DexYCB: A Benchmark for Capturing Hand Grasping of Objects

Yu-Wei Chao[1] Wei Yang[1] Yu Xiang[1] Pavlo Molchanov[1] Ankur Handa[1] Jonathan Tremblay[1]
Yashraj S. Narang[1] Karl Van Wyk[1] Umar Iqbal[1] Stan Birchfield[1] Jan Kautz[1] Dieter Fox[1,2]

[1]NVIDIA, [2]University of Washington
{ychao,weiy,yux,pmolchanov,ahanda,jtremblay,ynarang,kvanwyk,uiqbal,sbirchfield,jkautz,dieterf}@nvidia.com

Figure 1: Two captures (left and right) from the DexYCB dataset. In each case, the top row shows color images simultaneously captured from three views, while the bottom row shows the ground-truth 3D object and hand pose rendered on the darkened captured images.

Abstract

We introduce DexYCB, a new dataset for capturing hand grasping of objects. We first compare DexYCB with a related one through cross-dataset evaluation. We then present a thorough benchmark of state-of-the-art approaches on three relevant tasks: 2D object and keypoint detection, 6D object pose estimation, and 3D hand pose estimation. Finally, we evaluate a new robotics-relevant task: generating safe robot grasps in human-to-robot object handover.[1]

1. Introduction

3D object pose estimation and 3D hand pose estimation are two important yet unsolved vision problems. Traditionally, these two problems have been addressed separately, yet in many critical applications, we need both capabilities working together [28, 5, 12]. For example, in robotics, a reliable motion capture for hand manipulation of objects is crucial for both learning from human demonstration [11] and fluent and safe human-robot interaction [35].

State-of-the-art approaches for both 3D object pose [29, 19, 26, 23, 32, 22, 18] and 3D hand pose estimation [37, 20, 17, 1, 8, 13, 25] rely on deep learning and thus require large datasets with labeled hand or object poses for training. Many datasets [36, 37, 15, 16] have been introduced in both domains and have facilitated progress on these two problems in parallel. However, since they were introduced for either task separately, many of them do not contain interaction of hands and objects, i.e., static objects without humans in the scene, or bare hands without interacting with objects. In the presence of interactions, the challenge of solving the two tasks together not only doubles but multiplies, due to the motion of objects and mutual occlusions incurred by the interaction. Networks trained on either of the datasets will thus not generalize well to interaction scenarios.

Creating a dataset with accurate 3D pose of hands and objects is also challenging for the same reasons. As a result, prior works have attempted to capture accurate hand motion either with specialized gloves [9], magnetic sensors [36, 7], or marker-based mocap systems [2, 27]. While they can achieve unparalleled accuracy, the introduction of hand-attached devices may be intrusive and thus bias the naturalness of hand motion. It also changes the appearance of hands and thus may cause issues with generalization.

Due to the challenge of acquiring real 3D poses, there has been an increasing interest in using synthetic datasets to train pose estimation models. The success has been notable on object pose estimation. Using 3D scanned object models and photorealistic rendering, prior work [15, 26, 30, 4, 16] has generated synthetic scenes of objects with high fidelity in appearance. Their models trained only on synthetic data

[1]Dataset and code available at https://dex-ycb.github.io.

can thus translate to real images. Nonetheless, synthesizing hand-object interactions remains challenging. One problem is to synthesize realistic grasp poses for generic objects [3]. Furthermore, synthesizing natural looking human motions is still an active research area in graphics.

In this paper, we focus on marker-less data collection of real hand interaction with objects. We take inspiration from recent work [10] and build a multi-camera setup that records interactions synchronously from multiple views. Compared to the recent work, we instrument the setup with more cameras and configure them to capture a larger workspace that allows our human subjects to interact freely with objects. In addition, our pose labeling process utilizes human annotation rather than automatic labeling. We crowdsource the annotation so that we can efficiently scale up the data labeling process. Given the setup, we construct a large-scale dataset that captures a simple yet ubiquitous task: grasping objects from a table. The dataset, DexYCB, consists of 582K RGB-D frames over 1,000 sequences of 10 subjects grasping 20 different objects from 8 views (Fig. 1).

