TGF-Net: Sim2Real Transparent Object 6D Pose Estimation Based on Geometric Fusion

Haixin Yu, Shoujie Li, Houde Liu, Chongkun Xia, Wenbo Ding, *Member, IEEE*, and Bin Liang

***Abstract*—Transparent objects are a common part of daily life, but their unique optical properties make estimating their 6D pose a challenging task. In this letter, we propose TGF-Net, a monocular instance-level 6D pose estimation method for transparent objects based on geometric fusion. TGF-Net learns the edge features and surface fragments of transparent objects as intermediate features and reduces the influence of appearance changes on the 6D pose estimation of transparent objects by fusing rich geometric features in the network. Moreover, we propose an approach for generating high-fidelity large-scale synthetic datasets of transparent objects using Blender and use this approach to generate a synthetic dataset Trans6D-32 K. Trans6D-32 K contains rendered RGB images and poses information about transparent objects in a variety of different backgrounds, perspectives, and lighting conditions. To evaluate the performance of TGF-Net on 6D pose estimation of transparent objects, we compare with multiple related works on the dataset Trans6D-32 K. TGF-Net can be trained entirely on synthetic datasets without fine-tuning and applied directly to real-world scenarios. Multiple challenging real-scene experiments demonstrate the good performance of TGF-Net, while grasping experiments demonstrate the application value of TGF-Net in transparent object manipulation.**

***Index Terms*—Deep learning for visual perception, data sets for robotic vision, perception for grasping and manipulation.**

I. Introduction

ESTIMATING the 6D pose, i.e., the 3D rotation and 3D translation, of the object with respect to the camera, is a fundamental problem in the field of robot vision. It has been widely used in many real-world tasks, such as robot manipulation [1], [2] and augmented reality [3], [4]. With the deepening of research, researchers have extensively studied various challenging scenes such as textureless objects [5], [6], symmetrical objects [7], [8], and occluded objects [9], [10]. However, as transparent objects are extremely common types of objects in life, their 6D pose estimation has not been fully studied.

Transparent objects possess unique visual properties that make them difficult for robots to perceive and manipulate. The visual characteristics of transparent objects include that they do not have their own texture attributes, most of the generated textures are caused by light and background. At the same time, transparent objects have non-Lambertian surfaces that prevent commercial depth sensors from accurately measuring their depth [11]. The difficult-to-obtain depth information and variable appearance features make vision-based manipulation of transparent objects very challenging. The manipulation research on transparent objects is currently mainly focused on grasping [12]. One method is to complete the missing depth of transparent objects and plan the grasping manipulation according to the geometric features [11], [13], and the other method is to grasp transparent objects via transfer learning with an RGB-based grasping network [14]. However, such manipulation tasks do not involve precise 6D pose information of transparent objects, which makes it difficult to perform advanced manipulation tasks such as liquid pouring [15]. Currently, the 6D pose estimation of transparent objects faces two important challenges. First, large-scale transparent object 6D pose estimation datasets are difficult to obtain [12], [16]. An important difficulty in 6D pose estimation of transparent objects is the influence of background and lighting on the appearance of transparent objects. However, in real scenarios, fully considering different backgrounds and lighting will make the dataset production process extremely difficult. The second challenge we face is the effect of changes in the appearance of transparent objects on the 6D pose estimation. In fact, the dramatic changes in appearance usually lead to incorrect keypoint localization or deviation of dense correspondence in 6D pose estimation methods designed for opaque objects, resulting in erroneous pose estimation results.

In this letter, we study the instance-level transparent object 6D pose estimation method in detail and propose a high-fidelity, large-scale synthetic dataset for transparent object 6D pose estimation and its generation scheme. The overall system is shown in Fig. 1. Our main contributions are as follows:

Manuscript received 23 November 2022; accepted 1 April 2023. Date of publication 17 April 2023; date of current version 17 May 2023. This letter was recommended for publication by Associate Editor G. K. Tummala and Editor C. C. Lerma upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grants 61803221, U1813216, 62203260, and 62104125, in part by the Shenzhen Science Fund for Distinguished Young Scholars under Grant RCJC20210706091946001, in part by Guangdong Young Talent with Scientific and Technological Innovation under Grant 2019TQ05Z111, in part by China Postdoctoral Science Foundation under Grant 2022M711823, in part by Tsinghua SIGS Cross Research and Innovation Fund under Grant JC2021005, in part by Shenzhen Science and Technology Program under Grant JCYJ20220530143013030, and in part by the Natural Science Foundation of Guangdong Province under Grant 2023A1515011773. *(Haixin Yu and Shoujie Li contributed equally to this work.) (Corresponding author: Houde Liu.)*

Haixin Yu, Shoujie Li, Houde Liu, and Chongkun Xia are with the Tsinghua Shenzhen International Graduate School, Shenzhen 518055, China (e-mail: yuhx21@mails.tsinghua.edu.cn; lsj20@mails.tsinghua.edu.cn; liu.hd@sz.tsinghua.edu.cn; xiachongkun@qq.com).

Wenbo Ding is with the Tsinghua Shenzhen International Graduate School, Shenzhen 518055, China, and also with the RISC-V International Open Source Laboratory, Shenzhen 518055, China (e-mail: ding.wenbo@sz.tsinghua.edu.cn).

Bin Liang is with the Navigation and Control Research Center, Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: bliang@mail.tsinghua.edu.cn).

Supplementary material are at https://sites.google.com/view/tgfnet.

