Imitrob: Imitation Learning Dataset for Training and Evaluating 6D Object Pose Estimators [Supplementary material]

Jiri Sedlar[1,*], Karla Stepanova[1,*], Radoslav Skoviera[1], Jan K. Behrens[1], Matus Tuna[2], Gabriela Sejnova[1], Josef Sivic[1], and Robert Babuska[1,3]

This supplementary material provides additional information to paper *Imitrob: Imitation Learning Dataset for Training and Evaluating 6D Object Pose Estimators* (DOI 10.1109/LRA.2023.3259735). Sec. A covers the setup details and additional experimental results, Sec. B contains links to the dataset documentation and supplementary code and describes the intended uses of the dataset, Sec. C provides details on the dataset licensing and hosting, including the maintainance plan, and Sec. D contains a standardized datasheet for the *Imitrob* dataset.

A. Setup Details and Additional Experimental Results

This section provides details about the *Imitrob* dataset acqusition setup, evaluation metrics, segmentation methods, 6D object pose estimator setup, and all experiments, including a comparison with another object pose estimator and the impact of the tracker position.

The sensor setup calibration is described in Sec. A.1 and the method for calibration of the HTC Vive tracker to the tool is explained in Sec. A.2. The evaluation metrics are defined in detail in Sec. A.3 and the object segmentation methods used for background augmentation are described in Sec. A.4. The 6D object pose estimator DOPE [1] settings used for the experiments are given in Sec. A.5.

The remaining sections contain detailed experimental results and ablation studies. Sec. A.6 shows the impact of image resolution and batch size on the accuracy of the 6D object pose estimator, while Sec. A.7 compares the impact of different object segmentation methods on the benefit of the background augmentation. Secs. A.8-A.13 contain complete results of the experiments evaluating different data augmentation methods, generalization across camera viewpoints, left/right hand, demonstrators, robustness to clutter, and performance on different tools and tasks, respectively. In Sec. A.14 we compare the model-free estimator DOPE with a model-based 6D object pose estimator CosyPose [2] on the power drill tool and in Sec. A.15 we evaluate the performance of the 6D object pose estimator DOPE with respect to different HTC Vive tracker positions.

[1]Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, Czech Republic
[2]Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava, Slovakia
[3]Cognitive Robotics, Faculty of 3mE, Delft University of Technology, The Netherlands
*Both authors contributed equally. E-mail: jiri.sedlar@cvut.cz, karla.stepanova@cvut.cz

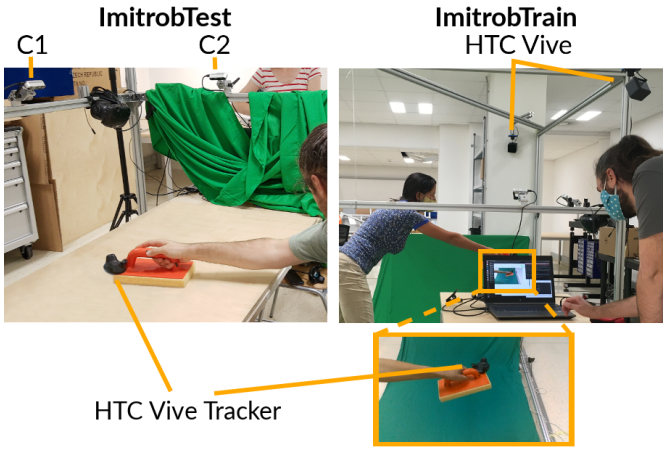


Fig. 1. The experimental setup for collection of *ImitrobTest* and *ImitrobTrain* datasets. The setup consists of two RGB-D cameras (front camera C1 and right-hand side camera C2), two HTC Vive lighthouses, and an HTC Vive tracker attached to the tool.

A.1 Sensor setup calibration

To calibrate the HTC Vive coordinate frame O_{htc} (in one of the lighthouses marked as HTC Vive in Fig. 1) to the chessboard coordinate frame O_{w}, spherical motion patterns centered at different chessboard corners p_{w} were recorded using an HTC Vive tracker mounted on a pointed metal rod (the rod is shown in Fig. 2a).

