEEE (2018)
21. Planche, B., Singh, R.V.: Physics-based differentiable depth sensor simulation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14387–14397 (2021)
22. Planche, B., Wu, Z., Ma, K., Sun, S., Kluckner, S., Lehmann, O., Chen, T., Hutter, A., Zakharov, S., Kosch, H., et al.: Depthsynth: Real-time realistic synthetic data generation from cad models for 2.5 d recognition. In: 2017 International Conference on 3D Vision (3DV). pp. 1–10. IEEE (2017)
23. Prakash, A., Boochoon, S., Brophy, M., Acuna, D., Cameracci, E., State, G., Shapira, O., Birchfield, S.: Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 7249–7255. IEEE (2019)
24. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems **30** (2017)
25. Qu, C., Liu, W., Taylor, C.J.: Bayesian deep basis fitting for depth completion with uncertainty. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16147–16157 (2021)
26. Sajjan, S., Moore, M., Pan, M., Nagaraja, G., Lee, J., Zeng, A., Song, S.: Clear grasp: 3d shape estimation of transparent objects for manipulation. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 3634–3642. IEEE (2020)
27. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain randomization for transferring deep neural networks from simulation to the real world. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). pp. 23–30. IEEE (2017)
28. Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S., Birchfield, S.: Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 969–977 (2018)
29. Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., Geiger, A.: Sparsity invariant cnns. In: 2017 international conference on 3D Vision (3DV). pp. 11–20. IEEE (2017)
30. Van Gansbeke, W., Neven, D., De Brabandere, B., Van Gool, L.: Sparse and noisy lidar completion with rgb guidance and uncertainty. In: 2019 16th international conference on machine vision applications (MVA). pp. 1–6. IEEE (2019)
31. Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.J.: Normalized object coordinate space for category-level 6d object pose and size estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2642–2651 (2019)

32. Weng, Y., Wang, H., Zhou, Q., Qin, Y., Duan, Y., Fan, Q., Chen, B., Su, H., Guibas, L.J.: Captra: Category-level pose tracking for rigid and articulated objects from point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13209–13218 (2021)
33. Xiong, X., Xiong, H., Xian, K., Zhao, C., Cao, Z., Li, X.: Sparse-to-dense depth completion revisited: Sampling strategy and graph construction. In: European Conference on Computer Vision. pp. 682–699. Springer (2020)
34. Xu, H., Wang, Y.R., Eppel, S., Aspuru-Guzik, A., Shkurti, F., Garg, A.: Seeing glass: Joint point-cloud and depth completion for transparent objects. In: 5th Annual Conference on Robot Learning (2021)
35. Yue, X., Zhang, Y., Zhao, S., Sangiovanni-Vincentelli, A., Keutzer, K., Gong, B.: Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2100–2110 (2019)
36. Zakharov, S., Kehl, W., Ilic, S.: Deceptionnet: Network-driven domain randomization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 532–541 (2019)
37. Zhang, X., Chen, R., Xiang, F., Qin, Y., Gu, J., Ling, Z., Liu, M., Zeng, P., Han, S., Huang, Z., et al.: Close the visual domain gap by physics-grounded active stereovision depth sensor simulation. arXiv preprint arXiv:2201.11924 (2022)
38. Zhu, L., Mousavian, A., Xiang, Y., Mazhar, H., van Eenbergen, J., Debnath, S., Fox, D.: Rgb-d local implicit function for depth completion of transparent objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4649–4658 (2021)