Digital Object Identifier 10.1109/LRA.2023.3268041

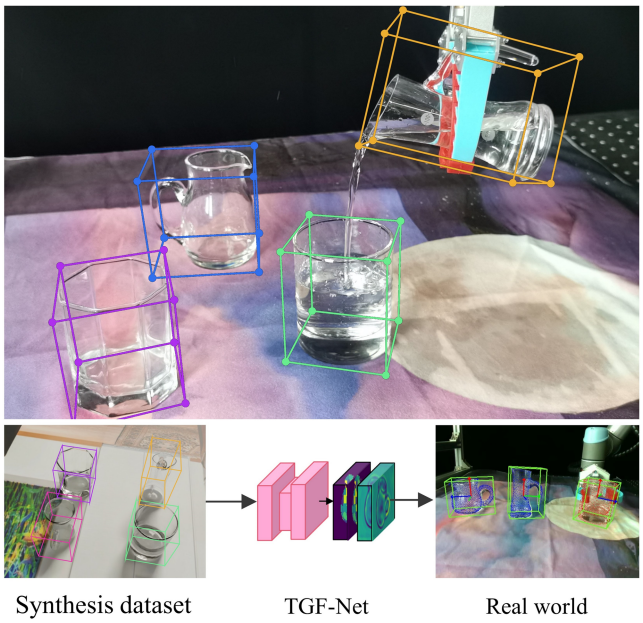

Fig. 1. TGF-Net is trained entirely using synthetic datasets and infers accurate 6D poses of transparent objects from monocular RGB images. The estimated 6D pose can be directly used for downstream manipulation tasks of the robot (e.g. grasping and pouring).

- We propose a monocular instance-level transparent object 6D pose estimation method based on geometric feature fusion named TGF-Net, which is robust to changes in the appearance of transparent objects.
- We propose a synthetic high-fidelity, large-scale dataset Trans6D-32 K for transparent objects 6D pose estimation and a low-cost dataset generation scheme, which solve the "difficult to label" problem of transparent objects.
- We design several experiments to demonstrate that TGF-Net can effectively implement sim2real while being robust to changing backgrounds, lighting, and liquids of various colors inside. The experiments also prove that the use of a synthetic transparent object 6D pose estimation dataset can effectively reduce the cost of dataset production and achieve sim2real.

The rest of the letter is organized as follows: Section II introduces the related work of transparent object perception, transparent object dataset and 6D pose estimation; Section III describes in detail the dataset Trans6D-32 K, the generation method, and the architecture of the network TGF-Net; Section IV presents the experimental evaluation and results; and Section V concludes the letter.

II. Related Works

A. Transparent Object Visual Perception and Manipulation

The perception and manipulation of transparent objects have long been studied. To solve the problem that the depth information of transparent objects cannot be accurately measured by commercial depth cameras, one solution is to estimate the missing depth through depth completion. Sajjan et al. [11] estimated the surface normals, masks, and occlusion boundaries of transparent objects, and used this information to complete the missing depth. There are also [13] and [17] in the research on the depth completion of transparent objects. Transparent objects can be grasped through the completed depth information. Another solution is to use a binocular camera to estimate the disparity map from the stereo input, thus avoiding the direct measurement of depth using the depth camera. Liu et al. [18] proposed a keypoint-based method using stereo input to estimate the 6D pose of transparent objects, planning grasps through accurate 6D pose estimation. Other studies have focused on using devices such as polarization cameras to obtain more prominent images of transparent objects [19]. However, a more convenient solution has not been widely studied, that is, 6D pose estimation of transparent objects through monocular RGB images, which can avoid the use of binocular cameras and deal with the broken depth information of transparent objects, and can be better deployed on mobile.

B. Transparent Object Datasets

With the deepening of research on transparent objects, more and more transparent object datasets have been proposed. Chen et al. [20] proposed a transparent object matting dataset containing 876 real images and 178 K synthetic images. Fang et al. [12] proposed a large-scale real dataset for depth completion and grasping. Liu et al. proposed a large-scale stereo RGB dataset StereOBJ-1M [21] containing 396 K frames for 6D pose estimation. Xie et al. proposed Trans10K [22], a dataset for transparent object segmentation. After that, the transparent objects were further subdivided and a new dataset Trans10K-v2 [23] was proposed.

Although the above datasets contain most aspects of transparent object research, they all show the characteristics of difficulty and high cost of labeling. And for some instance-level tasks, it is necessary to re-create the dataset when changing different instances. Moreover, in the actual situation, it is difficult to find a large number of changing backgrounds and lights. The above problems not only hinder the research on transparent objects but also hinder the practical application of transparent object perception and manipulation schemes. The application of synthetic data in the field of computer vision brings opportunities to solve these problems. Li et al. proposed a transparent object grasping synthetic dataset [24] for TaTa [25], but it contains only RGB data. [26] and [27] generate large-scale synthetic datasets with 3D ground truth by using synthetic human models, and the datasets contain random changes in environments, clothing, etc. Other studies obtain a large amount of labeled training data by directly capturing video game scenes to complete tasks such as segmentation [28], [29], human tracking [30] and human recovery [31]. Inspired by these studies, we propose to use Blender's physics engine [32] and a physically-based ray-tracing Blender Cycles rendering engine to generate large-scale transparent object datasets in a short time by rendering.

Fig. 2. (a) Dataset generation scenarios. (i) Scene setup for generating the transparent object synthetic dataset using Blender. (ii) Transparent object CAD models, #01 to #10 from left to right. (b) Train dataset in Trans6D-32 K. (c) Test dataset in Trans6D-32 K.