The sphere center points p_{htc} (relative to O_{htc}) were computed using orthogonal distance regression. The distances of all center points to the common plane found using RANSAC were smaller than 1 mm, i.e. all points lie on the flat plane of the chessboard pattern. The optimal Euclidean transformation H from p_{htc} to p_{w} (i.e. transformation between O_{htc} and O_{w}) was found using the SVD algorithm [3]. The average deviation (residual r_{avg}) of the acquired center points from the regular chessboard grid pattern (acquired from the cameras) was below 2 mm for all experiments. The deviation was calculated as

$$r_{\text{avg}} = \sum_{i=1}^{N} \frac{\left\| H p^i_{\text{htc}} - p^i_{\text{w}} \right\|_2 + \left\| H^{-1} p^i_{\text{w}} - p^i_{\text{htc}} \right\|_2}{2N} , \quad (1)$$

where N is the number of acquired center points and corresponding points in coordinate frames O_{htc} and O_{w} have the same index.

The final accuracy of the ground truth poses is also dependent on the ability of the HTC Vive to provide accurate and stable poses of the tracker attached to a tool with respect to O_{htc}. The accuracy of HTC Vive in dynamic situations is evaluated in detail in [4].

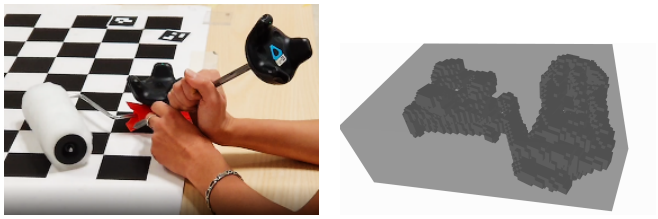
A.2 HTC Vive tracker to tool calibration

The method presented in this section allows finding a description of the object surface with respect to an attached motion tracker that provides reference 6D data. In this paper, it is used to find the bounding boxes of the manipulated objects relative to the tracker, which in turn are used to generate the reference bounding boxes for the *Imitrob* dataset.

Note that the computed bounding boxes do not affect the performance of 6D pose estimators because the training and testing are executed using the same bounding box calibration. The accuracy of the pose annotations is mainly determined by the HTC Vive dynamic accuracy, which was evaluated in [4]. Nonetheless, we look for bounding boxes that 1) contain the object, 2) align with the tracker axis, and 3) are minimal in size. In this way, the bounding boxes can be used to create the segmentation masks provided with the dataset and ensure consistent appearance in different experiments. The tracker attachment was chosen to allow unobstructed handling of the tool and good visibility of the tracker. If possible, the tracker was aligned with the main tool axis. To find the object dimensions relative to the tracker, we traced the tool and tracker surfaces with a pointing device (pointed rod with another HTC Vive tracker) while recording the positions of both trackers (see Fig. 2a). Contour tracing for surface reconstruction was described in [5]. We transform the N recorded pointing tip points into the frame of the tool tracker to obtain a set of measurements $P = \{p_i \subset \mathbb{R}^3\}$. The existence of a non-empty set $\hat{P} \subset P$ of outliers makes filtering of the measurements P necessary. Our filtering approach based on measurement density is explained next. For a regular grid V within the axis-aligned bounding box of the traced volume, we calculate a measurement density d_i as the number of measurements closer than a threshold δ at each grid vertex $v_i \in V$:

$$d_i = \sum_{p_i \in \Gamma} \begin{cases} 0 & \text{if } ||p_i - v_i||_2 > \delta \\ 1 & \text{if } ||p_i - v_i||_2 \leq \delta \end{cases} , \tag{2}$$

where Γ is a subset of the set of measured points P. We consider the voxel centered at grid vertex v_i to be part of the object surface if the density of measurements d_i at the given grid cell is larger than a threshold ζ, i.e. $d_i \geq \zeta$. To approximate Eq. (2) for the purpose of comparing it with the threshold ζ, we use a *k-d Tree* to efficiently organize the measurement points $p_i \in P$ and query it for the $\zeta + 1$ nearest neighbors P_i for each grid vertex v_i with a cut-off distance of $\delta + \epsilon$. We evaluate Eq. (2) for $\Gamma = P_i$ to decide if v_i is part of the object's surface. In short, we decide for each grid point whether it is part of the tool's surface by evaluating how many measurements are present in its vicinity. For efficiency, we check only just enough nearest neighbors to decide if the threshold was reached.



