

TransCG: A Large-Scale Real-World Dataset for Transparent Object Depth Completion and A Grasping Baseline

Hongjie Fang¹, Hao-Shu Fang¹, Sheng Xu¹ and Cewu Lu²

Abstract—Transparent objects are common in our daily life and frequently handled in the automated production line. Robust vision-based robotic grasping and manipulation for these objects would be beneficial for automation. However, the majority of current grasping algorithms would fail in this case since they heavily rely on the depth image, while ordinary depth sensors usually fail to produce accurate depth information for transparent objects owing to the reflection and refraction of light. In this work, we address this issue by contributing a large-scale real-world dataset for transparent object depth completion, which contains 57,715 RGB-D images from 130 different scenes. Our dataset is the first large-scale, real-world dataset that provides ground truth depth, surface normals, transparent masks in diverse and cluttered scenes. Cross-domain experiments show that our dataset is more general and can enable better generalization ability for models. Moreover, we propose an end-to-end depth completion network, which takes the RGB image and the inaccurate depth map as inputs and outputs a refined depth map. Experiments demonstrate superior efficacy, efficiency and robustness of our method over previous works, and it is able to process images of high resolutions under limited hardware resources. Real robot experiments show that our method can also be applied to novel transparent object grasping robustly. The full dataset and our method are publicly available at www.graspnet.net/transcg.

Index Terms—Perception for Grasping and Manipulation, Data Sets for Robotic Vision, Deep Learning in Grasping and Manipulation

I. INTRODUCTION

Transparent materials are widely used in modern industry, and robots inevitably need to process transparent objects no matter in manufacturing, logistics, or household services. Recently, much progress has been made in the field of robot grasping and manipulation [7, 23]. However, many of the advances are not directly applicable in scenes with transparent objects since these methods heavily rely on the depth information collected by the RGB-D cameras, yet ordinary depth sensors usually fail to construct a complete depth image in scenes that include transparent objects. The physical properties of transparent objects would lead to the distortion of light

Manuscript received: February, 21, 2022; Revised May, 6, 2022; Accepted June, 1, 2022.

This paper was recommended for publication by Editor Markus Vincze upon evaluation of the Associate Editor and Reviewers' comments.

¹Hongjie Fang, Hao-Shu Fang and Sheng Xu are with the Department of Computer Science, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: galaxies@sjtu.edu.cn; fhaoshu@gmail.com; xs1020@sjtu.edu.cn).

²Cewu Lu is the corresponding author, member of Qing Yuan Research Institute and MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: lucewu@sjtu.edu.cn).

Digital Object Identifier (DOI): see top of this page.

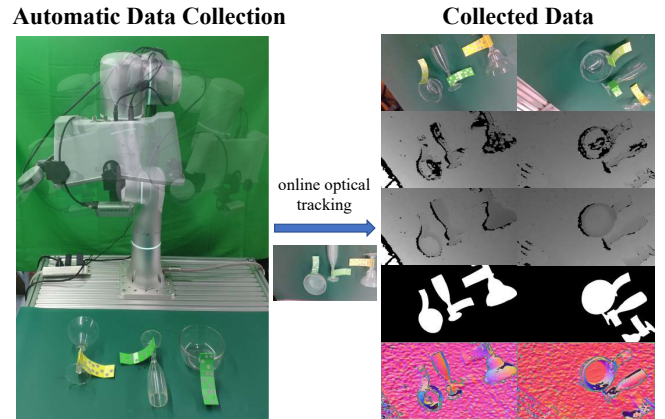


Fig. 1. The methodology for building our dataset. We first perform object-level annotation to enable the real-time object pose tracking for our collecting system. With the assistance of online optical tracking, we are able to collect and annotate data automatically. Our dataset consists of RGB images, raw depth information collected by the depth sensor of RGB-D camera, the generated ground-truth depth images, transparent ground-truth mask and the surface normal from top to bottom. Our methodology can significantly reduce the workload of manual annotation and generate a large amount of high-quality real-world data.

path by reflection and refraction, resulting in noisy depth maps. Therefore, many depth-based algorithms are incapable to handle transparent objects such as plastic bottles and glass containers which can be found everywhere in our daily life.

The geometry estimation of transparent objects remains a challenging task in the computer vision field, though progress has been made by many researchers. Ba *et al.* [2] utilized a special polarization camera to leverage polarization cues for shape estimation and reached satisfactory results, while Li *et al.* [17] proposed a two-stage physical-based network to reconstruct the shape of the transparent objects using multi-view images and material prior. However, both methods require specialized hardware, which is not a general setting for robotic manipulation. A more common setting is a robot arm with an RGB-D camera, which is the setting that we mainly focus on.

Under this circumstance, Sajjan *et al.* [31] adapt the depth completion pipeline [40] to scenes that contain transparent objects and then propose ClearGrasp, which predicts the surface normal and the transparent boundary, followed by the global optimization to solve the depth estimation. A synthetic dataset and a small real-world dataset are also proposed along with the method. Zhu *et al.* [41] present an end-to-end framework for depth completion using the local implicit depth function, along

with a synthetic Omniverse Object dataset. Both synthetic datasets provide images containing transparent objects and their ground-truth depth maps, but the lack of real depth maps degrades the performance of those methods [31, 41] in real-world applications inevitably.

To close the synthetic-to-real gap in the field of grasping concerning transparent objects, we propose **TransCG**, a large-scale real-world dataset for transparent object depth completion. A novel semi-automatic pipeline is proposed to accelerate the data collection and annotation process. In total, our dataset contains **57,715 RGB-D images of 51 transparent objects** and around 200 opaque objects captured from different perspectives of **130 scenes** under real-world settings. The 3D mesh model of the transparent objects are also provided in our dataset. The methodology for building our dataset is shown in Fig. 1.

