2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

MonoGraspNet: 6-DoF Grasping with a Single RGB Image

Guangyao Zhai[1,*], Dianye Huang[1,*], Shun-Cheng Wu[1], HyunJun Jung[1],
Yan Di[1,†], Fabian Manhardt[2], Federico Tombari[1,2], Nassir Navab[1,3] and Benjamin Busam[1]

***Abstract*— 6-DoF robotic grasping is a long-lasting but unsolved problem. Recent methods utilize strong 3D networks to extract geometric grasping representations from depth sensors, demonstrating superior accuracy on common objects but performing unsatisfactorily on photometrically challenging objects, e.g., objects in transparent or reflective materials. The bottleneck lies in that the surface of these objects can not reflect accurate depth due to the absorption or refraction of light. In this paper, in contrast to exploiting the inaccurate depth data, we propose the first RGB-only 6-DoF grasping pipeline called *MonoGraspNet* that utilizes stable 2D features to simultaneously handle arbitrary object grasping and overcome the problems induced by photometrically challenging objects. MonoGraspNet leverages a keypoint heatmap and a normal map to recover the 6-DoF grasping poses represented by our novel representation parameterized with 2D keypoints with corresponding depth, grasping direction, grasping width, and angle. Extensive experiments in real scenes demonstrate that our method can achieve competitive results in grasping common objects and surpass the depth-based competitor by a large margin in grasping photometrically challenging objects. To further stimulate robotic manipulation research, we annotate and open-source a multi-view grasping dataset in the real world containing 44 sequence collections of mixed photometric complexity with nearly 20M accurate grasping labels.**

I. Introduction

Humans visually perceive the world with passive RGB views and are able to perform sophisticated interactions with objects even if the objects are unseen previously, translucent, reflective, or transparent. Robotic grasping primarily relies on RGB cameras [1], or active depth sensors such as ToF [2], LiDAR [3], and active stereo [4], to perform arbitrary object grasping using the simplified grasping representation in $SE(2)$ or $SE(3)$ with no prior knowledge such as explicit object models or category information. Existing learning–based approaches can be categorized in two directions, namely planar grasping and 6-DoF grasping. Planar grasping [5, 6, 7, 8] relies on a simple but effective grasping representation, which defines grasps as oriented bounding boxes. Although such a low DoF grasp representation reduces the task to a simple detection problem, it limits the performance in 3D manipulation tasks. 6-DoF grasping enjoys more dexterity than planar grasping, which is suitable for handling complex scenarios. Most 6-DoF grasping methods [9, 10, 11, 12, 13] extract geometric information from dense input and use it to score the grasp candidates. Despite the satisfactory grasping performance on common objects, even if the shape is unknown, the reliance on the dense 3D information makes them vulnerable when the input sensing is unstable, especially on transparent or highly reflective surfaces [14]. Cross-modality fusion approaches for perception tasks can be used to combine RGB images with other sensor inputs [15, 16, 17]. This allows [18, 19] to deal with the aforementioned limitation and shows promising results. Nevertheless, these methods still rely on external depth information as indispensable input, making the methods less convenient and expensive. Motivated by this, we aim to fully explore the ability of the easily accessible and stable RGB modality on the 6-DoF robotic grasping task without any depth input.

Fig. 1. A Franka robot mounting an RGB camera driven by MonoGraspNet is performing transparent object grasping of a glass cup in a household scenario. The crop on the left shows the custom mount for the RGB camera on the gripper and the color overlays (green-blue-red) illustrate the grasping process for this photometrically challenging object.

In this paper, we propose *MonoGraspNet*, the first deep learning pipeline for 6-DoF grasping that requires only an RGB image to estimate accurate grasp poses. MonoGraspNet stacks two parallel networks to respectively predict 2D keypoints and surface normals. Then, it adopts a regression network to estimate keypoint depths, grasping directions, widths, and angles. The sparse setup further encourages accurate grasping property estimation on the cropped regions of interest, as shown in Fig. 2. Our method is thus more economical in training and more lightweight in deployment than those working on dense per-pixel estimation.

Furthermore, despite existing datasets [20, 21, 22, 23, 24, 25, 18, 19], we notice an absence of objects of varying photometric challenges and perfect annotation labels, preventing current methods from being used for truly arbitrary object grasping. Most importantly, previous datasets are

* Contribute equally. † Corresponding author.
[1] Technical University of Munich (TUM), Munich, Germany.
[2] Google.
[3] Johns Hopkins University, Baltimore, MD, USA.

not applicable to high-level tasks, like Robot-Environment Interaction, due to their limited perceptual views and single-scenario configurations. To serve both basic and advanced manipulation tasks, we provide multi-view and multi-scene annotations with large-scale grasping labels based on two public indoor datasets [26, 14], which comprise challenging objects of different photometric complexity. Along with other off-the-shelf dense annotations, the extended dataset can serve better for diverse grasping purposes.

In summary, our main contributions are the followings:

- We propose the first deep learning pipeline for 6-DoF grasping from a single RGB image, designing a novel grasp representation.
- We overcome grasping limitations for photometrically challenging objects (see Fig. 1 for a first impression). Our experiments show that MonoGraspNet can achieve competitive results in grasping common objects and surpass the depth-based competitor by a large margin in grasping photometrically challenging objects.
- We open-source a large-scale grasping dataset comprising objects of varying photometric complexity. The data includes approximately 20M grasping labels for 44 household sequence collections under multiple views[1].