(a) Tracing of a tool (b) Tool bounding box

Fig. 2. Calibration of the tool with respect to the tracking device. a) The tool (roller) surface is traced with a pointing device. b) The collected data (here 6 364 surface trace points) is used to calculate a voxel grid for the tool (dark gray) and the final bounding box (light gray).

From this, we create a voxel grid with the dimensions of the object, on which we calculate a minimal bounding box using the trimesh library [6]. To find instead the smallest bounding box that is aligned with the tracker's z-axis (second bounding box property), we rotate the occupied voxels in $0.1°$ steps around the z-axis and record the volume of each axis-aligned bounding box. The rotation with the smallest volume is then used. In this work, we used $\delta = 0.01\,\text{m}$ and a resolution of 200 grid points per meter. Fig. 2b visualizes the resulting voxel grid (dark gray) and the bounding box (light gray) for the roller.

A.3 Evaluation metrics

The 6D object pose can be defined by 3D coordinates of the bounding box vertices $\mathbf{p}^1, \ldots, \mathbf{p}^8 \in \mathbb{R}^3$ or by a rigid transformation $[\mathbf{R}|\mathbf{t}] \in SE(3)$, consisting of a rotation matrix $\mathbf{R} \in SO(3)$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$. To evaluate the performance of a 6D object pose estimator on the *ImitrobTest* dataset, we use the following three metrics.

a) ADD pass rate: The ADD [1] is defined as the average Euclidean distance between the corresponding predicted ($\mathbf{p}^i_{\text{pre}}$) and reference ($\mathbf{p}^i_{\text{ref}}$) vertices and centroid ($\mathbf{p}^9 \in \mathbb{R}^3$) of the object 3D bounding box:

$$\text{ADD} = \frac{1}{9} \sum_{i=1}^{9} ||\mathbf{p}^i_{\text{pre}} - \mathbf{p}^i_{\text{ref}}||_2 \ . \tag{3}$$

The ADD pass rate (ADD_t) measures the percentage of frames where the ADD value of the prediction (P) is lower than a selected threshold ($t \in \mathbb{R}$):

$$\text{ADD}_t = \frac{|\{P|\,\text{ADD} \leq t\}|}{|\{P\}|} \cdot 100\% \ . \tag{4}$$

A higher ADD_t value for a given threshold t indicates a better prediction accuracy of the object 3D bounding box. In our experiments, we report ADD pass rate values for thresholds $t = 2\,\text{cm}$ (ADD_2), $5\,\text{cm}$ (ADD_5) and $10\,\text{cm}$ (ADD_{10}). By definition, $\text{ADD}_2 \leq \text{ADD}_5 \leq \text{ADD}_{10}$.

For comparison of models trained with ($\text{ADD}^{\text{aug}}_t$) and without ($\text{ADD}^{\text{noaug}}_t$) augmentation, we use the ratio of their respective ADD pass rates:

$$\text{ADD}^{\text{ratio}}_t = \frac{\text{ADD}^{\text{aug}}_t}{\text{ADD}^{\text{noaug}}_t} \ . \tag{5}$$

A higher $\mathrm{ADD}_t^{\mathrm{ratio}}$ value indicates a bigger benefit of the augmentation.

b) Rotation error: The rotation error measures the angle between the predicted ($\mathbf{R}_{\mathrm{pre}}$) and reference ($\mathbf{R}_{\mathrm{ref}}$) rotation matrices:

$$E_{\mathrm{rot}} = \arccos\left(\frac{\mathrm{trace}({\mathbf{R}_{\mathrm{pre}}}^{-1}\mathbf{R}_{\mathrm{ref}}) - 1}{2}\right) . \quad (6)$$

A lower E_{rot} value corresponds to a better estimate of the object orientation.

c) Translation error: The translation error measures the Euclidean distance between the predicted ($\mathbf{t}_{\mathrm{pre}}$) and reference ($\mathbf{t}_{\mathrm{ref}}$) translation vectors:

$$E_{\mathrm{tra}} = ||\mathbf{t}_{\mathrm{pre}} - \mathbf{t}_{\mathrm{ref}}||_2 . \quad (7)$$

A lower E_{tra} value indicates a better localization of the object in space.