C. 6D Pose Estimation

According to the different input information required, 6D pose estimation can be mainly divided into RGB-D methods that rely on depth information and methods that only use RGB information. Although RGB-D methods can usually achieve higher performance [33], [34], [35], the depth information of transparent objects is noisy and broken, which brings huge errors. For the above reasons, we focus on methods that use only RGB information.

At present, the mainstream method of RGB-based 6D pose estimation is the corresponding method. BB8 [36] uses the corners of the 3D bounding box as the keypoints and predicts the 2D projection point position of the 3D keypoints through the network. PVNet [9] uses a pixel-wise voting method to locate sparse keypoints located on objects, and then constructs 2D-3D correspondences. With the deepening of research, more and more studies have begun to use dense correspondence methods. DPOD [37] proposes to use discrete UV maps to parameterize object surfaces, estimating a dense 2D-3D correspondence map between input images and available 3D models. GDR-Net [38] combines the ideas of CDPN [39] and EPOS [7], divides the surface of the object into fragments, and directly outputs the pose information. There is also zebrapose [40], which achieves fine-grained correspondence prediction by dividing the surface of the object into codes from coarse to fine.

III. Method

A. Dataset Generation

Due to the unique optical properties of transparent objects, their surface features will be seriously affected by lighting, background, and so on. Although some transparent object datasets have been proposed in previous research, the generation process of these datasets is very complex, and the variation of background and lighting is also very limited. At the same time, recreating the dataset is expensive when the object types or labels in the dataset need to be changed.

To more easily generate datasets with multiple backgrounds and changing lighting, we use Blender [41] to render transparent object datasets. Each frame of the rendered image contains a unique background captured from videos, including outdoor and indoor scenes. Two point lights are set in the scene, they illuminate the scene from directly above and from the side, and in each frame of the rendered image, the illumination intensity fluctuates randomly within $\pm 20\%$. During the rendering process, we randomly change the distance and angle between the camera and the object. In order to prevent excessive occlusion between objects, we limit the angle between the camera's optical axis and the z-axis of the world coordinate system between 22.5° and 67.5°. The application of random factors greatly increases the robustness and generalization ability of the model, so that the trained model can be directly transferred to the real scene. The scene setup for generating the synthetic dataset of transparent objects using Blender is shown in Fig. 2(a).

The proposed dataset contains ten kinds of objects, all of which are common types of objects in households. In order to include as many types of objects as possible, the ten objects include 5 symmetrical objects and 5 asymmetrical objects. Since the synthetic dataset is made without noise such as motion blur and camera distortion, it is purer than the real dataset. We select 400 images of each object for training, which is large enough to train an accurate 6D pose estimation network. At the same time, we generate 2800 images for each object as a test dataset, so the entire dataset contains 32000 images, which we name Trans6D-32 K. The test dataset and the train dataset do not have exactly the same background, which makes the background gap between them larger, so as to verify whether our method can resist the interference of the background by learning geometric features. Examples of the train dataset and the test dataset are shown in (b) and (c) in Fig. 2, respectively.

B. Network Architecture

The proposed TGF-Net network is shown in Fig. 3. TGF-Net is inspired by CDPN [39], a direct 6D pose estimation method based on dense correspondence. Specifically, we replace the rotation head and translation head of CDPN with two parallel flows: regular flow and geometric flow. ResNet-34 [42] is used as the backbone network. The geometric flow is responsible for extracting geometric information of objects. In geometric flow, we

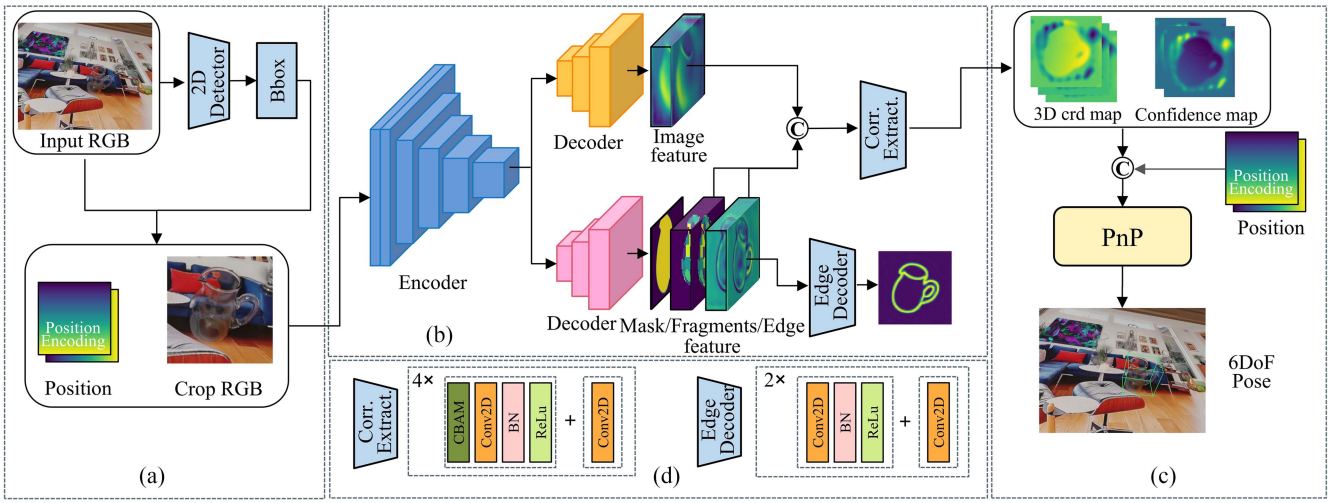

Fig. 3. Overview of our network. Our network consists of three building blocks: (a) Object detection and image enhancement. (b) The main structure of the network. After the input image is encoded and decoded, a regular flow and a geometric flow are formed, and the correspondence is extracted by fusion between the two flows. (c) A learnable PnP network for 6D pose estimation. (d) The detailed structure of part of the network, where CBAM means convolutional block attention module [43].