Furthermore, we propose a robust, efficient and effective network *Depth Filler Net (DFNet)* for depth completion based on our dataset, which allows it to assist 6-DoF grasping methods on transparent object grasping. The quantitative and qualitative results show that our proposed DFNet (1) generalizes better across different scenes compared to existing methods; (2) improves most of the metrics to a great extent; (3) yields the highest inference speed and consumes the least computation overhead. We also apply our network to real-world object grasping for novel transparent objects, and a promising performance is witnessed. The full dataset, source code and pretrained models are released at www.graspnet.net/transcg.

II. RELATED WORKS

A. Transparent Datasets

Due to the special optical properties of transparent objects which leads to undetermined and inaccurate results of optical sensors, the datasets concerning of transparent objects is usually difficult to build.

For tasks that do not need the depth information, *e.g.*, transparent object classification, segmentation and pose estimation, there are large-scale datasets such as Trans10K-V2 [37], Stanford2D-3D [1] and StereOBJ-1M [19]. For datasets that needs accurate sensor information as ground-truth, the common solutions are synthetic datasets, such as ClearGrasp synthetic dataset [31] and Omniverse object dataset [41]. Built by tools like SuperCaustics [24], the greatest shortcoming to those datasets is that the raw information collected by sensors is unlikely to be obtained in simulation. Xu *et al.* [38] propose Toronto Transparent Object Depth Dataset (TODD), another real-world depth completion dataset. However, the diversities of transparent objects, translucent objects and cameras in the dataset are limited. Liu *et al.* [20] build a real-world keypoint estimation dataset for transparent objects, which consists of the ground-truth depth map generated by substituting the transparent object with the identical opaque objects. But the large amount of real-world samples are all captured alone under certain environments, which is rare in real-world applications. The dataset collection approach is also used in ClearGrasp real-world dataset [31] which only has 286 samples since generating the missing information is both time-consuming and labor-consuming.

To overcome the difficulties of building datasets concerning transparent objects, we propose a novel pipeline for transparent dataset construction. Using the pipeline, we build TransCG, a complete large-scale real-world dataset for transparent object depth completion. Details will be introduced in Section III.

B. Depth Completion and Estimation

Depth completion and estimation has been studied by many researchers for a long time, and can be categorized into three classes: estimating depth directly from an RGB image [6, 10, 15, 39], estimating depth from an RGB image with sparse depth information [4, 5, 9, 16, 22] and depth completion from an RGB image with inaccurate depth information [31, 40, 38, 41]. Since depth information concerning reflective objects is usually noisy and inaccurate, transparent object depth completion falls into the last category, and it can be further classified into multi-view depth completion and single-view depth completion.

For multi-view depth completion, Ichnowski *et al.* [13] use NeRF to recover depth for grasping transparent objects. Li *et al.* [17] present a physical-based network, which uses multi-view images to recover high-quality 3D geometry of transparent objects.

For single-view depth completion, Zhang *et al.* [40] propose a two-stage depth completion pipeline, which predicts surface normals and occlusion boundaries according to the RGB image, followed by the global optimization to complete the depth map. ClearGrasp [31] makes a few critical modifications to the pipeline and adapts it in depth completion concerning transparent object, but its inference speed is unacceptable in real-time grasping scenarios. Tang *et al.* [34] utilize self-attentive adversarial network to replace the global optimization in ClearGrasp. Zhu *et al.* [41] propose a two-stage system consisting of local implicit depth function prediction and depth refinement. Though it outperforms previous works on speed and accuracy, its generalization ability is very limited according to our cross-domain tests in Sec. V-B, which makes it difficult to fit into real-world robotic manipulation settings. A concurrent work [38] combines previous depth completion method [32] with point cloud completion to improve the quality of the refined depth.

To achieve a better generalization and applicability in real-world environment, a tiny but robust model is needed, which requires a large amount of real-world data as support.

C. 6-DoF Grasping

6-DoF grasping refers to those methods that predict the position and rotation of the gripper in 3D domain. Enabling the robots to grasp objects from various angles, it is the basis of robotic manipulation. Due to the versatility and effectiveness of 6-DoF grasping, it is currently the main stream in the grasping field.

GPD [27] propose a two-stage 6-DoF grasping method, which estimates grasp candidates sampled under empirical constraints. PointNetGPD [18] improves GPD by adapting PointNet [28] in evaluation. Mousavian *et al.* [25] leverage

TABLE I
COMPARISONS OF TRANSPARENT DEPTH COMPLETION DATASETS

Type	Dataset	Dataset Completeness			Scene Type			Auto-Collection	#Cam.	#Obj.	#Img.
		RGB	Raw depth	Refined depth	Single	Isolated	Cluttered				
Syn	Clear-Syn [31]	✓		✓	✓	✓		✓	-	9	50K
	OOD [41]	✓		✓	✓	✓	✓	✓	-	9	60K
Real	Clear-Real [31]	✓	✓	✓	✓	✓			2	10	286
	TOD [20]	✓	✓	✓	✓	✓			1	15	48K
	TODD [38]	✓	✓	✓	✓	✓		✓	1	6	15K
	TransCG (ours)	✓	✓	✓	✓	✓	✓	✓	2	51	58K

Note. Clear refers to ClearGrasp Dataset (Synthetic/Real-world), OOD refers to Omniverse Object Dataset, TOD refers to Transparent Object Dataset. “#Cam.” denotes types of RGB-D cameras, “#Obj.” stands for the number of target objects and “#Img.” represents the amount of samples. Here cluttered scenes refer to scenes that consists of more than 3 transparent objects and several opaque objects.

variational auto-encoder to sample grasp poses, and add refinement process after evaluation for better performance. Qin *et al.* [30] regress the grasp pose directly from the partial-view point cloud, while Ni *et al.* [26] regress the grasp pose from features extracted by PointNet++ [29]. Fang *et al.* [7] propose the GraspNet-1Billion dataset for general object grasping and an end-to-end grasp pose prediction network. Gou *et al.* [11] incorporate RGB and depth information to improve the performance of 6-DoF grasping. Sundermeyer *et al.* [33] reduce 6-DoF grasp poses to 4-DoF grasp representations on a known contact point to facilitate the learning problems and improve the grasping quality in cluttered scenes. Wang *et al.* [35] propose a geometrically-based quality grasping to evaluate the graspable area in cluttered scenes.