A.4 Object segmentation methods for background augmentation

In order to augment the image background, we need to segment the shape of the object. Because we work with hand-held tools that are heavily occluded by the hand, we segment both the tool and the hand. We leverage the green background in the *ImitrobTrain* dataset to segment the image by thresholding. To remove the rest of the arm, we crop the segmentation mask by the convex hull of the tool 3D bounding box vertices projected into the 2D image. The result is a binary mask of the tool and hand (*MaskThresholding*, see Fig. 3b). We enhance the segmentation by F, B, Alpha Matting [7], which estimates also the foreground opacity and color along the boundaries. The output is an RGBA image with opaque foreground, transparent background, and smooth boundaries between them (*MaskFBA*, see Fig. 3c).

A.5 Object pose estimator DOPE settings

We estimate the 6D object pose by the DOPE method [1] in each frame and concatenate the frame predictions into a complete trajectory. Since we focus our evaluation on pose estimation in separate images, we do not post-process the individual frame predictions with any temporal or dynamic model. Our implementation of the DOPE method was based on the referenced PyTorch implementation by [1]. We trained our models on Nvidia 1080Ti and Nvidia V100 GPUs using the ADAM optimizer [8], learning rate 0.0001, and batch size 16. To be able to train with this batch size, we downsized the input image dimensions by half to 424×240 pixels, which decreased the training time without a negative impact on the accuracy (see Sec. A.6 for the corresponding ablation study). The ground truth belief maps we used for the DOPE training contain a 2D Gaussian with 2-pixel standard deviation and 2-pixel radius at the bounding box vertices and centroid. In all experiments, we train on subsets of the *ImitrobTrain* dataset and test on subsets of the *ImitrobTest* dataset.

Fig. 3. Segmentation of a frame from the *ImitrobTrain* dataset. a) Original image (*NoAug*) and segmentation of the tool and hand by b) *MaskThresholding* and c) *MaskFBA* (see Sec. A.4).

TABLE I
IMPACT OF INPUT IMAGE RESOLUTION AND BATCH SIZE ON DOPE 6D OBJECT POSE ACCURACY (SEE SEC. A.6). ADD PASS RATES ACHIEVED BY TRAINING WITH THE ORIGINAL RESOLUTION (848×480 PIXELS) AND BATCH SIZE 8 AND WITH THE IMAGES DOWNSIZED BY A FACTOR OF TWO (424×240 PIXELS) AND BATCH SIZE 16.

ADD_t	848×480 pixels			424×240 pixels		
threshold t	2 cm	5 cm	10 cm	2 cm	5 cm	10 cm
glue gun	7.4	49.8	75.6	**9.9**	**57.8**	**79.8**
grout float	**12.5**	**75.6**	93.0	9.7	73.9	**97.7**
roller	**4.5**	40.3	67.4	4.3	**48.6**	**84.9**
average	**8.1**	55.2	78.7	8.0	**60.1**	**87.5**

TABLE II
IMPACT OF OBJECT SEGMENTATION METHODS (SEE SEC. A.7). ADD PASS RATES ACHIEVED BY MODELS TRAINED USING DIFFERENT OBJECT SEGMENTATION METHODS (SEE SEC. A.4) FOR DATA AUGMENTATION.

ADD_t	*MaskThresholding*			*MaskFBA*		
threshold t	2 cm	5 cm	10 cm	2 cm	5 cm	10 cm
glue gun	9.4	57.4	78.4	**10.7**	**60.2**	**80.6**
grout float	**10.7**	**77.8**	97.1	10.5	73.0	**97.3**
roller	4.3	41.2	81.3	**4.3**	**50.5**	**86.3**
average	8.1	58.8	85.6	**8.5**	**61.2**	**88.1**

A.6 Impact of image resolution and batch size

We explore the impact of downsizing the input images and increasing the batch size on the quality of the 6D object pose estimation by DOPE. Table I presents a comparison of ADD pass rates for the original (848×480 pixels, batch size 8) and downsized (424×240 pixels, batch size 16) frames. While the similarity in performance for the 2 cm threshold could be attributed to a trade-off between the larger batch size and loss of detail, the increased batch size clearly improved the accuracy for the 5 cm and 10 cm thresholds. Therefore, we use the 424×240 pixel resolution and batch size 16 for all other experiments.