II. Related Work

The purpose of arbitrary/universal grasping is to extract shared representations for all objects in different shapes and categories without prior knowledge. Methods belonging to this field can be roughly divided by the sensor modality and grasp dimensions they use.

RGB-based 3-DoF Grasping: Previous methods using RGB as the input source are called planar grasping. They usually set RGB cameras in a top-down view and treat pose regression as a detection problem [27, 28, 29, 30, 31]. They formulate grasping points in fixed-height oriented rectangles whose width defines the grasping width. Thus, one can use robust and developed 2D detection networks to solve the problem from where oriented rectangles can also be transferred into other formats. GKNet [32] regards one rectangle as three keypoints, which can then be regressed by keypoint-related networks such as CenterNet [33] and CornerNet [34]. They do not introduce depth into the pipeline, so their grasp representation is limited to 3-DoF. In contrast, our method can recover grasping poses in $SE(3)$, although we only require RGB input.

Depth-based 3-DoF Grasping: Methods in this direction represented by Dex-Net 2.0 [20] go further by directly deploying 2D-based networks, like GQ-CNN [20], on depth image processing to perform the grasping task. As the representation is aligned with planar grasping, they still belong to 3-DoF grasping and share the same issue.

Depth-based 6-DoF Grasping: 6-DoF grasping plays a more critical role in the grasping community, allowing robots to plan higher dexterous grasps. Two ways exist in the depth processing: one represented by [35] directly uses a depth image and leverages both 2D detection and accurate depth to estimate the rotated angles and gripper depth to recover 6-DoF poses. However, due to the top-down view and eye-on-hand requirement, some constraints from the pose representation persist, making it unsuitable in practice for some positions. Consistent with planar grasping, it is enough for the bin-picking task, but the performance is limited for more complex manipulations. The other way which has surged to solve the problem is based on point cloud processing represented by a large body of method [13, 36, 10, 9, 37] using 3D backbones [38, 39, 40]. They aim to achieve so-called "AnyGrasp" performance. The success rate of these approaches is high when the depth is reliable. Due to the strong dependency on the depth sensor, the performance drops severely for photometrically challenging objects.

RGB-D based 6-DoF Grasping: Fusing RGB with depth in a completion setup [41] is used to deal with extreme situations and photometrically challenging objects. ClearGrasp [18] masks out the transparent objects in the scene using RGB features and uses estimated normals and edges to reconstruct missing depth. TransCG [19] collects depth using a 3D scanner and IR system, which is used to supervise the network taking fused features of inaccurate depth and RGB image. The improved performance lays the fact that RGB can alleviate the depth-missing problem. Our work takes the idea to another level, where we explore the ability of RGB on 6-DoF grasping without capturing the whole depth map.

III. Methodology

In this work, we propose a novel method, which we dub MonoGraspNet, for estimating the 6-DoF grasp poses from a single monocular image, and a new dataset with multi-view, multi-scene, and large-scale grasping labels.

A. MonoGraspNet

Given an RGB image, our method estimates a set of 6-DoF grasping representations, which are then used to recover 6-DoF poses leveraging a simple conversion. The input image is fed to the Keypoint- and Normal-Network to estimate a set of keypoints represented by a heatmap and a normal map. Those keypoints are then used to crop RGB and normal maps into joint patches, which are used as the input to our DWA-Network for estimating depth, width, and angle (DWA). The system pipeline is illustrated in Figure 2.

Grasp Representation: To better suit our setup, we introduce a novel representation, combining and extending the different representations from Contact-GraspNet [10] and L2G [37] to derive our new formulation (TABLE I). Contact-GraspNet uses the visible grasping point P_1, grasping axis $\mathbf{n_x}$, approaching axis $\mathbf{n_z}$ and grasping width w to represent the 6-DoF grasp, whereas L2G relies on two 3D grasping keypoints P_1, P_2 together with the angle between gripper plane and platform plane ϕ. As recovering sensitive 3D information from 2D input is a difficult ill-posed problem, the pattern we propose in this work is to leverage 2D features to a large extent. Specifically, we split P_1 into a 2d keypoint p aligned with planar grasping represented by [32] and a