All these methods rely heavily on depth image, which makes them unsuitable for transparent object grasping. Thus, utilizing color information to generate high-quality depth map and point cloud to aid the grasping deserves further exploration.

III. DATASET

A. Overview

As introduced before, the previous transparent datasets that need accurate ground-truth depth usually use simulation platforms to generate synthetic data, and real-world datasets are usually small in scale due to the workload of annotations. To overcome the difficulties, we propose a novel pipeline to build the dataset efficiently, which utilizes a robot to perform data collection after limited object-level manual annotations.

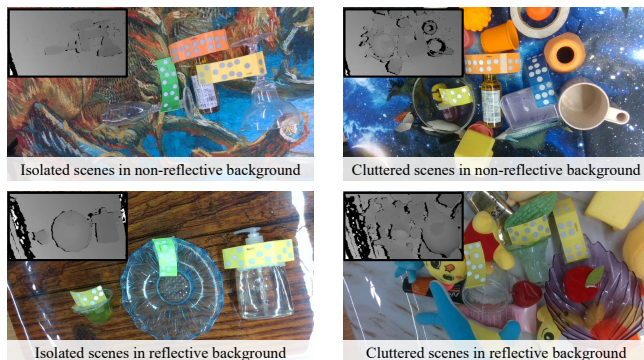


Fig. 2. Different types of scenes in our TransCG dataset. The sub-figures on the top left are the refined ground-truth depth images.

In brief, we aim to collect a dataset that contains RGBD images using real-world sensors, along with detail annotations for the transparent objects include their depths, masks, 6D poses, normals, etc. To reduce the annotation effort, our methodology for building the dataset is to automatically localize the transparent objects during data collection. To achieve that, we resort to an optical tracker that can accurately localize a target’s 6D pose in real-time from several IR markers attached to it. Besides, we manage to obtain the 3D model of each transparent object in the training set by wrapping them with opaque materials and scanning with a 3D scanner. With these two preliminaries, we can easily obtain the transparent objects’ 6D pose during data collection. The geometry of transparent objects can be restored by leveraging the tracker results and their 3D models during data annotation. Our pipeline is able to build large-scale real-world dataset conveniently and reduce the human workload by a large extent.

Using the pipeline, we build our TransCG dataset, which contains 57,715 RGB-D images captured by two different cameras, along with the refined ground-truth depth images, transparent ground-truth mask and the surface normals, from 130 scenes under various background settings, within a week. We collect 51 common objects in daily life that can lead to inaccurate depth map, including transparent objects, translucent objects, reflective objects and objects with dense tiny holes. Apart from 65 simple isolated scenes that are similar to the scenes in the previous datasets, we also provide 65 challenging cluttered scenes that are closer to the real-world grasping environment, as shown in Fig. 2. More details are presented in supplementary video. The comparisons between our dataset and existing datasets are summarised in Tab. I.

B. System Setup

To support fast and accurate data collection process, we build a transparent object tracking system, which is the only part that requires human efforts in our dataset building pipeline. The system setup process is illustrated in Fig. 3.

Given a transparent object, we firstly attach a fixer with IR markers to it, as shown in Fig. 3b. A PST optical tracker¹ is used to record the IR markers and tracks them afterwards. Although we can directly attach IR markers to the object, we found that using a flat fixer can make the tracking more robust. Next, we temporarily wrap the transparent object with some

¹<https://www.ps-tech.com/products-pst-base/>

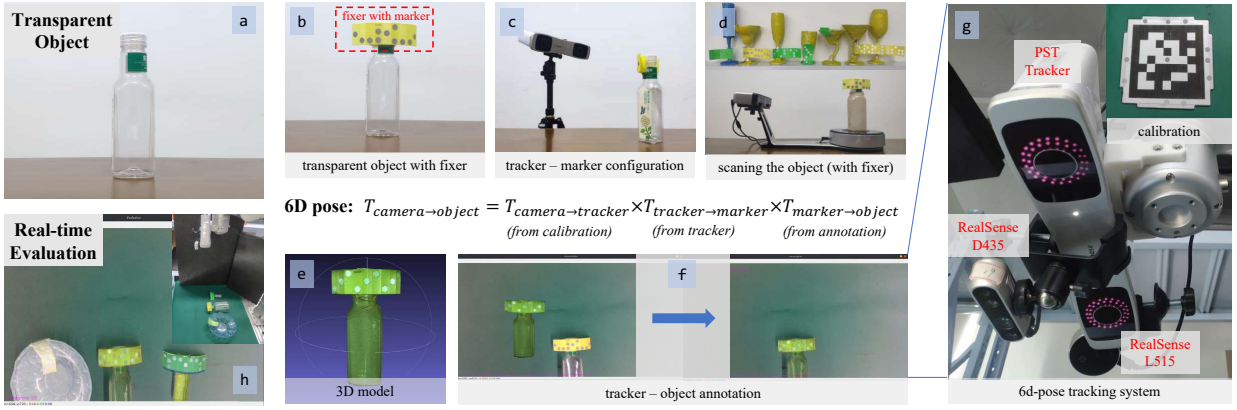


Fig. 3. System setup process. Given a transparent object (a), we attach a fixer with IR markers to it (b) and record its pattern with an optical tracker (c), which can enable tracking afterwards. Then we scan the object (d) and get its 3D model (e). After that, we manually perform tracker-object annotation to get the transformation matrix from marker to object (f), where a GUI is developed for real-time evaluation (h). The whole annotation and tracking process is assisted by our 6D pose tracking system (g), which consists of a PST tracker, an Intel RealSense D435 camera and an Intel RealSense L515 camera.

