TransGrasp: Grasp Pose Estimation of a Category of Objects by Transferring Grasps from Only One Labeled Instance (Supplementary Material)

Hongtao Wen*, Jianhang Yan*, Wanli Peng†, and Yi Sun

Table 1. Detailed results of real robot experiments in the Table 3 of our main paper

Category	Instances	GPD[3]					6-DOF GraspNet[2]					TransGrasp				
		#1	#2	#3	#4	#5	#1	#2	#3	#4	#5	#1	#2	#3	#4	#5
Mug		✔	✘	✔	✘	✘	✔	✔	✘	✔	✘	✘	✔	✔	✔	✔
		✘	✘	✘	✘	✘	✘	✔	✘	✘	✔	✘	✘	✔	✘	✘
		✘	✔	✘	✘	✘	✔	✔	✔	✔	✘	✘	✔	✔	✔	✔
		✘	✔	✔	✔	✔	✔	✘	✘	✔	✔	✔	✔	✔	✔	✔
		✘	✘	✘	✘	✘	✘	✔	✔	✔	✔	✔	✔	✔	✔	✔
	Success Rate	7 / 25					16 / 25					**19 / 25**				
Bottle		✔	✔	✔	✔	✔	✘	✔	✘	✔	✔	✔	✔	✔	✘	✔
		✘	✘	✘	✔	✘	✘	✘	✔	✘	✘	✘	✔	✘	✔	✔
		✔	✔	✔	✔	✔	✔	✘	✔	✔	✘	✔	✔	✔	✔	✔
		✘	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
		✘	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✘
	Success Rate	19 / 25					17 / 25					**21 / 25**				
Bowl		✘	✘	✘	✔	✘	✘	✘	✔	✔	✔	✔	✔	✔	✔	✔
		✘	✔	✘	✔	✘	✔	✔	✔	✔	✔	✔	✔	✘	✔	✔
		✔	✔	✘	✘	✘	✔	✔	✔	✔	✘	✘	✔	✔	✘	✔
		✘	✔	✔	✔	✔	✔	✔	✘	✔	✔	✘	✔	✔	✘	✔
		✔	✔	✘	✘	✘	✘	✔	✔	✘	✔	✔	✔	✘	✔	✔
	Success Rate	11 / 25					**19 / 25**					**19 / 25**				
Average Success Rate		37 / 75					52 / 75					**59 / 75**				

* H. Wen and J. Yan—Equal contributions. † W. Peng—Corresponding author.

1 Detailed Results of Real Robot Experiments

Table 1, where ✔and ✘denote successful and failed trials, respectively, shows the detailed results of the real robot experiments in Table 3 of our main paper. For each method, we performed 5 trials per object and totally 25 trials per category. Our method outperforms GPD[3] and 6-DOF GraspNet[2] for the mug and bottle category, and achieves the same performance as 6-DOF GraspNet for the bowl category. It's noteworthy that for the transparent bottle, the success rate of our method is significantly higher than that of GPD and 6-DOF GraspNet. Owing to the shape reconstruction for the target object, our method can easily reduce the influence of background points on grasp estimation, unlike GPD and 6-DOF GraspNet that generate grasps directly from point clouds.