[1] https://sites.google.com/view/monograsp/dataset

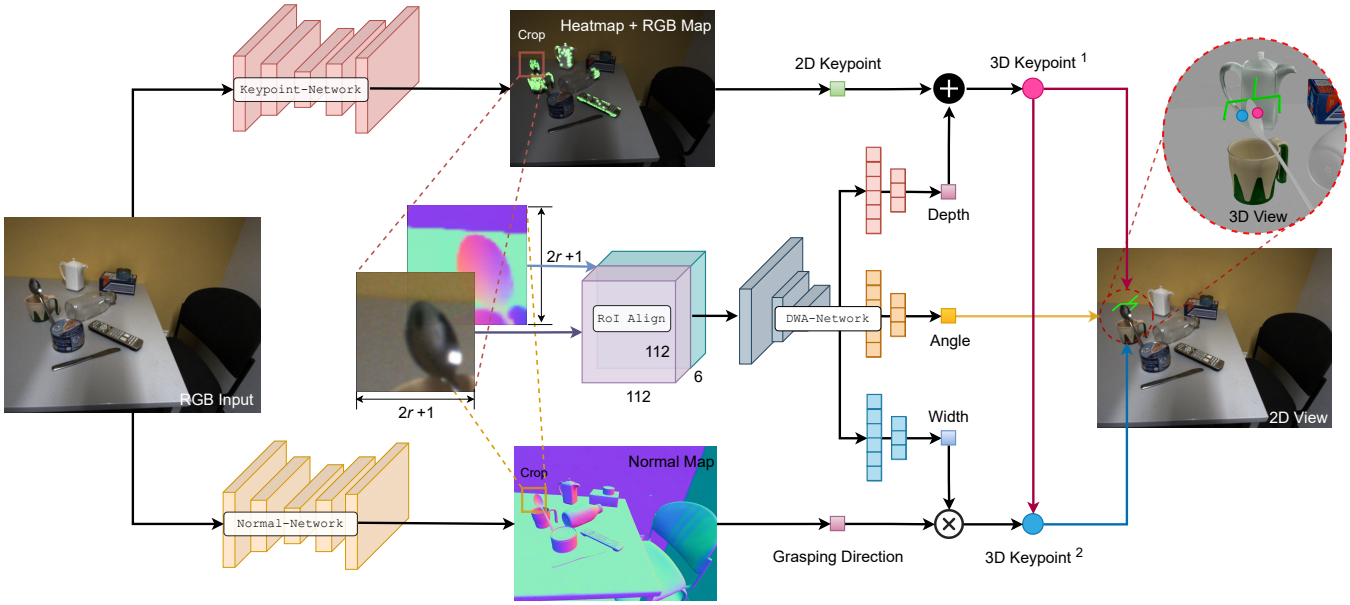

Fig. 2. Schematic overview over our MonoGraspNet pipeline. Given a monocular image, our Keypoint-Network and Normal-Network predict a keypoint heatmap and a normal map. After 2D keypoint selection and local region cropping, the DWA-Network regresses the rest grasping instructions. Exemplary for one keypoint, given the detected keypoint location, we crop the same regions in RGB image and normal map using an adjustable radius r to obtain a $2 \times (2r+1) \times (2r+1) \times 3$ feature map, which we then reshape to $(2r+1) \times (2r+1) \times 6$. Besides, we employ RoI Align to aggregate the features and bring them to a size of $112 \times 112 \times 6$, the same as for the other crops. The DWA-Network utilizes three branches for regressing depth, width, and angle associated with the estimated keypoint. Finally, the visible grasping point (3D Keypoint^1) and invisible grasping point (3D Keypoint^2) can be derived.

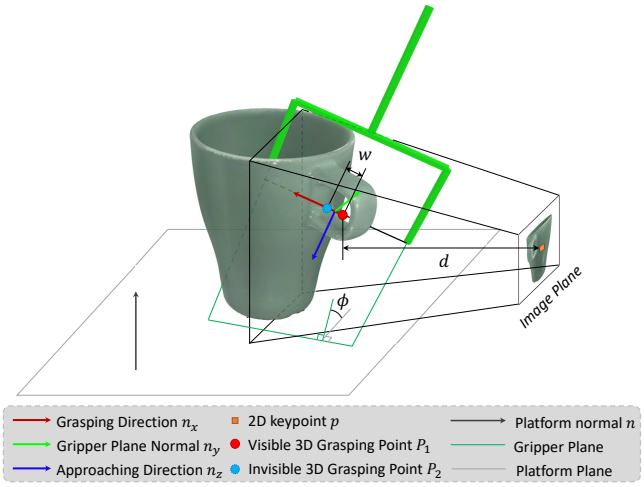

Fig. 3. Our grasp representation. d is the depth value of 2D keypoint p. ϕ is the dihedral angle of gripper plane (green) and platform plane (grey). w is the grasping width calculated as $w = \|P_1 - P_2\|$.

further depth value d, ending up with $\{\mathbf{n_x}, p, d, w, \phi\}$ as our final parameterization, as shown in Fig. 3 (see Sec. III-Bfor 6-DoF pose recovery).

Direction Regression: Robust regression of the grasping axis $\mathbf{n_x}$, which describes the gripper closure direction, is an important component in obtaining our grasp pose. It can be calculated by antipodal sampling through force closure inspection, assuming the availability of an object mesh. For parallel-jaw grippers, force closure depends on the friction cone and the direction of the line connecting two grasping points. Furthermore, the friction cone is determined by the object surface normal $\mathbf{v} \in \mathbb{R}^3$ and the friction coefficient

TABLE I

GRASP REPRESENTATION COMPARISON WITH OTHERS

Methods	Input	#Format
C-GraspNet [10]	Point Cloud	$P_1, \mathbf{n_x}, \mathbf{n_z}, w$
L2G [37]	Point Cloud	P_1, P_2, ϕ
MonoGraspNet (Ours)	RGB	$\mathbf{n_x}, p, d, w, \phi$

μ, which are unknown in the real-world implementation. Nonetheless, the smaller the cosine distance between $\mathbf{n_x}$ and $-\mathbf{v}$, the higher the grasping success rate, regardless of μ. Thus, we can transform the problem of direction regression to the problem of normal estimation, treating $\mathbf{n_x} = -\mathbf{v}^* \approx -\mathbf{v}$, with $\mathbf{v}^*$ being the estimated surface normal. We employ our normal estimation network (Normal-Network), shown in Fig. 2, to infer the normal map. Since the normal consistency around the edges tends to be lower than other parts, grasping can become very complicated for the robot. To alleviate this problem, we use detected 2D keypoints to search for the corresponding normal, as discussed below.