predict surface fragments, edge features and object mask. This part of geometric features are less disturbed by object color and can effectively avoid the influence of background and lighting changes on 6D pose estimation of transparent objects. Since the edge features of transparent objects have higher contrast than surface fragments, it is easier to locate, which is also in line with human visual perception. Therefore, we use edge features as auxiliary information to localize surface fragments. We also predict the mask of the object, which also helps to describe the geometric information of the object. However, the characteristics of transparent objects cannot be effectively described only by geometric information. The features such as surface brightness of transparent objects can also assist in describing the pose of transparent objects. Therefore, we extract features in addition to geometric feature through the regular flow, so as to provide a more comprehensive description of transparent objects.

After completing feature extraction with geometric flow and regular flow, the extracted features are concatenated, and a correspondence extraction network is used to extract the features to obtain the output dense coordinate map and the confidence of the two channels. The 2D-3D correspondence is then obtained by stacking the dense coordinate map with the 2D pixel coordinates. In the subsequent network, we use the Patch-PnP proposed by GDR-Net [38] to solve the 6D pose. Different from GDR-Net, TGF-Net uses 2D-3D correspondence as well as confidence map as the input of Patch-PnP.

C. Surface Fragments

Inspired by methods such as ZebraPose [7], [38], [40], surface fragments are applied to the 6D pose estimation of transparent objects. Surface fragments contain rich geometry information, which can effectively reduce the impact of texture changes of transparent objects on the recognition and 6D pose estimation. Directly regressing 3D coordinates without geometry drive is more likely to produce smooth output, which ignores some sharp geometric information, such as the handle of a cup, resulting in inaccurate 3D coordinate estimation. Using surface fragments can transform the regression task into a classification task and allow the algorithm to perform 3D regression in a smaller area. In the case of the known object model point set, use the farthest point sampling algorithm to select surface fragments center point set $\mathcal{C} = \{\mathbf{c}_j \in \mathbb{R}^3 \mid j = 1, 2, \ldots, N\}$ from the model point set $\mathcal{O} = \{\mathbf{x}_i \in \mathbb{R}^3 \mid i = 1, 2, \ldots, M\}$. The surface fragment S_j is defined as:

$$S_j = \{s \mid s \in \mathcal{O} \wedge d(s, c_j) < d(s, c_k)\} \forall c_k \in \mathcal{C}, k \neq j, \quad (1)$$

where $d(.)$ represents the Euclidean distance between the model point and the pre-selected surface fragments center point.

D. Edge Feature

The edges of transparent objects have the highest contrast, so they are more stable and prominent in the changing background. It is easier to use the network to learn surface fragments at the same time as edge features, rather than locating surface fragments directly in the image. However, due to the special optical properties of transparent objects, the edge cannot be directly detected by Canny operator like [44], because it will be seriously disturbed by background information and lighting. Therefore, we propose a novel method for automatically extracting the edge of transparent objects. By projecting the object model point into the depth map D and detecting edges in the depth map, the influence of color features can be avoided and more geometric features can be obtained. The edge map E is obtained by using the Canny operator on depth map D. Points above the detected edge line are set to 1, other points are set to 0. The distance conversion map L can be obtained from the edge map E, which represents the distance of each pixel from the nearest edge point. To make it easier for the network to

learn edge features, we convert the edge map E to the ground truth edge heatmap M using Gaussian expression with standard deviation σ, the Gaussian kernel is shown as (2):

$$M(u,v) = \frac{1}{2\pi\sigma^2} e^{-\left(L(u,v)^2\right)/\left(2\sigma^2\right)}, \tag{2}$$

where (u, v) represents the position of the pixel in the image, the standard deviation σ is set to 1.4.

E. Loss Function

The objective function $\mathcal{L}$ of training TGF-Net is divided into two parts, namely the geometric loss function $\mathcal{L}_{GEO}$ and the pose loss function $\mathcal{L}_{Pose}$, $\mathcal{L}$ is described as (3):

$$\mathcal{L} = \mathcal{L}_{GEO} + \mathcal{L}_{Pose}. \tag{3}$$

The geometric loss $\mathcal{L}_{GEO}$ describes the learning of the geometric intermediate representation, which includes the edge loss $\mathcal{L}_{edge}$, the surface fragments loss $\mathcal{L}_{fragment}$, the mask loss $\mathcal{L}_{mask}$ and the dense coordinates maps loss $\mathcal{L}_{xyz}$, so the geometric loss $\mathcal{L}_{GEO}$ can be expressed as (4):

$$\mathcal{L}_{GEO} = \lambda_1\mathcal{L}_{edge} + \lambda_2\mathcal{L}_{fragment} + \lambda_3\mathcal{L}_{mask} + \lambda_4\mathcal{L}_{xyz}. \tag{4}$$

The pose loss $\mathcal{L}_{Pose}$ represents the accuracy of the 6D pose estimation, which includes the point matching loss [45] $\mathcal{L}_{pm}$, center loss $\mathcal{L}_{center}$ and distance loss $\mathcal{L}_z$, so the pose loss $\mathcal{L}_{Pose}$ can be expressed as (5):

$$\mathcal{L}_{Pose} = \lambda_5\mathcal{L}_{pm} + \lambda_6\mathcal{L}_{center} + \lambda_7\mathcal{L}_z. \tag{5}$$