opaque materials that can preserve the shape of objects, and obtain its 3D model with a Shining3D EinScan-SP scanner². With these two steps, we can obtain the transparent object’s 6D pose during data collection, where the details is further explained.

The core of our data collection system consists of a PST optical tracker, an Intel RealSense D435 camera and an Intel RealSense L515 camera. The tracker outputs the 6D pose of the markers in real-time and the two cameras provide RGBD images with different quality. The 6D pose of the i -th object *w.r.t* the j -th camera $\mathbf{T}_{\text{cam}_j}^{\text{obj}_i}$ can be calculated as follows:

$$\mathbf{T}_{\text{cam}_j}^{\text{obj}_i} = \mathbf{T}_{\text{cam}_j}^{\text{tracker}} \mathbf{T}_{\text{tracker}}^{\text{marker}_i} \mathbf{T}_{\text{marker}_i}^{\text{obj}_i}, \quad (1)$$

where $\mathbf{T}_{\text{cam}_j}^{\text{tracker}}$ denotes the transformation matrix of the tracker origin *w.r.t* the j -th camera, $\mathbf{T}_{\text{tracker}}^{\text{marker}_i}$ denotes the 6D pose of the markers attached on the i -th object *w.r.t* the tracker, and $\mathbf{T}_{\text{marker}_i}^{\text{obj}_i}$ denotes the transformation matrix of the i -th object’s origin *w.r.t* the markers.

To obtain $\mathbf{T}_{\text{cam}_j}^{\text{tracker}}$, we perform tracker-camera calibration using a calibration board with both Aruco marker and IR markers. $\mathbf{T}_{\text{tracker}}^{\text{marker}_i}$ is the output of the optical tracker, and $\mathbf{T}_{\text{marker}_i}^{\text{obj}_i}$ are annotated by human. We develop a GUI application for the annotation process and evaluate the results in real time by rendering the objects in the corresponding RGB image.

On average, the overall human efforts to process an object is around 1 hour. With the transparent object tracking system we introduced, we can easily recover the ground-truth depth information of the transparent objects afterwards.

C. Data Collection

To collect a large amount of data automatically, we attach our tracking system to a robot arm that moves along a fixed trajectory containing 240 distinct viewpoints. Transparent objects are randomly selected and placed in the scene. To align with real-world setting, we augment the scene with various table covers and opaque objects. At each viewpoint, the tracking system will capture the RGBD images and the tracking results.

Before data collection, we perform camera extrinsic calibration for the 240 viewpoints, which can assist to recover the 6D pose of objects in case of tracker failure. ArUco markers are used during camera calibration to obtain the pose of the j -th camera $\mathbf{T}_k^{\text{cam}_j}$ *w.r.t* the marker coordinate in the k -th viewpoint for all k . Specifically, the 6D pose of the i -th object *w.r.t* the j -th camera in the perspective k can be obtained from the corresponding 6D pose in perspective k_0 and the camera poses in viewpoints k, k_0 , where k_0 is the successfully-tracked viewpoint:

$$\left(\mathbf{T}_{\text{cam}_j}^{\text{obj}_i} \right)_k = \mathbf{T}_k^{\text{cam}_j} \left(\mathbf{T}_{k_0}^{\text{cam}_j} \right)^{-1} \left(\mathbf{T}_{\text{cam}_j}^{\text{obj}_i} \right)_{k_0}. \quad (2)$$

Thus, we can recover the 6D pose of the objects that are not tracked in some viewpoints using the tracking results in the first viewpoint, which are always successfully detected.

After collecting the raw data, we use the collected 6D poses to render the ground-truth depth maps, the transparent masks and ground-truth surface normals.

D. Dataset Postprocessing

1) *Data Verification*: The blurry samples and improperly exposed samples are automatically detected and removed from the dataset using tools like Laplacian operators and histograms provided in OpenCV library [3]. Manual validations are conducted by rendering the object mesh to the scene using the 6D pose and see if they can overlap with the original object. Finally, we generate a metadata containing all valid viewpoints for each scene.

2) *Dataset Split*: We randomly select 12 objects from different categories and regard all scenes that contain these objects as the testing set, while the other scenes are used as the training set. There are totally 34,191 training samples and 23,524 testing samples.