Keypoint Detection: Successful grasping is highly dependent on the accurate prediction of the underlying 3D geometry. Unfortunately, leveraging a state-of-the-art depth estimator is not a promising direction due to the bleeding-out effect around depth discontinuities. It is proved in Sec. IV. Therefore, we instead focus on the utilization of 2D features. In particular, we propose to detect 2D keypoints p first and subsequently recover the associated visible 3D keypoints P_1, as shown in Sec. III-B. Notice that we formulate keypoint detection as heatmap regression using our Keypoint-

Network, as depicted in Fig. 2. Thanks to dense and accurate annotations in our dataset, we observe that ground truth keypoints spread on the smooth surfaces of objects instead of sharp edges. By using estimated keypoints to query surface normals after properly training the network and inspecting the normal consistency around these keypoints, we can overcome the issue mentioned above in the part of normal vector selection.

DWA Regression: In order to obtain the final pose, after having estimated the normal, we extra need to predict depth, width, and angle (DWA). NeWCRFs [42], a Conditional Random Field (CRF) based method, has shown that: (i) the value to be estimated coming from RGB relies on cues such as colors and pixel positions. Further, (ii) the depth of a pixel is usually not determined by distant pixels but by pixels that are only within a certain distance of the target pixel. For a single pixel, too many pairwise connections result in similar performance but require a lot of redundant computations. Based on (i), we additionally improve the idea by adding learned surface normal into consideration:

$$\begin{aligned} E(\mathbf{q}) &= \sum_i E_u\left(q_i\right) + \sum_{ij} E_p\left(q_i, q_j\right) \\ E_p &= f\left(q_i, q_j\right) g\left(I_i, I_j\right) h\left(p_i, p_j\right) k\left(\mathbf{v_i}, \mathbf{v_j}\right), \end{aligned} \tag{1}$$

where $E(\mathbf{q})$ is the energy function for the pixels in the image, q_i is the feature value of pixel i, and j represents the other pixels. The unary potential function E_u uses features of pixel i to calculate its associated energy. Further, the pairwise potential function E_p computes the energy for pairs of pixels, and $f\left(q_i, q_j\right)$ is the pairwise weight. Finally, I_i denotes the RGB value of pixel i. p_i is the coordinate of pixel i and $\mathbf{v_i}$ is the normal vector at p_i. The motivation is that surface normals can provide strong prior information about objects' shapes. For extracting features from photometrically challenging objects, especially transparent ones, color and position are insufficient, as background and transparent parts can look similar in RGB. However, their orientations represented by the surface normals are different. Introducing the normal penalization $k\left(\mathbf{v_i}, \mathbf{v_j}\right)$ into the function can make original $f\left(q_i, q_j\right)$ in [42] reweighted and optimized.

Inspired by (ii), we crop local regions centered on each keypoint to let the feature encoder only focus on the respective regions, as our task only cares about depth estimation for part of the map. Besides, we further extend (ii) to the grasping width and angle regression. In practice, we achieve this by concatenating RGB-normalized crops and normal crops along the feature channel. $f\left(q_i, q_j\right)$ can be represented by various existed feature encoder, such as [43, 44, 45]. Before sending feature crops into the encoder, we perform RoI Align [6] on each crop, as some of the keypoints may appear near the boundaries of RGB images, resulting in these crops with different sizes. Moreover, RoI Align allows flexible feature output adjustment. After obtaining the high-dimension feature map, we employ fully connected layers to regress depth d, grasping width w and rotated angle ϕ for each 2D keypoint.

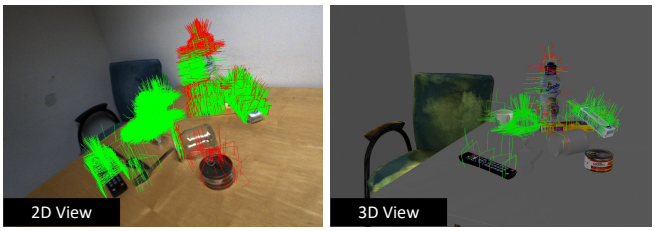

Fig. 4. Examples from one scene in the proposed dataset. Failed grasps are in red, while successful grasps are in green. The first picture shows the dense annotations in the image. The second one demonstrates downsampled 6-DoF grasps in the 3D view for a better look.