Our contributions are threefold. First, we introduce a new dataset for capturing hand grasping of objects. We empirically demonstrate the strength of our dataset over a related one through cross-dataset evaluation. Second, we provide in-depth analysis of current approaches thoroughly on three relevant tasks: 2D object and keypoint detection, 6D object pose estimation, and 3D hand pose estimation. To the best of our knowledge, our dataset is the first that allows joint evaluation of these three tasks. Finally, we demonstrate the importance of joint hand and object pose estimation on a new robotics relevant task: generating safe robot grasps for human-to-robot object handover.

2. Constructing DexYCB

2.1. Hardware Setup

In order to construct the dataset, we built a multi-camera setup for capturing human hands interacting with objects. A key design choice was to enable a sizable capture space, where a human subject can freely interact and perform tasks with multiple objects. Our multi-camera setup is shown in Fig. 2. We use 8 RGB-D cameras (RealSense D415) and mount them such that collectively they can capture a tabletop workspace with minimal blind spots. The cameras are extrinsically calibrated and temporally synchronized. For data collection, we stream and record all 8 views together at 30 fps with both color and depth of resolution 640×480.

2.2. Data Collection and Annotation

Given the setup, we record videos of hands grasping objects. We use 20 objects from the YCB-Video dataset [34], and record multiple trials from 10 subjects. For each trial, we select a target object with 2 to 4 other objects and place

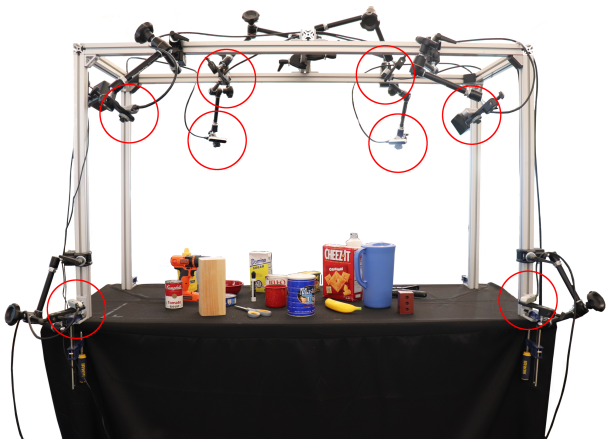

Figure 2: Our setup with 8 RGB-D cameras (red circle).

them on the table. We ask the subject to start from a relaxed pose, pick up the target object, and hold it in the air. We also ask some subjects to pretend to hand over the object to someone across from them. We record for 3 seconds, which is sufficient to contain the full course of action. For each target object, we repeat the trial 5 times, each time with a random set of accompanied objects and placement. We ask the subject to perform the pick-up with the right hand in the first two trials, and with the left hand in the third and fourth trials. In the fifth trial, we randomize the choice. We rotate the target among all 20 objects. This gives us 100 trials per subject, and 1,000 trials in total for all subjects.

To acquire accurate ground-truth 3D pose for hands and objects, our approach (detailed in Sec. 2.3) relies on 2D keypoint annotations for hands and objects in each view. To ensure accuracy, we label the required keypoints in RGB sequences fully through human annotation. Our annotation tool is based on VATIC [31] for efficient annotation of videos. We set up annotation tasks on the Amazon Mechanical Turk (MTurk) and label every view in all the sequences.

For hands, we adopt 21 pre-defined hand joints as our keypoints (3 joints plus 1 tip for each finger and the wrist). We explicitly ask the annotators to label and track these joints throughout a given video sequence. The annotators are also asked to mark a keypoint as invisible in a given frame when it is occluded.

Pre-defining keypoints exhaustively for every object would be laborious and does not scale as the number of objects increases. Our approach (Sec. 2.3) explicitly addresses this issue by allowing user-defined keypoints. Specifically, given a video sequence in a particular view, we first ask the annotator to find 2 distinctive landmark points that are easily identified and trackable on a designated object, and we ask them to label and track these points throughout the sequence. We explicitly ask the annotators to find keypoints that are visible most of the time, and mark a keypoint as invisible whenever it is occluded.