E. Discussion

To ensure that the markers have minimal influence to the object, we do not directly place them on the surface of transparent objects. Instead they are placed on the surface

²<https://www.einscan.com/desktop-3d-scanners/einscan-sp/>

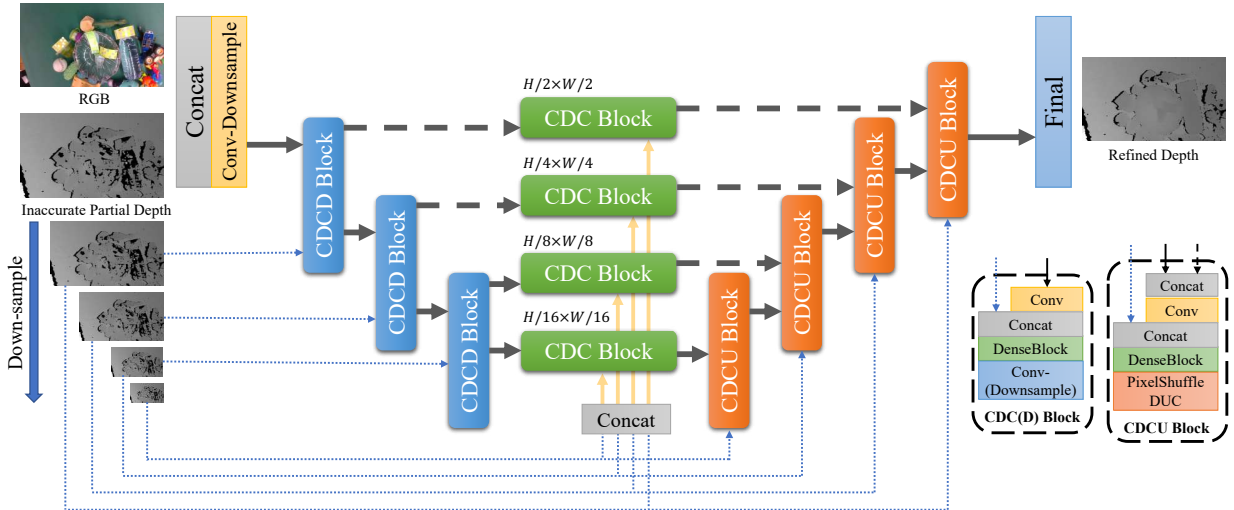


Fig. 4. The architecture of our proposed end-to-end depth completion network DFNet. Our network utilizes a U-Net architecture with CDCD blocks, CDC blocks and CDCU blocks. These blocks are mainly composed of dense blocks [12], with DUC [36] replacing deconvolution layer in up-sampling of CDCU block. All convolution layers except the last one are followed by batch normalizations [14] and ReLU activations, and have 3×3 kernels.

of a fixer that is attached to the object. Therefore, we can make sure that the transparent part of the object is retained to the greatest extent. From the cross-dataset experiment conducted in Table III and Table IV, we can see that methods trained on our dataset generalize well to other testing datasets. These experiments demonstrate that the markers have minimal impact on the training of network.

IV. METHOD

A. Overview

In this section, we detail our method for depth completion and grasping. For depth completion, we propose an end-to-end network which is illustrated in Fig. 4. Given an RGB image $\mathcal{C} \in \mathbb{R}^{H \times W \times 3}$ and an inaccurate partial depth image $\mathcal{D} \in \mathbb{R}^{H \times W}$, our network completes the depth information and predicts the full depth map $\hat{\mathcal{D}} \in \mathbb{R}^{H \times W}$, where $H \times W$ is the size of an image. The details of the network will be introduced in Sec. IV-B. In Sec. IV-C, we demonstrate how our depth completion network can be applied to the grasping task and serve as a grasping baseline method. We apply our depth completion to a grasp pose detection network [7] which takes point cloud as inputs and outputs the grasp poses. We demonstrate that our network can generate high quality depth for transparent objects that can enable depth-based grasping.

B. Depth Completion

Inspired by previous literature [4] about depth estimation from sparse sensing, we propose our end-to-end depth completion network *Depth Filler Net* (**DFNet**) that predicts full depth map according to RGB information and inaccurate partial depth. Adapting a U-Net architecture with depth of four layers, our network takes dense blocks [12] as backbones and formulates *Conv-Dense-Conv-Downsample* (CDCD) blocks, *Conv-Dense-Conv* (CDC) blocks and *Conv-Dense-Conv-Upsample* (CDCU) blocks. Empirical statistics about depth estimations and completions show that original depth information is critical throughout the networks. Hence we provide the original

depth information as an input to every CDCD, CDC and CDCU block. Skip paths are added to retain information in high resolutions. Inspired by AlphaPose [8], we adapt dense up-sampling convolution (DUC) [36] instead of ordinary deconvolution layers in CDCU blocks.

Our network is trained using the following loss function:

$$\mathcal{L} = \mathcal{L}_d + \beta \mathcal{L}_s, \quad (3)$$

where β is the weight parameter, \mathcal{L}_d penalizes depth inaccuracy and \mathcal{L}_s is the cosine distance of surface normals computed from predicted depth map and ground-truth depth map [41], which penalizes unsmoothness. Formally,

$$\begin{aligned} \mathcal{L}_d &= \left\| \hat{\mathcal{D}} - \mathcal{D}^* \right\|^2, \\ \mathcal{L}_s &= 1 - \cos \left\langle \hat{\mathcal{D}}_h \times \hat{\mathcal{D}}_w, \mathcal{D}_h^* \times \mathcal{D}_w^* \right\rangle, \end{aligned} \quad (4)$$

where $\hat{\mathcal{D}}$ and \mathcal{D}^* denotes the predicted depth and the ground-truth depth, \mathcal{D}_w and \mathcal{D}_h are gradient vectors along width-axis and height-axis of depth map \mathcal{D} respectively. For both losses, we regard depths out of range $[0.3, 1.5]$ as invalid pixels and remove them from losses to reduce the impact of outliers.

C. Object Grasping

Given an RGB image along with a depth image collected by an RGB-D camera, we first scale the images to an appropriate size and feed into our depth completion model **DFNet**, which outputs the refined depth in the same resolution as the input. Then, the refined depth is scaled back to the original size, which can be used to construct the scene point cloud using camera intrinsics. After that, the scene point cloud is sent to GraspNet-baseline [7] as the input to the end-to-end grasp pose detection network. Other depth based grasping methods are also applicable. Finally, the grasp pose detection network outputs the grasp candidates, and the grasp will be executed by a parallel-jaw robot.