Network Selections: In this work, we leverage the state-of-the-art approach [46] for normal prediction and adopt the same loss functions originally proposed, *NLL loss*. For the Keypoint-Network, we use the center-point regression branch in GKNet [32] with DLA [47] as the backbone. We use a variant of the *focal loss* as in [34]. Finally, we employ an attention-based backbone Swin-Transformer Base [45] as the backbone for our DWA-Network, as it has been proven in several regression works to be capable of acting as a very strong feature extractor. To be consistent with Swin-Transformer, we set the input after RoI Align as $112\times112\times6$ and utilize a patch size of 2. The output feature map is of dimension $7 \times 7 \times 1024$, consistent with the original paper. We use *L2 loss* for all three branches in our DWA-Network.

B. 6-DoF Pose Recovery

The visible and invisible 3D keypoints P_1 and P_2 can be obtained by

$$P_1 = dK^{-1}p, \quad P_2 = P_1 - w\mathbf{v}^*, \tag{2}$$

where K is the camera intrinsic matrix. Finally, using P_1, P_2, and ϕ, we can now calculate the 6-DoF grasp pose. Note that ϕ is collected in the robot base frame, with P_1 and P_2 in the camera frame. So we first transform P_1 and P_2 to robot base and calculate the center point P_c^* according to

$$\begin{aligned} P_c^* &= (P_1^* + P_2^*)/2 \\ [P_1^*, P_2^*] &= \mathbf{T}_{base \leftarrow cam}[P_1, P_2]. \end{aligned} \tag{3}$$

We can also obtain each component vector in the rotation matrix for the grasp by:

$$\mathbf{n_x} = -\mathbf{a}^*, \tag{4}$$

$$\begin{cases} \mathbf{n_y} \cdot \mathbf{n_x} = 0 \\ \phi = \arccos\left(\mathbf{n_y} \cdot \mathbf{n}\right) \\ \|\mathbf{n_y}\| = 1 \end{cases}, \tag{5}$$

$$\mathbf{n_z} = \mathbf{n_x} \times \mathbf{n_y}, \tag{6}$$

where $\mathbf{n}$ is the normal of the platform plane $[0, 0, 1]^{\mathrm{T}}$. From (5), two $\mathbf{n_y}$ solutions are obtained by solving a quadratic equation. We always take the one whose accompanying $\mathbf{n_z}$ points to the platform since it is safer and easier for the robot to reach. Then the grasp rotation matrix can be represented by $\mathbf{R} = [\mathbf{n_x}, \mathbf{n_y}, \mathbf{n_z}]$. The final 6-DoF pose $\mathbf{G} = [\mathbf{R}, P_c^*; \mathbf{0}, 1]$ in the robot base frame.

C. Dataset Collection

To explore RGB-only information in this challenging grasping task, we need to learn 3D information from a dataset that contains pixel-perfect depth rather than the measurement coming from a regular depth camera, as it does not work when facing photometrically challenging objects. Moreover, the data should preferably originate from the real world to avoid creating any synthetic-to-real domain gap.

Given this, we choose PhoCaL [26] and HAMMER [14] as our base dataset, which contains various accurate geometry data. Meanwhile, As a robot arm collected both datasets in real-world scenarios, the perception field and manipulation workspace are consistent with the grasping task. By taking advantage of this, we label grasping poses for each mesh in the dataset by antipodal sampling and inspect poses with physical simulators like [48, 49], which is a well-developed routine in [22, 50]. Then we reproject these grasps back into the scene in the robot frame based on 6-DoF object poses $\mathbf{T}_{base \leftarrow obj}$, and perform collision inspection with background meshes. After these procedures, we can obtain grasp labels in each camera frame $\mathbf{G}_{cam}$ by recorded camera poses relative to the robot base $\mathbf{T}_{cam \leftarrow base}$:

$$\begin{aligned} \mathbf{G}_{cam} &= \mathbf{T}_{cam \leftarrow obj} \mathbf{G}_{obj} \\ \mathbf{T}_{cam \leftarrow obj} &= \mathbf{T}_{cam \leftarrow base} \mathbf{T}_{base \leftarrow obj}. \end{aligned} \tag{7}$$

An example is shown in Fig. 4. This extended dataset will be available with original [26] and [14] for the community to research.

As described in Sec. III-A, our pipeline calculates grasps by using visible 2D features from which we extract invisible information. To serve this purpose better, We further search for the nearest 3D points with respect to the grasping points in the point cloud, which can be obtained from depth ground truth with intrinsic camera parameters, and then reproject these points to the image plane to get corresponding pixels. These pixels are treated as the ground truth of visible 2D keypoints. Moreover, we record the correspondence between each keypoint and the corresponding grasping point pair to attain the ground truth of grasping width and grasping angle for each grasp.

IV. Experimental Evaluation

In this section, we first provide our utilized implementation details. Then we introduce our experimental setup and present our results compared with the depth-based state-of-the-art grasping method Contact-GraspNet in various real-world experiments, defined as the ultimate test of the grasp performance in [11].

A. Implementation Details

Training Settings: All the experiments are conducted on a single NVIDIA RTX 3090 GPU with Adam optimizer. The Keypoint-Network is trained for 30 epochs with an initial learning rate of 1e-4 and batch size of 2. For Normal-Network training, we follow the instructions in the original paper [46]. The proposed DWA-Network is trained for 30 epochs with an initial learning rate to be 1.25e-4 and batch size to be 32. We decrease the learning rate by 10 times at the 15th and the 20th epoch. Our input image size is 832×1088.