2.3. Solving 3D Hand and Object Pose

To represent 3D hand pose, we use the popular MANO hand model [24]. The model represents a right or left hand with a deformable triangular mesh of 778 vertices. The mesh is parameterized by two low-dimensional embeddings (θ, β), where $\theta \in \mathbb{R}^{51}$ accounts for variations in pose (i.e. articulation) and $\beta \in \mathbb{R}^{10}$ in shape. We use the version from [13], which implements MANO as a differentiable layer in PyTorch that maps (θ, β) to the mesh together with the 3D positions of 21 hand joints defined in the keypoint annotation. We pre-calibrate the hand shape β for each subject and fix it throughout each subject's sequences.

Since our objects from YCB-Video [34] also come with texture-mapped 3D mesh models, we use the standard 6D pose representation [15, 16] for 3D object pose. The pose of each object is represented by a matrix $T \in \mathbb{R}^{3\times4}$ composed of a 3D rotation matrix and a 3D translation vector.

To solve for hand and object pose, we formulate an optimization problem similar to [38, 10] by leveraging depth and keypoint annotations from all views and multi-view geometry. For a given sequence with N_H hands and N_O objects, we denote the overall pose at a given time frame by $P = (P_H, P_O)$, where $P_H = \{\theta_h\}_{h=1}^{N_H}$ and $P_O = \{T_o\}_{o=1}^{N_O}$. We define the pose in world coordinates where we know the extrinsics of each camera. Then at each time frame, we solve the pose by minimizing the following energy function:

$$E(P) = E_{\text{depth}}(P) + E_{\text{kpt}}(P) + E_{\text{reg}}(P). \tag{1}$$

Depth The depth term E_{depth} measures how well the models given poses explain the observed depth data. Let $\{d_i \in \mathbb{R}^3\}_{i=1}^{N_D}$ be the total point cloud merged from all views after transforming to the world coordinates, with N_D denoting the number of points. Given a pose parameter, we denote the collection of all hand and object meshes as $\mathcal{M}(P) = (\{\mathcal{M}_h(\theta_h)\}, \{\mathcal{M}_o(T_o)\})$. We define the depth term as

$$E_{\text{depth}}(P) = \frac{1}{N_D}\sum_{i=1}^{N_D} |\text{SDF}(d_i, \mathcal{M}(P))|^2, \tag{2}$$

where $\text{SDF}(\cdot)$ calculates the signed distance value of a 3D point from a triangular mesh in mm. While E_{depth} is differentiable, calculating E_{depth} and also the gradients is computationally expensive for large point clouds and meshes with a huge number of vertices. Therefore, we use an efficient point-parallel GPU implementation for it.

Keypoint The keypoint term E_{kpt} measures the reprojection error of the keypoints on the models with the annotated keypoints, and can be decomposed by hand and object:

$$E_{\text{kpt}}(P) = E_{\text{kpt}}(P_H) + E_{\text{kpt}}(P_O). \tag{3}$$

For hands, let $J_{h,j}$ be the 3D position of joint j of hand h in the world coordinates, $p^c_{h,j}$ be the annotation of the same joint in the image coordinates of view c, and $\gamma^c_{h,j}$ be its visibility indicator. The energy term is defined as

$$E_{\text{kpt}}(P_H) = \frac{1}{\sum \gamma^c_{h,j}} \sum_{c=1}^{N_C}\sum_{h=1}^{N_H}\sum_{j=1}^{N_J} \gamma^c_{h,j} ||\text{proj}^c(J_{h,j}) - p^c_{h,j}||_2^2, \tag{4}$$

where $\text{proj}^c(\cdot)$ returns the projection of a 3D point onto the image plane of view c, and $N_C = 8$ and $N_J = 21$.