In the loss function, λ_i is the trade-off hyper-parameter, where $i \in \{1, 2, \ldots, 7\}$. During training, we set λ_1 to 15, λ_2 to 0.1, λ_3 to 0.0001, λ_4, λ_5, λ_6, λ_7 are set to 1. In the loss function, $\mathcal{L}_{fragment}$ is the cross entropy loss, $\mathcal{L}_{xyz}$, $\mathcal{L}_{mask}$, $\mathcal{L}_{center}$, $\mathcal{L}_z$ are $L1$ loss, $\mathcal{L}_{pm}$ is point matching loss, and $\mathcal{L}_{edge}$ is Dice Loss [46]. For Dice Loss, dice coefficient DC is a value between 0 and 1, expressed as (6):

$$DC = \frac{2\sum_{u,v} P_{uv}G_{uv}}{\sum_{u,v} P_{uv}^2 + \sum_{u,v} G_{uv}^2}, \tag{6}$$

where P_{uv} and G_{uv} refer to the predicted and ground truth pixel values at (u, v) respectively.

IV. Experiments

A. Comparison With State of the Art

Implementation Details: All the experiments were conducted on an Intel(R) Xeon(R) Gold 6256 CPU with an NVIDIA 3090 GPU, using the deep learning framework Pytorch. We train TGF-Net using the Ranger optimizer with a batch size of 64. During training, we found that a new training strategy can achieve better results. First, we only take edges, surface fragments, mask and dense coordinate map as the optimization goals of the network, without supervising the pose information. At this stage, we set the learning rate to 0.0002 for all parts of the network. Then, we train the entire network at the same time, the learning rate of geometric flow and backbone is reduced to 0.0001, and the learning rate of other parts is 0.0002. We trained 240 epochs at each stage. Our new training strategy outperforms end-to-end training, we believe that the first stage of training can extract finer geometric features of transparent objects, and when the entire network is trained at the same time, more accurate 6D pose estimation can be effectively obtained with the help of finer geometric features.

Evaluation Metrics: We measure the accuracy of 6D pose estimation using two main metrics: ADD(-S) and 2D Projection. ADD(-S) includes ADD and ADD-S, which are designed for asymmetric and symmetric objects, respectively. For ADD, a pose is considered correct if the average distance of the model points between the predicted pose and the ground truth pose is smaller than 10% of the model diameter. For ADD-S, the closest model point is used to compute the average distance. Given an object with 3D model point set of $\mathcal{O} = \{\mathbf{x}_i \in \mathbb{R}^3 \mid i = 1, 2, \ldots, M\}$:

$$\text{ADD} = \frac{1}{|\mathcal{O}|}\sum_{\mathbf{x}\in\mathcal{O}} \left\|(\mathbf{R}\mathbf{x} + \mathbf{T}) - (\mathbf{R}^*\mathbf{x} + \mathbf{T}^*)\right\| \tag{7}$$

$$\text{ADD–S} = \frac{1}{|\mathcal{O}|}\sum_{\mathbf{x}_1\in\mathcal{O}} \min_{\mathbf{x}_2\in\mathcal{O}} \left\|(\mathbf{R}\mathbf{x}_1 + \mathbf{T}) - (\mathbf{R}^*\mathbf{x}_2 + \mathbf{T}^*)\right\|, \tag{8}$$

where $[\mathbf{R}, \mathbf{T}]$ is the predicted pose and $[\mathbf{R}^*, \mathbf{T}^*]$ is the ground truth pose.

When computing 2D projection error, the model point set is transformed by the predicted and the ground truth poses respectively. Given the camera projection function $\Pi : \mathbb{R}^3 \to \mathbb{R}^2$, the 2D projection error 2D–Proj. is calculated as (9):

$$2\text{D} - \text{Proj.} = \frac{1}{|\mathcal{O}|}\sum_{\mathbf{x}\in\mathcal{O}} \left\|\Pi(\mathbf{R}\mathbf{x} + \mathbf{T}) - \Pi\left(\mathbf{R}^*\mathbf{x} + \mathbf{T}^*\right)\right\|. \tag{9}$$

A pose is considered correct if the average 2D projection error of the object's model points is smaller than 2 pixels.

Comparison Results: We compare our method with PVNet [9], CDPN [39] and GDR-Net [38]. We reimplemented these methods and trained and tested on the Trans6D-32 K dataset. For a fair comparison, we used the hyperparameters from the official implementation of each method. The results of the comparison experiment are illustrated in Table I, the visualizations of results of our method on the Trans6D-32 K dataset are illustrated in Fig. 4. Our method achieves the best mean ADD(-S) accuracy of 88.2% among all other methods. In terms of 2D projection accuracy, our method also outperforms all other methods with 60.0% accuracy. In addition to the end result, we still found some unique properties. In #03 of asymmetric objects, both PVNet and CDPN achieve better results than other asymmetric objects, while #03 in our method and GDR-Net is not as good as other asymmetric objects. This is because the geometry of #03 is more complex, with more bends and curved surfaces, which makes it easier for PVNet and CDPN to locate keypoints or solve 3D correspondence through features such as curved surfaces. However, this complex geometry is not friendly to surface fragments or edge features, so the performance is not