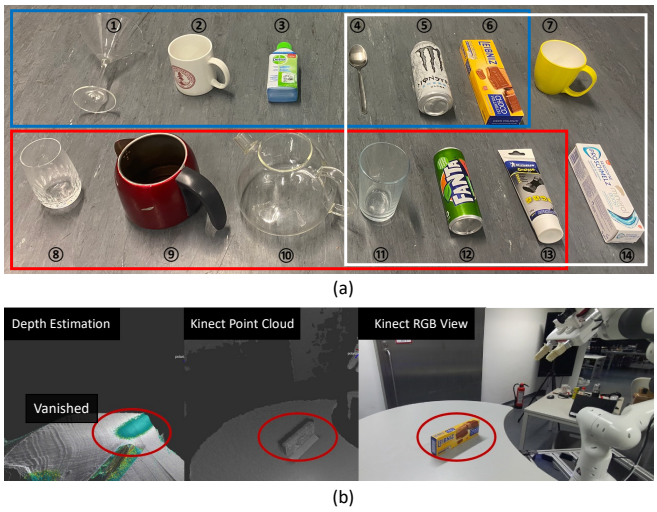

Fig. 5. a) Objects used for testing. 1-6 are familiar objects (blue block). 8-13 are unfamiliar objects (red block). Objects in white block are used for clutter removal. b) Recovered point cloud from monocular depth estimation is out of the shape, resulting in the biscuit box vanished from the scene.

Robot Hardware: We use a 7-DoF Franka Panda robot with a parallel-jaw gripper as the end-effector. The RGB camera mounted on the gripper base is a Phoenix 5.0 MP Polarization camera, the same as the one used in [26, 14] shown in Fig. 1. For testing Contact-GraspNet, we install an Azure Kinect depth camera beside the robot. All cameras are hand-eye calibrated.

B. Experimental Details

Evaluation Protocols: We select 14 objects shown in Fig. 5.a) to compare our method with Contact-GraspNet and report the *success rate* and *completion rate*. The success rate is used for testing single-object grasping in seen and unseen scenarios. We let the robot execute 15 grasps by putting the objects at three random positions in the scene, and then we calculate successful grasps. The completion rate is the percentage of objects removed from the clutter, which can be a robust metric when testing multi-object grasping. We execute the grasping task in four scenarios with different configurations sorted by ascending difficulty: 1) familiar objects in seen scenes, 2) familiar objects in unseen scenes, 3) unfamiliar objects in seen scenes, and 4) unfamiliar objects in unseen scenes. We additionally set an experiment by combining Contact-GraspNet and NeWCRFs [42] to support our claim in Sec. III-A-Keypoint Detection, which is that the path of estimating dense depth map and grasping objects using the recovered point cloud results in bad performance.

Running Pipeline: In our experiment, we directly use basic joint control instead of advanced options that require 3D information for the motion planning part. To ensure impartiality, we apply the same planning method for Contact-GraspNet. Since there is no segmentation module inside MonoGraspNet, we also remove the relevant module [51]

TABLE II

SUCCESS RATE (%) OF FAMILIAR SINGLE OBJECT GRASPING EXPERIMENTS IN SEEN AND UNSEEN SCENES

Method	**Avg.**		(1) Cocktail glass		(2) Mug		(3) Liquid		(4) Spoon		(5) Monster		(6) Biscuit	
	S	Un-S	S	Un-S	S	Un-S	S	Un-S	S	Un-S	S	Un-S	S	Un-S
[42]+C-GraspNet [10]	23.3	10.0	0	0	0	13.3	33.3	26.7	6.7	0	26.7	0	73.3	20.0
C-GraspNet [10]	62.2	60.0	6.7	0	**93.3**	**93.3**	**60.0**	**66.7**	20.0	6.7	**93.3**	**93.3**	**100**	**100**
MonoGraspNet (Ours)	**83.9**	**72.2**	**80.0**	**60.0**	86.7	86.7	**60.0**	53.3	**86.7**	**80.0**	86.7	73.3	93.3	80.0

TABLE III

SUCCESS RATE (%) OF UNFAMILIAR SINGLE OBJECT GRASPING EXPERIMENTS IN SEEN AND UNSEEN SCENES

Method	**Avg.**		(8) Narrow glass		(9) Red kettle		(10) Trans-kettle		(11) Glass		(12) Fanta		(13) Gel	
	S	Un-S	S	Un-S	S	Un-S	S	Un-S	S	Un-S	S	Un-S	S	Un-S
[42]+C-GraspNet [10]	26.2	11.9	0	0	30.7	13.3	0	0	13.3	6.7	40.0	20.0	73.3	40.0
C-GraspNet [10]	42.2	42.2	0	0	**86.7**	80.0	6.7	0	0	0	**93.3**	**100**	66.7	73.3
MonoGraspNet (Ours)	**75.5**	**72.2**	**60.0**	**40.0**	80.0	**86.7**	**60.0**	**53.3**	**80.0**	**80.0**	86.7	93.3	**80.0**	**80.0**