2 Objects of Various Shapes in Each Category

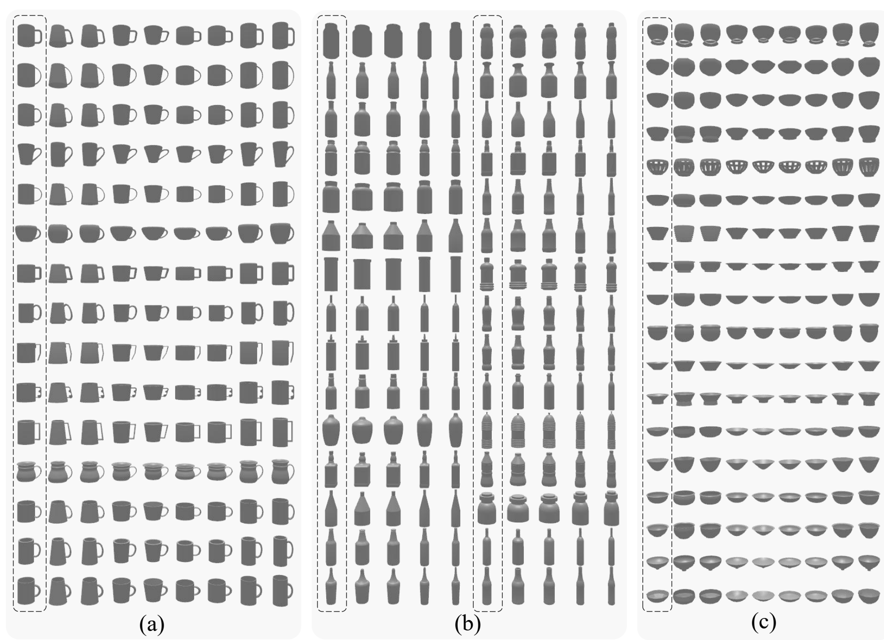

Fig. 1. Objects of various shapes in the test sets of each category, where the models in the dashed box are the original models selected from ShapeNetCore [1] and the others are the models after augmentation

To increase the diversity of shapes in the dataset, we introduce a data augmentation strategy by deforming the models selected from ShapNetCore [1] as shown in Fig. 1. Specifically, for each model of these three categories, its height is first increased or decreased and then it is normalized to unit space to produce new objects with different ratios of length, width and height. To further increase the diversity of the shapes for the mug and bowl category, we additionally perform deformations by enlarging and shrinking the mouths of them. As shown in Fig. 1, There are various shapes of objects in our test set, which are enough to evaluate the effectiveness of our proposed grasp pose estimation method.

3 Effect of Grasp Pose Refinement

Fig. 2. Red grippers are not refined with the antipodal refinement loss L_{anti} and green grippers are refined with L_{anti}

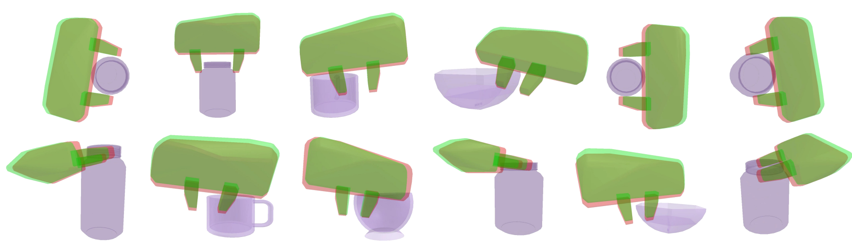

Fig. 3. Red grippers are not refined with the collision avoidance refinement loss $L_{collision}$ and green grippers are refined with $L_{collision}$

We illustrate the contribution of two main loss functions defined in our main paper in Fig. 2 and Fig. 3. Clearly, the antipodal loss L_{anti} that encourages the grasping points to satisfy the antipodal principle guides the gripper to achieve a more stable grasp shown in Fig. 2. Moreover, the green gripper refined with $L_{collision}$ can keep a safer distance from the object than the red one not refined with $L_{collision}$ in Fig. 3, avoiding interpenetration between gripper and object. The two refinement loss functions, together with L_{touch} forcing grasping points on the object surface and L_{reg} avoiding grasp pose beyond the limit, jointly ensure successful grasp. More examples about the grasp poses before and after refinement are shown in Fig. 4.

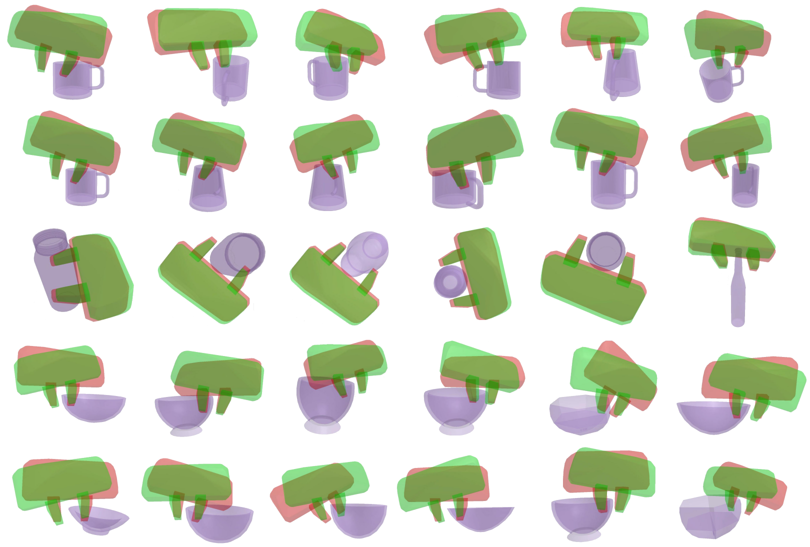

Fig. 4. Qualitative results of the proposed grasp pose refinement module. Red grippers represent the grasp poses before refinement and green grippers represent the grasp poses after refinement