TABLE I
COMPARISON WITH STATE-OF-THE-ART RGB-ONLY METHODS ON TRANS6D-32 K DATASET USING ADD(-S) AND 2D PROJECTION

Metrics	ADD(-S)				2D Proj.			
Methods	PVNet	CDPN	GDR-Net	Ours	PVNet	CDPN	GDR-Net	Ours
#01	13.4	16.9	73.5	**83.4**	2.5	2.0	46.1	**47.4**
#02	17.2	36.3	76.3	**83.5**	4.0	4.9	**36.4**	35.8
#03	26.2	38.9	67.4	**67.7**	10.0	17.4	49.5	**49.8**
#04	19.6	37.4	82.9	**85.4**	4.6	5.8	49.0	**49.6**
#05	28.7	42.0	78.5	**89.6**	5.9	2.6	30.5	**35.5**
#06	39.1	62.2	**91.5**	89.6	41.5	13.6	**90.5**	90.0
#07	56.2	60.8	90.6	**92.6**	13.8	13.6	**75.4**	74.3
#08	66.3	84.9	96.6	**97.3**	7.5	10.2	**71.1**	69.3
#09	54.4	83.3	96.0	**97.5**	5.2	11.8	67.0	**69.8**
#10	35.9	52.3	92.4	**94.9**	15.7	2.6	**78.2**	77.9
Mean	35.7	51.5	84.6	**88.2**	11.1	8.4	59.4	**60.0**

The bold values indicate optimal values within the same metric.

Fig. 4. Visualizations of results on the dataset. Blue 3D bounding boxes represent the ground truth poses while green 3D bounding boxes represent our predictions.

as good as other asymmetric objects. In symmetrical objects, PVNet and CDPN do not work well for #10, because the glass wall of #10 is thinner relative to other objects, so in the case of background changes, the color of the background will interfere more obviously.

B. Ablation Studies

In order to verify the influence of geometric representation on the network, we conduct ablation experiments, and all ablation experiments use the same network initialization scheme and training scheme.

1) The Effect of Surface Fragments: We remove the prediction of surface fragments in the geometry flow, and only keep the edge features and mask, which makes the results drop to a certain extent. However, since the edge features are also able to encode location information, this results in no drastic drop in results.

2) The Effect of Edge Features: We remove the use of edge features in the geometry flow, and only keep the surface fragments and mask, and the prediction accuracy also decreases to a certain extent. Since the surface fragments can describe the position, there is no significant decrease.

Through these ablation experiments, we know that both surface fragments and edge features can encode geometric information, and when they are used together, the geometric information can be described more effectively. The results of ablation study are illustrated in Table II.

TABLE II
ABLATION STUDY OF GEOMETRY FEATURE IN ADD(-S) AND 2D PROJECTION ON TRANS6D-32 K

	Edge	Fragments	ADD(-S)	2D Proj.
w/o geometry feature	✗	✗	82.0	54.0
w/o edge feature	✗	✓	86.6	58.0
w/o surface fragments	✓	✗	84.8	55.0
Full model	✓	✓	**88.2**	**60.0**

C. Sim2Real Experiment

To verify that the proposed TGF-Net can be directly applied to real-world scenarios, we conduct experiments with real objects.

Hardware: We use a UR5 robotic arm with a soft gripper for real-world experiments. Compared with rigid grippers, soft grippers can more effectively grasp cups with larger diameters and perform manipulations such as pouring. The lighting of the experimental platform comes from the combined effect of lighting and sunlight. A RealSense D435 camera is installed on the side of the platform for 6D pose estimation of transparent objects. TGF-Net is implemented using a desktop computer with an i7-11700 CPU and an NVIDIA 2060 GPU.

Implementation: We selected fixed grasp points based on the CAD model of the object in advance and trained a YOLOv5 [47] detection network using the synthetic dataset Trans6D-32 K to detect the 2D bounding box of the object. We conducted a total of four experiments to study the effects of changing backgrounds, different internal liquids, and changing lights on the 6D pose estimation of transparent objects. Finally, we used a robotic arm to grasp transparent objects to demonstrate the application value of our method. The experimental results are shown in Fig. 5.

1) Experiment 1: 6D Pose Estimation of Transparent Objects on Different Backgrounds: To verify that Trans6D-32 K and TGF-Net can directly perform sim2real and resist the effects of background changes, we deployed method such as GDR-Net, CDPN and PVNet at the same time and compared them with ours. In our experiments, we used four different backgrounds with complex lines and rich colors. The experimental results are shown in Fig. 5(a). By comparison, TGF-Net is more stable,