A.7 Comparison of object segmentation methods

Table II compares the ADD pass rates for the *MaskThresholding* and *MaskFBA* object segmentation methods (see Sec. A.4). Because *MaskFBA* outperforms *MaskThresholding* on average for all three thresholds, we use *MaskFBA* for object segmentation in all other experiments.

A.8 Benefits of data augmentation

Table III shows 2 cm, 5 cm, and 10 cm ADD pass rates for the DOPE estimator using different background augmentation methods from Sec. IV-B in the main paper. For all three thresholds, the best average ADD pass rates were achieved by

TABLE III

COMPARISON OF DATA AUGMENTATION METHODS (SEE SEC. A.8). ADD PASS RATES ACHIEVED BY MODELS TRAINED WITH DIFFERENT DATA AUGMENTATION METHODS (SEE SEC. V IN THE MAIN PAPER).

ADD_t	*NoAug*			*BgNoise*			*BgRandom*			*BgAlternate*			*BgBlend*		
threshold t	2 cm	5 cm	10 cm	2 cm	5 cm	10 cm	2 cm	5 cm	10 cm	2 cm	5 cm	10 cm	2 cm	5 cm	10 cm
glue gun	*1.1*	*17.0*	*36.3*	1.1	18.9	41.7	6.7	36.5	53.4	5.6	45.4	72.2	**9.9**	**57.8**	**79.8**
grout float	*5.3*	*45.4*	*80.3*	3.3	43.6	86.1	7.7	60.4	84.2	9.6	71.8	95.8	**9.7**	**73.9**	**97.7**
roller	*3.4*	*25.3*	*56.6*	3.8	26.3	52.9	2.9	39.8	85.5	**6.3**	**52.2**	**85.6**	4.3	48.6	84.9
average	*3.3*	*29.2*	*57.8*	2.7	29.6	60.2	5.8	45.6	74.4	7.2	56.5	84.5	**8.0**	**60.1**	**87.5**

TABLE IV

GENERALIZATION ACROSS CAMERA VIEWPOINTS (SEE SEC. A.9). COMPARISON OF ADD PASS RATES FOR COMBINATIONS OF CAMERA VIEWPOINTS (FRONT CAMERA C1 AND RIGHT-HAND SIDE CAMERA C2) BETWEEN TRAINING AND TESTING. "SAME" REFERS TO TRAINING AND TESTING ON THE SAME CAMERA, "OTHER" TO TRAINING ON ONE CAMERA AND TESTING ON THE OTHER, AND "BOTH" TO TRAINING ON BOTH CAMERAS. THE LAST ROW SHOWS THE AVERAGE VALUES FOR MODELS TRAINED WITHOUT DATA AUGMENTATION (*NoAug*).

ADD_t	Training	Test C1			Test C2		
threshold t	camera	2 cm	5 cm	10 cm	2 cm	5 cm	10 cm
glue gun	Same	8.7	61.6	**84.6**	5.8	**53.3**	**82.6**
	Other	0.0	0.2	1.8	0.1	2.2	12.0
	Both	**13.7**	**63.6**	83.4	**6.3**	50.6	75.0
grout float	Same	2.4	50.7	90.1	**6.7**	66.0	96.2
	Other	0.0	0.0	0.1	0.0	1.2	21.5
	Both	**13.2**	**78.1**	**97.8**	**6.7**	**77.3**	**97.3**
roller	Same	5.9	56.7	80.9	**0.4**	**37.2**	**89.0**
	Other	0.0	0.1	1.2	0.0	0.0	0.0
	Both	**8.2**	**66.3**	**85.8**	0.0	19.1	72.8
average	Same	5.7	56.3	85.2	4.3	**52.2**	**89.3**
	Other	0.0	0.1	1.0	0.0	1.1	11.2
	Both	**11.7**	**69.3**	**89.0**	**4.4**	49.0	81.7
NoAug	*Same*	*1.7*	*28.7*	*66.0*	*3.2*	*41.8*	*74.3*
	Other	*0.0*	*0.0*	*0.6*	*0.0*	*0.8*	*8.0*
	Both	*3.0*	*23.5*	*54.3*	*3.5*	*34.8*	*61.1*