For objects, recall that we did not pre-define keypoints for annotation, but rather asked annotators to select distinctive points to track. Here, we assume an accurate initial pose is given at the first frame where an object's keypoint is labeled visible. We then map the selected keypoint to a vertex on the object's 3D model by back-projecting the keypoint's position onto the object's visible surface. We fix that mapping afterwards. Let $K^c_{o,k}$ be the 3D position of the selected keypoint k of object o in view c in world coordinates. Similar to Eq. (4), with $N_K = 2$, the energy term is

$$E_{\text{kpt}}(P_O) = \frac{1}{\sum \gamma^c_{o,k}} \sum_{c=1}^{N_C}\sum_{o=1}^{N_O}\sum_{k=1}^{N_K} \gamma^c_{o,k} ||\text{proj}^c(K^c_{o,k}) - p^c_{o,k}||_2^2. \tag{5}$$

To ensure an accurate initial pose for keypoint mapping, we initialize the pose in each time frame with the solved pose from the last time frame. We initialize the pose in the first frame by running PoseCNN [34] on each view and select an accurate pose for each object manually.

Regularization Following [21, 12], we add an ℓ_2 regularization to the low-dimensional pose embedding of MANO to avoid irregular articulation of hands:

$$E_{\text{reg}}(P) = \frac{1}{N_H}\sum_{h=1}^{N_H} ||\theta_h||_2^2. \tag{6}$$

To minimize Eq. (1), we use the Adam optimizer with a learning rate of 0.01. For each time frame, we initialize the pose P with the solved pose from the last time frame and run the optimizer for 100 iterations.

3. Evaluation Setup

To evaluate different scenarios, we generate train/val/test splits in four different ways (referred to as "setup"):

- **S0 (default).** The train split contains all 10 subjects, all 8 camera views, and all 20 grasped objects. Only the sequences are not shared with the val/test split.
- **S1 (unseen subjects).** The dataset is split by subjects (train/val/test: 7/1/2).
- **S2 (unseen views).** The dataset is split by camera views (train/val/test: 6/1/1).

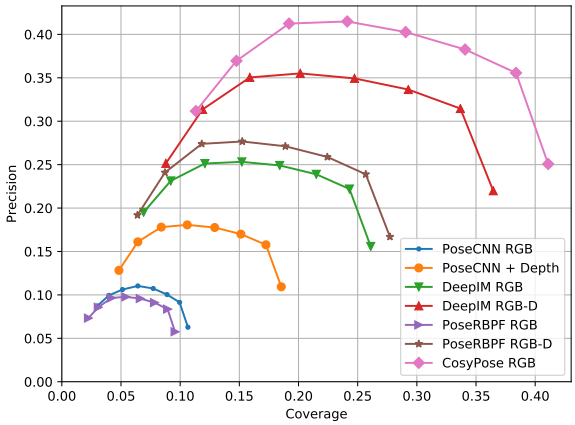

Figure 3: Precision-coverage curves for grasp generation on S1.

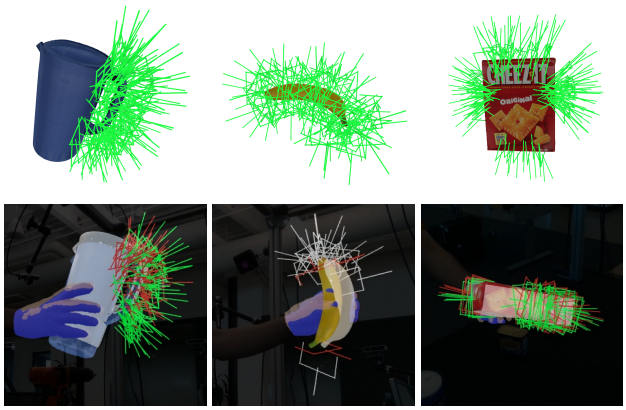

Figure 4: Top: 100 successful grasps sampled from [6]. Bottom: predicted grasps generated by the predicted object pose (textured model) and hand segmentation (blue masks). Green ones denote those covering successful grasps, red ones denote those collided with the object or hand, and gray ones are failures not covering any successful grasps in the reference set. Ground-truth objects and hands are shown in translucent white and brown meshes.

- **S3 (unseen grasping).** The dataset is split by grasped objects (train/val/test: 15/2/3). Objects being grasped in the test split are never being grasped in the train/val split, but may appear static on the table. This way the training set still contain examples of each object.