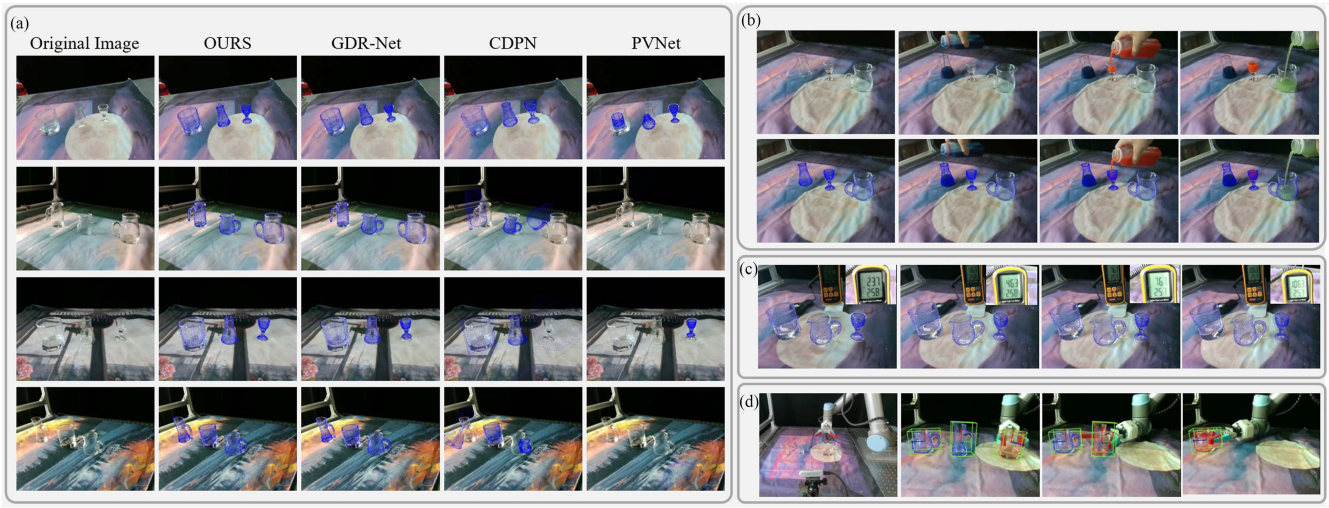

Fig. 5. Qualitative results in real scenarios. Each object point cloud is transformed with the estimated pose and then projected to the 2D image. (a) Real-world experiments in different background. We compare TGF-Net with GDR-Net, CDPN and PVNet under the same conditions. (b) The 6D pose estimation of transparent objects with different colored internal liquids. (c) The 6D pose estimation of transparent objects under different illumination conditions. (d) Transparent objects grasping experiment. The robotic arm grasps the transparent object according to the 6D pose, and the red point cloud projection indicates the selected object to be grasped.

although the GDR-Net also has good results, it has a larger offset at the edge of the object.

2) Experiment 2: 6D Pose Estimation of Transparent Objects With Different Internal Liquids: Changes in appearance caused by liquid in a transparent container are different from changes caused by background and lighting. In order to verify that our method can effectively deal with such situations, we add translucent solutions of different colors into the cup, and estimate the 6D pose of the transparent object during the process of adding the liquid. According to the experimental results in Fig. 5(b), it can be proved that our method is still stable and accurate even in the process of adding liquid.

3) Experiment 3: 6D Pose Estimation of Transparent Objects Under Different Lighting Conditions: Lighting also has a big effect on the appearance of transparent objects. On the one hand, the light will be reflected on the surface of the transparent object to generate bright spots, which will interfere with the detection of the geometric information of the transparent object. On the other hand, the light refracted by the transparent object will be irradiated near the transparent object, resulting in obvious changes in positions such as the edge of the transparent object due to the illumination. To demonstrate that our method can effectively cope with changes in lighting, we illuminate the experimental platform with a fill light and gradually increase the light intensity. As the light intensity increases from 23.7 LUX to 106.3 LUX, our method consistently shows good performance. The experimental results are shown in Fig. 5(c).

4) Experiment 4: Grasping Experiment of Transparent Objects: In order to verify the application value of our method in the real-world scenarios, we use the robotic arm to grasp the transparent objects. We estimate the 6D pose of the transparent objects, and sequentially grasp them according to the pre-specified grasping points, as shown in Fig. 5(d). We conduct multiple experiments to count the success rate of grasping. The success criterion is that the object can be transported to the designated position after grasping. In 40 grasping experiments, our method succeed 38 times in total, and the success rate reached 95%.

V. Conclusion

In this letter, we propose TGF-Net, a monocular instance-level 6D pose estimation method for transparent objects. TGF-Net integrates geometric features such as surface fragments and edge features, which can effectively cope with the impact of changing backgrounds and internal solutions on the appearance of transparent objects. At the same time, we propose a low-cost synthetic dataset generation scheme for transparent objects, through which we generate a high-fidelity large-scale transparent object 6D pose estimation dataset Trans6D-32 K. By training TGF-Net on the synthetic dataset Trans6D-32 K and performing sim2real directly, TGF-Net shows more stable and reliable results in real-world experiments than other related methods. We also demonstrate that training with high-fidelity transparent object synthetic datasets can be directly transferred to real-world scenarios, thus greatly reducing the difficulty of labeling transparent object datasets.

References

[1] 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 *Proc. Conf. Robot Learn.*, 2018.

[2] S. Tyree et al., "6-DoF pose estimation of household objects for robotic manipulation: An accessible dataset and benchmark," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 13081–13088.

[3] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: A hands-on survey," *IEEE Trans. Visualization Comput. Graph.*, vol. 22, no. 12, pp. 2633–2651, Dec. 2016.

[4] Y. Su, J. Rambach, N. Minaskan, P. Lesur, A. Pagani, and D. Stricker, "Deep multi-state object pose estimation for augmented reality assembly," in *Proc. IEEE Int. Symp. Mixed Augmented Reality Adjunct*, 2019, pp. 222–227.

[5] T. Hodan, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, and X. Zabulis, "T-Less: An RGB-D dataset for 6 D pose estimation of texture-less objects," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2017, pp. 880–888.

[6] C. Wu, L. Chen, Z. He, and J. Jiang, "Pseudo-siamese graph matching network for textureless objects'6-D pose estimation," *IEEE Trans. Ind. Electron.*, vol. 69, no. 3, pp. 2718–2727, Mar. 2022.

[7] T. Hodan, D. Barath, and J. Matas, "EPOS: Estimating 6D pose of objects with symmetries," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11703–11712.

[8] J. Richter-Klug and U. Frese, "Handling object symmetries in CNN-based pose estimation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 13850–13856.

[9] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNet: Pixel-wise voting network for 6DoF pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4561–4570.