TABLE II
EXPERIMENT RESULTS ON TRANSCG DATASET

Methods	Metrics						GPU Memory Occupation	Inference Time	Model Size
	RMSE ↓	REL ↓	MAE ↓	$\delta_{1.05} \uparrow$	$\delta_{1.10} \uparrow$	$\delta_{1.25} \uparrow$			
ClearGrasp [31]	0.054	0.083	0.037	50.48	68.68	95.28	2.1 GB	2.2813s	934 MB
LIDF-Refine [41]	0.019	0.034	0.015	78.22	94.26	99.80	6.2 GB	0.0182s	251 MB
TranspareNet [38]	0.026	0.023	0.013	88.45	96.25	99.42	1.9 GB	0.0354s	336 MB
DFNet (ours)	0.018	0.027	0.012	83.76	95.67	99.71	1.6 GB	0.0166s	5.2 MB

Note. ↓ means lower is better, and ↑ means higher is better. GPU Memory occupation and inference time are measured with an NVIDIA GeForce RTX 3090 GPU.

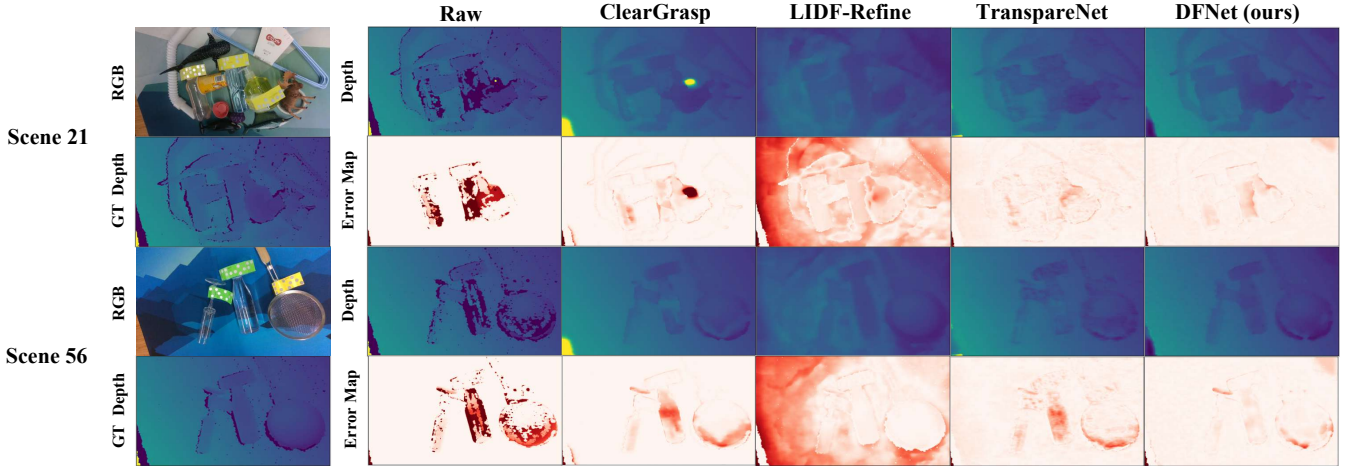


Fig. 5. Visualizations of the refined depth maps and error maps of different methods on several testing scenes of TransCG dataset.

V. EXPERIMENTS

A. Depth Completion Experiments

We compare our method with several representative approaches on our TransCG dataset. ClearGrasp [31] is the first algorithm which leverages deep learning with synthetic training data to estimate depth information concerning transparent objects, and LIDF-Refine [41] utilizes local implicit depth function to solve transparent object depth completion task. We also evaluate the concurrent work TranspareNet [38] in experiments. All baselines are trained in our TransCG dataset using their released source codes and optimal hyperparameters for fair comparisons.

For our model, the hidden channels in the network is set to 64. In every dense block, the layers L and the feature channels of each layer k are set to 5, 12 respectively, as suggested in literature [4]. We use AdamW optimizer [21] with initial learning rate of 10^{-3} and multi-step learning rate scheduler which decays the learning rate by 5 after 5, 15, 25, 35 epochs. We train the model for 40 epochs with the batch size of 32. Several data augmentation approaches such as random flipping, rotation, noise adding and chromatic transformations in HLS color space are conducted during training. Concerning loss, we set $\beta = 0.001$.

For all methods, we scale the images to 320×240 during training and testing. We use 4 NVIDIA GeForce RTX 3090 GPUs for training and one for testing, and the average time to train an epoch is approximately 20 minutes.

The following common metrics of depth completion for transparent objects are used in comparisons. All metrics are calculated on the transparent areas according to transparent masks unless specified.

- RMSE: the rooted mean squared error between depth estimates and ground-truth depths.
- REL: the mean absolute relative difference.
- MAE: the mean absolute error between depth estimates and ground-truth depths.
- Threshold δ : the percentage of pixels with predicted depths satisfying $\max(d/d^*, d^*/d) < \delta$, where d, d^* are corresponding pixels of $\mathcal{D}, \mathcal{D}^*$, and δ is set to 1.05, 1.10 and 1.25 following [31, 41].

The quantitative results are reported in Tab. II, and the qualitative results are visualized in Fig. 5. Our method outperforms ClearGrasp [31] and LIDF-Refine [41] on both metrics and quality, and performs better than the concurrent work TranspareNet [38] on some metrics. Qualitatively, it completes the inaccurate region of depth maps more completely (Fig. 5) than [38]. It is also worth noticing that though LIDF-Refine [41] performs well on masked metrics, it may introduce shadows on areas without transparent objects, which may result from the outliers of the original depth map. Moreover, our method has the smallest size, fastest inference time and lowest GPU memory occupation, which allows it to perform depth completion of high quality under limited resources (Tab. II).