4. Safe Human-to-Robot Object Handover

Task Given an RGB-D image with a person holding an object, the goal is to generate a diverse set of robot grasps to take over the object without pinching the person's hand (we refer to as "safe" handovers). The diversity of grasps is important since not all the grasps are kinematically feasible for execution. We assume a parallel-jaw Franka Panda gripper and represent each grasp as a point in $SE(3)$.

Evaluation We first sample 100 grasps for each YCB object using farthest point sampling from a diverse set of grasps pre-generated for that object in [6]. This ensures a dense coverage of the pose space (Fig. 4). For each image, we transform these grasps from the object frame to camera frame using the ground-truth object pose, and remove those collided with the ground-truth object and hand mesh. This generates a reference set of successful grasps $\mathcal{R}$.

Given a set of predicted grasps χ, we evaluate its diversity by computing the *coverage* [6] of $\mathcal{R}$, defined by the percentage of grasps in $\mathcal{R}$ having at least one matched grasp in χ that is neither collided with the object nor the hand. Specifically, two grasps g, h are considered matched if $|g_t - h_t| < \sigma_t$ and $\arccos(|\langle g_q, h_q \rangle|) < \sigma_q$, where g_t is the translation and g_q is the orientation in quaternion. We use $\sigma_t = 0.05$ m and $\sigma_q = 15°$.

One could potentially hit a high coverage by sampling grasps exhaustively. Therefore we also compute *precision*, defined as the percentage of grasps in χ that have at least one matched successful grasp in $\mathcal{R}$.

Baseline We experiment with a simple baseline that only requires hand segmentation and 6D object pose. Similar to constructing $\mathcal{R}$, we transform the 100 grasps to the camera frame but using the estimated object pose, then remove those that are collided with the hand point cloud obtained by the hand segmentation and the depth image. Specifically, a grasp is collided if the distance of a pair of points from the gripper point cloud and the hand point cloud is less than a threshold ϵ. The gripper point cloud is obtained from a set of pre-sampled points on the gripper surface. We use the hand segmentation results from Mask R-CNN [14, 33].

Results We evaluate grasps generated with different object pose methods at different threshold $\epsilon \in [0, 0.07\text{ m}]$ and show the precision-coverage curves on S1 in Fig. 3. We see that better object pose estimation leads to better grasp generation. Fig. 4 shows qualitative examples of the predicted grasps. We see that most of the failure grasps (red and gray) are due to inaccurate object pose. Some are hand-colliding grasps caused by a miss detected hand when the hand is partially occluded by the object (e.g., "003_cracker_box"). This can be potentially addressed by model based approaches that directly predict the full hand shape.

References

[1] Adnane Boukhayma, Rodrigo de Bem, and Philip H.S. Torr. 3D hand shape and pose from images in the wild. In *CVPR*, 2019. 1

[2] Samarth Brahmbhatt, Chengcheng Tang, Christopher D. Twigg, Charles C. Kemp, and James Hays. ContactPose: A dataset of grasps with object contact and hand pose. In *ECCV*, 2020. 1

[3] Enric Corona, Albert Pumarola, Guillem Alenyà, Francesc Moreno-Noguer, and Grégory Rogez. GanHand: Predicting

human grasp affordances in multi-object scenes. In *CVPR*, 2020. 2

[4] Xinke Deng, Arsalan Mousavian, Yu Xiang, Fei Xia, Timothy Bretl, and Dieter Fox. PoseRBPF: A Rao-Blackwellized particle filter for 6D object pose estimation. In *RSS*, 2019. 1

[5] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J. Crandall. HOPE-Net: A graph-based model for hand-object pose estimation. In *CVPR*, 2020. 1

[6] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. A billion ways to grasps: An evaluation of grasp sampling schemes on a dense, physics-based grasp data set. In *ISRR*, 2019. 4

[7] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In *CVPR*, 2018. 1

[8] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3D hand shape and pose estimation from a single RGB image. In *CVPR*, 2019. 1

[9] Oliver Glauser, Shihao Wu, Daniele Panozzo, Otmar Hilliges, and Olga Sorkine-Hornung. Interactive hand pose estimation using a stretch-sensing soft glove. In *SIGGRAPH*, 2019. 1