TABLE V

GENERALIZATION ACROSS LEFT AND RIGHT HAND (SEE SEC. A.10). COMPARISON OF ADD PASS RATES FOR COMBINATIONS OF HOLDING THE TOOL IN THE LEFT (LH) OR RIGHT (RH) HAND BETWEEN TRAINING AND TESTING. "SAME" REFERS TO TRAINING AND TESTING ON THE SAME HAND, "OTHER" TO TRAINING ON ONE HAND AND TESTING ON THE OTHER, AND "BOTH" TO TRAINING ON BOTH LEFT AND RIGHT HAND. THE LAST ROW SHOWS THE AVERAGE VALUES FOR MODELS TRAINED WITHOUT DATA AUGMENTATION (*NoAug*).

ADD_t	Training	Test LH			Test RH		
threshold t	hand	2 cm	5 cm	10 cm	2 cm	5 cm	10 cm
glue gun	Same	5.6	**60.9**	77.6	1.8	25.6	56.9
	Other	0.7	8.6	25.0	1.1	22.8	54.8
	Both	**8.3**	60.8	**87.0**	**8.2**	**51.4**	**74.9**
grout float	Same	3.2	60.7	95.3	3.4	57.1	93.7
	Other	1.2	31.1	74.5	2.3	53.4	81.2
	Both	**7.9**	**70.8**	**97.8**	**11.0**	**83.8**	**98.3**
roller	Same	5.9	**49.9**	83.0	1.6	**37.9**	73.4
	Other	0.8	23.3	45.1	0.3	4.9	22.7
	Both	**9.4**	**49.9**	**89.3**	**4.9**	35.7	**75.4**
average	Same	4.9	57.2	85.3	2.3	40.2	74.6
	Other	0.9	21.0	48.2	1.2	27.0	52.9
	Both	**8.5**	**60.5**	**91.4**	**8.0**	**57.0**	**82.8**
NoAug	*Same*	*1.9*	*29.4*	*57.1*	*1.5*	*22.5*	*51.9*
	Other	*0.9*	*13.7*	*35.0*	*0.8*	*18.4*	*47.6*
	Both	*4.0*	*31.3*	*58.4*	*2.0*	*25.6*	*52.6*

TABLE VI

GENERALIZATION ACROSS DEMONSTRATORS (SEE SEC. A.11). COMPARISON OF ADD PASS RATES FOR VARIOUS COMBINATIONS OF THE FOUR DEMONSTRATORS (S1-S4) BETWEEN TRAINING AND TESTING. "ALLTOALL" REFERS TO TRAINING ONE MODEL FOR ALL SUBJECTS, "THREETODIFF" TO TRAINING ON THREE SUBJECTS AND TESTING ON THE REMAINING ONE, AND "ONETOSAME" TO TRAINING AND TESTING ON THE SAME SUBJECT. THE ADD PASS RATES ARE AVERAGED ACROSS ALL TEST SUBJECTS (I.E. S1-S4). THE LAST ROW SHOWS THE AVERAGE FOR MODELS TRAINED WITHOUT DATA AUGMENTATION (*NoAug*).

ADD_t	AllToAll			ThreeToDiff			OneToSame		
threshold t	2 cm	5 cm	10 cm	2 cm	5 cm	10 cm	2 cm	5 cm	10 cm
glue gun	**10.1**	**57.2**	**80.1**	6.2	52.6	78.2	3.4	32.8	69.4
grout float	**9.5**	**77.2**	**97.5**	6.0	64.1	95.3	1.8	38.8	81.6
roller	3.9	**41.8**	**78.1**	**4.5**	39.1	77.3	0.7	21.8	53.8
average	**7.9**	**58.8**	**85.2**	5.6	52.0	83.6	2.0	31.1	68.3
NoAug	*3.1*	*28.9*	*57.7*	*1.9*	*22.3*	*52.3*	*1.2*	*18.7*	*51.6*

the *BgBlend* augmentation. $\text{ADD}_2^{\text{ratio}} = 2.4$, $\text{ADD}_5^{\text{ratio}} = 2.1$, and $\text{ADD}_{10}^{\text{ratio}} = 1.5$ values for *BgBlend* indicate a big improvement in accuracy with respect to training without augmentation (*NoAug*).