[10] G. Wang, F. Manhardt, X. Liu, X. Ji, and F. Tombari, "Occlusion-aware self-supervised monocular 6 D object pose estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Dec. 17, 2021, doi: 10.1109/TPAMI.2021.3136301.

[11] S. Sajjan et al., "Clear grasp: 3D shape estimation of transparent objects for manipulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 3634–3642.

[12] H. Fang, H.-S. Fang, S. Xu, and C. Lu, "TransCG: A large-scale real-world dataset for transparent object depth completion and a grasping baseline," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 7383–7390, Jul. 2022.

[13] L. Zhu et al., "RGB-D local implicit function for depth completion of transparent objects," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4649–4658.

[14] T. Weng, A. Pallankize, Y. Tang, O. Kroemer, and D. Held, "Multi-modal transfer learning for grasping transparent and specular objects," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 3791–3798, Jul. 2020.

[15] G. Narasimhan, K. Zhang, B. Eisner, X. Lin, and D. Held, "Self-supervised transparent liquid segmentation for robotic pouring," in *Proc. Int. Conf. Robot. Automat.*, 2022, pp. 4555–4561.

[16] X. Chen, H. Zhang, Z. Yu, A. Opipari, and O. C. Jenkins, "Clearpose: Large-scale transparent object dataset and benchmark," in *Proc. 17th Eur. Conf. Comput. Vis.*, 2022, pp. 381–396.

[17] J. Jiang, G. Cao, T.-T. Do, and S. Luo, "A4T: Hierarchical affordance detection for transparent objects depth reconstruction and manipulation," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 9826–9833, Oct. 2022.

[18] X. Liu, R. Jonschkowski, A. Angelova, and K. Konolige, "Keypose: Multi-view 3D labeling and keypoint estimation for transparent objects," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11602–11610.

[19] D. Gao et al., "Polarimetric pose prediction," in *Proc. 17th Eur. Conf. Comput. Vis.*, 2022, pp. 735–752.

[20] G. Chen, K. Han, and K.-Y. K. Wong, "TOM-NET: Learning transparent object matting from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9233–9241.

[21] X. Liu, S. Iwase, and K. M. Kitani, "StereOBJ-1 M: Large-scale stereo image dataset for 6D object pose estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10870–10879.

[22] E. Xie, W. Wang, W. Wang, M. Ding, C. Shen, and P. Luo, "Segmenting transparent objects in the wild," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 696–711.

[23] E. Xie et al., "Segmenting transparent object in the wild with transformer," in *Proc. 30th Int. Joint Conf. Artif. Intell.*, 2021, pp. 1194–1200.

[24] S. Li et al., "Visual-tactile fusion for transparent object grasping in complex backgrounds," 2022, *arXiv:2211.16693*.

[25] S. Li, X. Yin, C. Xia, L. Ye, X. Wang, and B. Liang, "Tata: A universal jamming gripper with high-quality tactile perception and its application to underwater manipulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 6151–6157.

[26] E. G. Bazavan, A. Zanfir, M. Zanfir, W. T. Freeman, R. Sukthankar, and C. Sminchisescu, "HSPACE: Synthetic parametric humans animated in complex environments," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021.

[27] P. Patel, C.-H. P. Huang, J. Tesch, D. T. Hoffmann, S. Tripathi, and M. J. Black, "AGORA: Avatars in geography optimized for regression analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13468–13478.

[28] P Krähenbühl, "Free supervision from video games," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2955–2964.

[29] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *Proc. 14th Eur. Conf. Comput. Vis.*, 2016, pp. 102–118.

[30] M. Fabbri, F. Lanzi, S. Calderara, A. Palazzi, R. Vezzani, and R. Cucchiara, "Learning to detect and track visible and occluded body joints in a virtual world," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 430–446.

[31] Z. Cai et al., "Playing for 3D human recovery," 2021, *arXiv:2110.07588*.

[32] Blender, "Blender physics engine," 2022. [Online]. Available: https://docs.blender.org/manual/en/latest/physics/index.html

[33] W. Chen, X. Jia, H. J. Chang, J. Duan, and A. Leonardis, "G2L-Net: Global to local network for real-time 6 D pose estimation with embedding vector features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4233–4242.

[34] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11632–11641.

[35] C. Wang et al., "DenseFusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3343–3352.

[36] M. Rad and V. Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3828–3836.

[37] S. Zakharov, I. Shugurov, and S. Ilic, "DPOD: 6D pose object detector and refiner," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1941–1950.

[38] G. Wang, F. Manhardt, F. Tombari, and X. Ji, "GDR-Net: Geometry-guided direct regression network for monocular 6D object pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16611–16621.

[39] Z. Li, G. Wang, and X. Ji, "CDPN: Coordinates-based disentangled pose network for real-time RGB-based 6-DoF object pose estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7678–7687.

[40] Y. Su et al., "ZebraPose: Coarse to fine surface encoding for 6DoF object pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6738–6748.

[41] Blender Online Community, "Blender - a 3D modelling and rendering package," 2018. https://www.blender.org/

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[43] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.

[44] Y. Wen, H. Pan, L. Yang, and W. Wang, "Edge enhanced implicit orientation learning with geometric prior for 6D pose estimation," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4931–4938, Jul. 2020.

[45] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "DeepIM: Deep iterative matching for 6D pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 683–698.

[46] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. IEEE 4th Int. Conf. 3D Vis.*, 2016, pp. 565–571.

[47] G. Jocher, "Ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation" Nov. 2022, doi: 10.5281/zenodo.7347926.