[10] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. HOnnotate: A method for 3D annotation of hand and object poses. In *CVPR*, 2020. 2, 3

[11] Ankur Handa, Karl Van Wyk, Wei Yang, Jacky Liang, Yu-Wei Chao, Qian Wan, Stan Birchfield, Nathan Ratliff, and Dieter Fox. DexPilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *ICRA*, 2020. 1

[12] Yana Hasson, Bugra Tekin, Federica Bogo, Ivan Laptev, Marc Pollefeys, and Cordelia Schmid. Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction. In *CVPR*, 2020. 1, 3

[13] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019. 1, 3

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 4

[15] Tomáš Hodaň, Frank Michel, Eric Brachmann, Wadim Kehl, Anders Glent Buch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, Caner Sahin, Fabian Manhardt, Federico Tombari, Tae-Kyun Kim, Jiří Matas, and Carsten Rother. BOP: Benchmark for 6D object pose estimation. In *ECCV*, 2018. 1, 3

[16] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. BOP challenge 2020 on 6D object localization. *ECCV Workshops*, 2020. 1, 3

[17] Umar Iqbal, Pavlo Molchanov, Thomas Breuel, Juergen Gall, and Jan Kautz. Hand pose estimation via latent 2.5D heatmap regression. In *ECCV*, 2018. 1

[18] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. CosyPose: Consistent multi-view multi-object 6D pose estimation. In *ECCV*, 2020. 1

[19] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. DeepIM: Deep iterative matching for 6D pose estimation. In *ECCV*, 2018. 1

[20] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. GANerated Hands for real-time 3D hand tracking from monocular RGB. In *CVPR*, 2018. 1

[21] Franziska Mueller, Micah Davis, Florian Bernard, Oleksandr Sotnychenko, Mickeal Verschoor, Miguel A. Otaduy, Dan Casas, and Christian Theobalt. Real-time pose and shape reconstruction of two interacting hands with a single depth camera. In *SIGGRAPH*, 2019. 3

[22] Kiru Park, Timothy Patten, and Markus Vincze. Pix2Pose: Pixel-wise coordinate regression of objects for 6D pose estimation. In *ICCV*, 2019. 1

[23] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. PVNet: Pixel-wise voting network for 6DoF pose estimation. In *CVPR*, 2019. 1

[24] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. In *SIGGRAPH Asia*, 2017. 3

[25] Adrian Spurr, Umar Iqbal, Pavlo Molchanov, Otmar Hilliges, and Jan Kautz. Weakly supervised 3D hand pose estimation via biomechanical constraints. In *ECCV*, 2020. 1

[26] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3D orientation learning for 6D object detection from RGB images. In *ECCV*, 2018. 1

[27] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *ECCV*, 2020. 1

[28] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+O: Unified egocentric recognition of 3D hand-object poses and interactions. In *CVPR*, 2019. 1

[29] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-time seamless single shot 6D object pose prediction. In *CVPR*, 2018. 1

[30] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. In *CoRL*, 2018. 1

[31] Carl Vondrick, Donald Patterson, and Deva Ramanan. Efficiently scaling up crowdsourced video annotation. *IJCV*, 101(1):184–204, Jan 2013. 2

[32] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martin-Martin, Cewu Lu, Li Fei-Fei, and Silvio Savarese. DenseFusion: 6D object pose estimation by iterative dense fusion. In *CVPR*, 2019. 1

[33] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 4

[34] Yu Xiang, Tanner Schmidt, Venkatraman Nafurayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In *RSS*, 2018. 2, 3

[35] Wei Yang, Chris Paxton, Maya Cakmak, and Dieter Fox. Human grasp classification for reactive human-to-robot handovers. In *IROS*, 2020. 1

[36] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhant Jain, and Tae-Kyun Kim. BigHand2.2M benchmark: Hand pose dataset and state of the art analysis. In *CVPR*, 2017. 1

[37] Christian Zimmermann and Thomas Brox. Learning to estimate 3D hand pose from single RGB images. In *ICCV*, 2017. 1

[38] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max J. Argus, and Thomas Brox. FreiHAND: A dataset for markerless capture of hand pose and shape from single RGB images. In *ICCV*, 2019. 3