B. Cross-Domain Experiments

Cross-domain experiments are performed to verify the robustness of our proposed depth completion method and the generalization ability of our proposed TransCG dataset.

For cross-domain experiments of different methods, two experiments are conducted, namely (1) test the performance on our TransCG dataset after training on ClearGrasp synthetic

TABLE III
RESULTS OF CROSS-DOMAIN EXPERIMENTS OF METHODS

Training/ Testing	Methods	Metrics					
		RMSE ↓	REL ↓	MAE ↓	$\delta_{1.05}$ ↑	$\delta_{1.10}$ ↑	$\delta_{1.25}$ ↑
Clear+OOD/ TransCG	[31]	0.061	0.108	0.049	33.59	54.73	92.48
	[41]	0.146	0.262	0.115	13.70	26.39	57.95
	[38]	0.071	0.060	0.036	62.99	82.92	95.93
	Ours	0.039	0.067	0.030	56.68	74.61	98.01
TransCG/ Clear-Real	[31]	0.085	0.095	0.052	47.26	70.76	92.54
	[41]	0.152	0.225	0.139	9.86	20.63	46.02
	[38]	0.045	0.071	0.040	33.43	70.14	99.40
	Ours	0.041	0.054	0.031	62.74	83.31	97.33

TABLE IV
RESULTS OF CROSS-DOMAIN EXPERIMENTS OF DATASETS

Training	Testing	(Global) Metrics*					
		RMSE ↓	REL ↓	MAE ↓	$\delta_{1.05}$ ↑	$\delta_{1.10}$ ↑	$\delta_{1.25}$ ↑
Clear+OOD TransCG	TOD	0.077	0.043	0.037	83.44	89.46	95.53
Clear+OOD TransCG	Clear-Real	0.040	0.058	0.032	55.08	81.23	98.75
		0.041	0.054	0.031	62.74	83.32	97.33

*: For TOD dataset, global metrics is used since masks are not provided. It denotes that the evaluation is conducted on the full depth map instead of the masked area.

dataset and Omniverse object dataset; (2) test the performance on ClearGrasp real-world dataset after training on our TransCG dataset. Results shown in Tab. III reveal that our method is the most robust one among all methods. Also, it is worth noticing that although LIDF-Refine [41] reaches satisfactory results when training domain is similar to the testing domain, the cross-domain testing decreases its performance a lot since its local implicit depth function is environment-dependent. On the contrary, our method is less sensitive to domain changes and is able to achieve satisfactory results under different environment settings.

For cross-domain experiments of different datasets, we select our method as the depth completion model to test the generality of the training dataset on a third-party dataset TOD [20] and ClearGrasp real-world dataset. Results shown in Tab. IV demonstrate that our real-world dataset is more universal compared to the previous synthetic datasets, even though ClearGrasp real-world dataset has a similar environment as ClearGrasp synthetic dataset which is used for training. The performance differences also reflect the shortcomings of synthetic datasets compared to real-world datasets.

C. Real Robot Experiments

To verify the performance of our method in real-world settings, we conduct real robot object grasping experiments. The object grasping pipeline incorporated with our method is introduced in Sec. IV-C. The experiments are conducted on a UR-5 robot with an Intel RealSense D435 camera and a Robotiq two-finger gripper, as shown in Fig. 6.

We randomly select 8 transparent objects to perform real-robot experiments, 6 of which are completely novel, and the rest 2 objects are the same objects without fixers and markers from our testing set. For every experiment, we randomly put the objects and repeat grasping until an object fails for

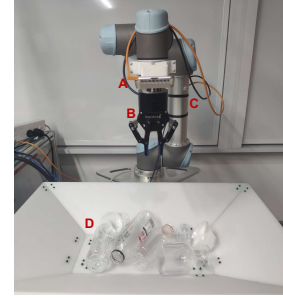


Fig. 6. Real robot experiments on a cluttered scene that consists of daily transparent objects. A: Intel RealSense RGB-D camera. B: Robotiq two-finger gripper. C: UR-5 Robot. D: Daily transparent objects.

3 times. The success rate is defined as $\frac{\#objects}{\#attempts}$, and the completion rate is defined as $\frac{\#successfully-grasped\ objects}{\#objects}$. Table V reports the experiment results, which shows the effectiveness and feasibility of our method. More details are presented in supplementary video.

TABLE V
RESULTS OF REAL ROBOT EXPERIMENTS

	# Objects	# Attempts	Success Rate	Completion Rate
Exper. 1	7	11	63.6%	100.0%
Exper. 2	7	8	87.5%	100.0%
Exper. 3	7	9	77.8%	100.0%
Exper. 4	8	8	100.0%	100.0%
Exper. 5	8	10	80.0%	100.0%
Total	37	46	80.4%	100.0%

VI. CONCLUSION

In this paper, we propose **TransCG**, the first large-scale real-world dataset for transparent object depth completion and grasping, built by our novel data collecting pipeline. Our dataset fills the synthetic-to-real gap in the transparent depth completion area and is more general compared to previous synthetic datasets in real-world environments. Moreover, we propose an end-to-end depth completion network **DFNet**, which is more efficient and robust compared to previous methods according to experiments on various datasets. Real robot experiments of grasping also demonstrate that our method is applicable in real-world settings with novel objects. The compatibility of our method with depth-based manipulation methods allows it to become a default pre-processing step for all downstream tasks concerning transparent objects.