TABLE IV

COMPLETION RATE OF CLUTTER REMOVAL EXPERIMENTS

Type	C-GraspNet [10]		MonoGraspNet (Ours)	
	1st trial	2nd trial	1st trial	2nd trial
4-object scene	3/4	3/4	3/4	3/4
5-object scene	4/5	4/5	4/5	**5/5**

TABLE V

SUCCESS RATE (%) COMPARISON WITH DIFFERENT DEPTH ESTIMATION

Method	Normal		Challenging	
	Fanta	Biscuit	Spoon	Glass
DWA-Net w/o normal	80.0	93.3	86.7	66.7
DWA-Net	**86.7**	93.3	86.7	**80.0**

accompanying Contact-GraspNet, which is allowed by the instruction from [10].

Results TABLE II and TABLE III show the comparison results under the four situations. *Familiar objects:* trained objects, but in untrained views. *Unfamiliar objects:* untrained objects. *S:* scenes that were seen in the training set but in a novel view. *Un-S:* unseen scenes. As shown in Row 1, directly feeding the point cloud from [42] into Contact-GraspNet yields the worst results, with only 23.3% success rate for familiar objects in seen scenes, as the recovered point cloud is distorted in an unstructured shape. An example is shown in Fig. 5.b). In TABLE II, it can be seen that Contact-GraspNet is slightly superior to MonoGraspNet when grasping normal objects, i.e., objects (2, 5, 6), with an average of 6.6% leap forward. This seemingly discouraging outcome may be expected since depth from sensors is typically more accurate than predicted depth from RGB images. However, when dealing with photometrically challenging objects (1, 3, 4), MonoGraspNet surpasses Contact-GraspNet by a large margin, with an average success rate of 78.9% against 28.9%. In this situation, inaccurate depth limits the performance of grasp pose estimation of Contact-GraspNet.

We also verified the generalization ability of MonoGraspNet. By comparing the results in unseen scenes with the ones in seen scenes, we can see that the success rate remains stable and consistent, with only a maximum 11.7% decrease in average. It proves that MonoGraspNet can generalize to other real-world scenarios without pre-training or fine-tuning. For unfamiliar object grasping in TABLE III, even though the textures or shapes of these objects are distinct from the trained ones in the dataset, MonoGraspNet can still yield an adequate success rate, as it uses local 2D features to infer grasps, which remains stable across different objects and different scenes.

We further perform a more challenging clutter removal task. As we do not have a collision check module, we always let the robot execute the nearest grasp. We randomly choose four or five objects with one challenging object inside from the white block of Fig. 5. TABLE IV shows that we still show a competitive performance compared to Contact-GraspNet when moving simple clutter with photometrically challenging objects inside.

C. Ablation Study

To prove the motivation that introducing the normal map can help depth estimation in DWA-Network. We re-train a DWA-Network with RGB crops as the only input. TABLE V shows that the success rate performance of grasping normal objects and even the reflective object (Spoon as an example here) are similar, but the improvement is obvious for grasping transparent objects (unfamiliar Glass).

V. CONCLUSIONS

In contrast to previous conventional routines which rely on accurate depth, this paper proposes the first RGB-only 6-DoF grasping pipeline called MonoGraspNet. It utilizes stable 2D features to simultaneously handle arbitrary object grasping and overcome the problems induced by photometrically challenging objects. Given an RGB image, our method estimates a keypoint map and a normal map to recover the 6-DoF grasping poses represented by our novel representation. The experiments demonstrate that our method can achieve competitive results on grasping common objects with simple textures while surpassing the depth-based competitor significantly on grasping photometrically challenging objects. We additionally annotate a large-scale real-world grasping dataset, containing 44 object settings of mixed photometric complexity with approximately 20M accurate grasping labels.

REFERENCES

[1] D. Guo, T. Kong, F. Sun, and H. Liu, "Object discovery and grasp detection with a shared convolutional neural network," in *ICRA*, 2016.

[2] A. Maldonado, U. Klank, and M. Beetz, "Robotic grasping of unmodeled objects using time-of-flight range data and finger torque information," in *IROS*, 2010.

[3] K. M. Popek, M. S. Johannes, K. C. Wolfe, R. A. Hegeman, J. M. Hatch, J. L. Moore, K. D. Katyal, B. Y. Yeh, and R. J. Bamberger, "Autonomous grasping robotic aerial system for perching (agrasp)," in *IROS*, 2018.

[4] R. B. Rusu, A. Holzbach, R. Diankov, G. Bradski, and M. Beetz, "Perception for mobile manipulation and grasping using active stereo," in *Humanoids*, 2009.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *NeurIPS*, 2015.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017.

[7] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *CVPR*, 2018.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016.

[9] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, "Pointnetgpd: Detecting grasp configurations from point sets," in *ICRA*, 2019.

[10] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in *ICRA*, 2021.

[11] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *ICCV*, 2019.

[12] C. Wu, J. Chen, Q. Cao, J. Zhang, Y. Tai, L. Sun, and K. Jia, "Grasp proposal networks: an end-to-end solution for visual learning of robotic grasps," in *NeurIPS*, 2020.