4 Grasp Pose Representation

To connect grasp pose $[R, t]$ with grasping points on the object surface and facilitate the optimization of pose parameters, we also design a new grasp pose representation as $g = (p^1, p^2, d, v)$ where (p^1, p^2) denotes grasp points on the object surface, d the approaching depth and v the approaching vector. In this section, we describe in detail how these two representations are converted to each other.

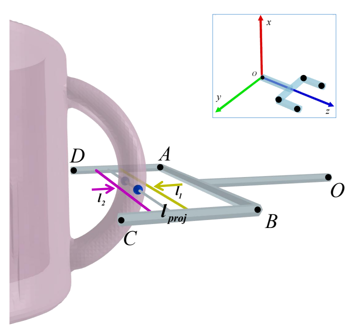

Fig. 5. Process of obtaining grasp points on mug handle surface

The conversion process from $[R, t]$ to g is illustrated in Algorithm 1. We define 4 corner points on the gripper and transform these points to the Object

Coordinates (denoted as $[A, B, C, D]$) using the gripper pose $[R, t]$ as shown in Fig. 5. To obtain the grasp points (p^1, p^2) on the object surface, we firstly define two searching lines l_1 in yellow and l_2 in purple which are parallel to each other with the starting positions coincided with AB and DC respectively. The two lines move towards each other and any line stops when contacting the object surface. After they both stop, their mid-line is then determined as the projection line l_{proj}.Finally, where the projection line l_{proj} intersects the object surface is the grasp points (p^1, p^2). In addition, the approaching vector v is easily derived from $[R, t]$ and the approaching depth d is the distance from gripper origin point O to l_{proj}.

In grasping, grasp pose representation $[R, t]$ is usually used to adjust the gripper, thus we need to convert the estimated grasp pose g back to $[R, t]$ representation. The detailed conversion process is explained in Algorithm 2.

Algorithm 1 Conversion from $[R, t]$ to g

Input: gripper model G, grasp pose matrix $[R, t]$ and object model M
$[O, A, B, C, D] \leftarrow GetCornerPoints(G, R, t)$
$i \leftarrow 0$
repeat
  $E \leftarrow (A * (100 - i) + D * i)/100$
  $F \leftarrow (B * (100 - i) + C * i)/100$
  $i \leftarrow i + 1$
until $IsContact(EF, M)$
$l_1 \leftarrow EF$
$i \leftarrow 0$
repeat
  $E \leftarrow (D * (100 - i) + A * i)/100$
  $F \leftarrow (C * (100 - i) + B * i)/100$
  $i \leftarrow i + 1$
until $IsContact(EF, M)$
$l_2 \leftarrow EF$
$l_{proj} \leftarrow (l_1 + l_2)/2$
$p^1, p^2 \leftarrow GetGraspPoints(l_{proj}, M)$
$v \leftarrow GetApproachVector(R)$
$d \leftarrow GetDistance(O, l_{proj})$
Output: grasp pose $g = (p^1, p^2, d, v)$

Algorithm 2 Conversion from g to $[R, t]$

Input: grasp pose $g = (p^1, p^2, d, v)$
$y_{rot} \leftarrow (p^2 - p^1)/|p^2 - p^1|$
$x_{rot} \leftarrow y_{rot} \times v$
$x_{rot} \leftarrow x_{rot}/|x_{rot}|$
$z_{rot} \leftarrow x_{rot} \times y_{rot}$
$R \leftarrow Concatenate\,(x_{rot}, y_{rot}, z_{rot})$
$t \leftarrow (p^1 + p^2)/2 - z_{rot} * d$
Output: grasp pose matrix $[R, t]$

References

1. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago (2015)
2. Mousavian, A., Eppner, C., Fox, D.: 6-dof graspnet: Variational grasp generation for object manipulation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2901–2910 (2019)
3. ten Pas, A., Gualtieri, M., Saenko, K., Platt, R.: Grasp pose detection in point clouds. The International Journal of Robotics Research **36**(13-14), 1455–1473 (2017)