REFERENCES

- [1] Iro Armeni et al. “Joint 2D-3D-Semantic Data for Indoor Scene Understanding”. In: *CoRR* abs/1702.01105 (2017).
- [2] Yunhao Ba et al. “Deep shape from polarization”. In: *European Conference on Computer Vision*. Springer, 2020, pp. 554–571.
- [3] Gary Bradski. “The openCV library.” In: *Dr. Dobb’s Journal: Software Tools for the Professional Programmer* 25.11 (2000), pp. 120–123.
- [4] Zhao Chen et al. “Estimating depth from rgb and sparse sensing”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 167–182.
- [5] Xinjing Cheng, Peng Wang, and Ruigang Yang. “Depth estimation via affinity learned with convolutional spatial propagation network”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 103–119.

- [6] David Eigen, Christian Puhrsch, and Rob Fergus. “Depth map prediction from a single image using a multi-scale deep network”. In: *Advances in neural information processing systems* 27 (2014).
- [7] Hao-Shu Fang et al. “Graspnet-1billion: A large-scale benchmark for general object grasping”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11444–11453.
- [8] Hao-Shu Fang et al. “Rmpe: Regional multi-person pose estimation”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2334–2343.
- [9] Chen Fu et al. “Depth Completion via Inductive Fusion of Planar LIDAR and Monocular Camera”. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 10843–10848.
- [10] Ravi Garg et al. “Unsupervised cnn for single view depth estimation: Geometry to the rescue”. In: *European conference on computer vision*. Springer. 2016, pp. 740–756.
- [11] Minghao Gou et al. “Rgb matters: Learning 7-dof grasp poses on monocular rgb-d images”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 13459–13466.
- [12] Gao Huang et al. “Densely connected convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.
- [13] Jeffrey Ichnowski et al. “Dex-NeRF: Using a Neural Radiance Field to Grasp Transparent Objects”. In: *5th Annual Conference on Robot Learning*. 2021.
- [14] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.
- [15] Marvin Klingner et al. “Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 582–600.
- [16] Byeong-Uk Lee, Kyunghyun Lee, and In So Kweon. “Depth Completion using Plane-Residual Representation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 13916–13925.
- [17] Zhengqin Li, Yu-Ying Yeh, and Manmohan Chandraker. “Through the looking glass: neural 3D reconstruction of transparent shapes”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 1262–1271.
- [18] Hongzhuo Liang et al. “Pointnetgpd: Detecting grasp configurations from point sets”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 3629–3635.
- [19] Xingyu Liu, Shun Iwase, and Kris M Kitani. “StereoObj-1m: Large-scale stereo image dataset for 6d object pose estimation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10870–10879.
- [20] Xingyu Liu et al. “Keypose: Multi-view 3d labeling and keypoint estimation for transparent objects”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11602–11610.
- [21] Ilya Loshchilov and Frank Hutter. “Fixing weight decay regularization in Adam”. In: *CoRR* abs/1711.05101 (2017).
- [22] Fangchang Ma and Sertac Karaman. “Sparse-to-dense: Depth prediction from sparse depth samples and a single image”. In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 4796–4803.
- [23] Jeffrey Mahler et al. “Learning ambidextrous robot grasping policies”. In: *Science Robotics* 4.26 (2019).
- [24] Mehdi Mousavi and Rolando Estrada. “SuperCaustics: Real-time, open-source simulation of transparent objects for deep learning applications”. In: *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2021, pp. 649–655.
- [25] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. “6-dof graspnet: Variational grasp generation for object manipulation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 2901–2910.
- [26] Peiyuan Ni et al. “Pointnet++ grasping: Learning an end-to-end spatial grasp generation algorithm from sparse point clouds”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 3619–3625.
- [27] Andreas ten Pas et al. “Grasp pose detection in point clouds”. In: *The International Journal of Robotics Research* 36.13-14 (2017), pp. 1455–1473.
- [28] Charles R Qi et al. “Pointnet: Deep learning on point sets for 3d classification and segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 652–660.
- [29] Charles Ruizhongtai Qi et al. “Pointnet++: Deep hierarchical feature learning on point sets in a metric space”. In: *Advances in neural information processing systems* 30 (2017).
- [30] Yuzhe Qin et al. “S4g: Amodal single-view single-shot se (3) grasp detection in cluttered scenes”. In: *Conference on robot learning*. PMLR. 2020, pp. 53–65.
- [31] Shreeyak Sajjan et al. “Clear grasp: 3d shape estimation of transparent objects for manipulation”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 3634–3642.
- [32] Dmitry Senushkin et al. “Decoder modulation for indoor depth completion”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 2181–2188.
- [33] Martin Sundermeyer et al. “Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 13438–13444.
- [34] Yingjie Tang et al. “DepthGrasp: Depth Completion of Transparent Objects Using Self-Attentive Adversarial Network with Spectral Residual for Grasping”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 5710–5716.
- [35] Chenxi Wang et al. “Graspness Discovery in Clutters for Fast and Accurate Grasp Detection”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15964–15973.
- [36] Panqu Wang et al. “Understanding convolution for semantic segmentation”. In: *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE. 2018, pp. 1451–1460.
- [37] Enze Xie et al. “Segmenting Transparent Objects in the Wild with Transformer”. In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*. 2021, pp. 1194–1200.
- [38] Haoping Xu et al. “Seeing Glass: Joint Point-Cloud and Depth Completion for Transparent Objects”. In: *Conference on Robot Learning*. PMLR. 2022, pp. 827–838.
- [39] Zhichao Yin and Jianping Shi. “Geonet: Unsupervised learning of dense depth, optical flow and camera pose”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1983–1992.
- [40] Yinda Zhang and Thomas Funkhouser. “Deep depth completion of a single rgb-d image”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 175–185.
- [41] Luyang Zhu et al. “RGB-D Local Implicit Function for Depth Completion of Transparent Objects”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 4649–4658.