[13] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *CVPR*, 2020.

[14] H. Jung, P. Ruhkamp, G. Zhai, N. Brasch, Y. Li, Y. Verdie, J. Song, Y. Zhou, A. Armagan, S. Ilic, A. Leonardis, N. Navab, and B. Busam, "On the importance of accurate geometry priors for dense 3d vision tasks," in *CVPR*, 2023.

[15] A. Lopez-Rodriguez, B. Busam, and K. Mikolajczyk, "Project to adapt: Domain adaptation for depth completion from noisy and sparse sensor data," in *ACCV*, 2020.

[16] S. Gasperini, P. Koch, V. Dallabetta, N. Navab, B. Busam, and F. Tombari, "R4dyn: Exploring radar for self-supervised monocular depth estimation of dynamic scenes," in *3DV*, 2021.

[17] Y. Verdié, J. Song, B. Mas, B. Busam, A. Leonardis, and S. McDonagh, "Cromo: Cross-modal learning for monocular depth estimation," in *CVPR*, 2022.

[18] S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song, "Clear grasp: 3d shape estimation of transparent objects for manipulation," in *ICRA*, 2020.

[19] 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," *RA-L*, vol. 7, no. 3, 2022.

[20] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *RSS*, 2017.

[21] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *IJRR*, vol. 37, no. 4-5, 2018.

[22] C. Eppner, A. Mousavian, and D. Fox, "Acronym: A large-scale grasp dataset based on simulation," in *ICRA*, 2021.

[23] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *IROS*, 2018.

[24] (2017) Cornell grasping dataset. [Online]. Available: http://pr.cs.cornell.edu/grasping/rect_data/data.php

[25] G. Zhai, Y. Zheng, Z. Xu, X. Kong, Y. Liu, B. Busam, Y. Ren, N. Navab, and Z. Zhang, "Da^2 dataset: Toward dexterity-aware dual-arm grasping," *RA-L*, vol. 7, no. 4, 2022.

[26] P. Wang, H. Jung, Y. Li, S. Shen, R. P. Srikanth, L. Garattoni, S. Meier, N. Navab, and B. Busam, "Phocal: A multi-modal dataset for category-level object pose estimation with photometrically challenging objects," in *CVPR*, 2022.

[27] D. Guo, F. Sun, H. Liu, T. Kong, B. Fang, and N. Xi, "A hybrid deep architecture for robotic grasp detection," in *ICRA*, 2017.

[28] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," in *IROS*, 2017.

[29] U. Asif, J. Tang, and S. Harrer, "Graspnet: An efficient convolutional neural network for real-time grasp detection for low-powered devices." in *IJCAI*, 2018.

[30] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *ICRA*, 2015.

[31] X. Zhou, X. Lan, H. Zhang, Z. Tian, Y. Zhang, and N. Zheng, "Fully convolutional grasp detection network with oriented anchor box," in *IROS*, 2018.

[32] R. Xu, F.-J. Chu, and P. A. Vela, "Gknet: grasp keypoint network for grasp candidates detection," *IJRR*, vol. 41, no. 4, 2022.

[33] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *ICCV*, 2019.

[34] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *ECCV*, 2018.

[35] X. Zhu, L. Sun, Y. Fan, and M. Tomizuka, "6-dof contrastive grasp proposal network," in *ICRA*, 2021.

[36] M. Breyer, J. J. Chung, L. Ott, S. Roland, and N. Juan, "Volumetric grasping network: Real-time 6 dof grasp detection in clutter," in *CoRL*, 2020.

[37] A. Alliegro, M. Rudorfer, F. Frattin, A. Leonardis, and T. Tommasi, "End-to-end learning to grasp via sampling from object point clouds," *RA-L*, vol. 7, no. 4, 2022.

[38] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *CVPR*, 2017.

[39] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *NeurIPS*, 2017.

[40] A. Alliegro, D. Valsesia, G. Fracastoro, E. Magli, and T. Tommasi, "Denoise and contrast for category agnostic shape completion," in *CVPR*, 2021.

[41] H. Jung, N. Brasch, A. Leonardis, N. Navab, and B. Busam, "Wild tofu: Improving range and quality of indirect time-of-flight depth with rgb fusion in challenging environments," in *3DV*, 2021.

[42] W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan, "Newcrfs: Neural window fully-connected crfs for monocular depth estimation," in *CVPR*, 2022.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[44] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[45] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *ICCV*, 2021.

[46] G. Bae, I. Budvytis, and R. Cipolla, "Estimating and exploiting the aleatoric uncertainty in surface normal estimation," in *ICCV*, 2021.

[47] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *CVPR*, 2018.

[48] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance gpu-based physics simulation for robot learning," 2021. [Online]. Available: https://arxiv.org/abs/2108.10470

[49] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *RA-M*, vol. 11, no. 4, 2004.

[50] C. Eppner, A. Mousavian, and D. Fox, "A billion ways to grasps - an evaluation of grasp sampling schemes on a dense, physics-based grasp data set," in *ISRR*, 2019.

[51] Y. Xiang, C. Xie, A. Mousavian, and D. Fox, "Learning rgb-d feature embeddings for unseen object instance segmentation," in *CoRL*, 2021.