A.9 Generalization across camera viewpoints

Table IV shows the 2 cm, 5 cm, and 10 cm ADD pass rates for various combinations of the camera viewpoints between training and testing (see Sec. V in the main paper for a detailed description of the scenarios).

A.10 Generalization across left/right hand

Table V shows the 2 cm, 5 cm, and 10 cm ADD pass rates for various combinations of the left/right hand between training and testing (see Sec. V in the main paper for a detailed description of the scenarios).

A.11 Generalization across demonstrators

Table VI shows the 2 cm, 5 cm, and 10 cm ADD pass rates for various combinations of the demonstrators between training and testing (see Sec. V in the main paper for a detailed description of the scenarios).

A.12 Robustness to clutter

Table VII shows the 2 cm, 5 cm, and 10 cm ADD pass rates for the 6D object pose estimator DOPE for the presence/absence of clutter in the test environment (see Sec. V in the main paper for a detailed description of the scenarios).

A.13 Performance on different tools and tasks

Table VIII shows the 2 cm, 5 cm, and 10 cm ADD pass rates as well as the rotation and translation errors (see Sec. A.3) for all tools and tasks in the *ImitrobTest* dataset.

A.14 Comparison of DOPE and CosyPose results

The power drill has a 3D model available in the YCB Object dataset [9], which enables comparison of model-free and model-based object pose estimation methods on this tool. Here we compare the performance of model-free estimator DOPE [1] and model-based estimator CosyPose [2]. While CosyPose had been trained extensively on rendered images of 3D models of the objects from the YCB Object dataset, DOPE was trained on short video sequences of the hand-held power drill from the *ImitrobTrain* dataset (see Fig. 4a). Below we compare the detection rates and the rotation and translation errors of DOPE and CosyPose on power drill in the *ImitrobTest* dataset.

Using 80% confidence threshold, CosyPose detected the power drill in 34% of the test frames. Additional wrong objects were detected in less than 1% of the frames, but the wrong detections had significantly lower confidence than the correct ones and there were no cases where only wrong objects were detected. No object was detected in the remaining 66% of the frames.

TABLE VII

ROBUSTNESS TO CLUTTER (SEE SEC. A.12). COMPARISON OF ADD PASS RATES FOR GLUE GUN TASK FRAME TESTED ON A TABLE WITH ONLY THE GLUING FRAME (*NoClutter*) AND WITH A CLUTTER OF OTHER OBJECTS AROUND THE FRAME (*Clutter*). THE BOTTOM ROW SHOWS RESULTS FOR A MODEL TRAINED WITHOUT DATA AUGMENTATION (*NoAug*).

ADD_t	*NoClutter*			*Clutter*		
threshold t	2 cm	5 cm	10 cm	2 cm	5 cm	10 cm
glue gun (frame)	**8.6**	**61.8**	**90.1**	**11.0**	**61.5**	**83.7**
NoAug	*1.7*	*22.8*	*47.7*	*0.3*	*4.9*	*19.8*

TABLE VIII

PERFORMANCE OF THE 6D OBJECT POSE ESTIMATOR DOPE ON DIFFERENT TOOLS AND MANIPULATION TASKS (SEE SEC. A.13). 2 CM, 5 CM, AND 10 CM ADD PASS RATE ACCURACY (ADD_t) AND AVERAGE ROTATION (E_{rot}) AND TRANSLATION (E_{tra}) ERRORS FOR DIFFERENT TOOLS AND TASKS. INVALID DETECTIONS WERE EXCLUDED FROM THE COMPUTATION OF AVERAGE E_{rot} AND E_{tra}.

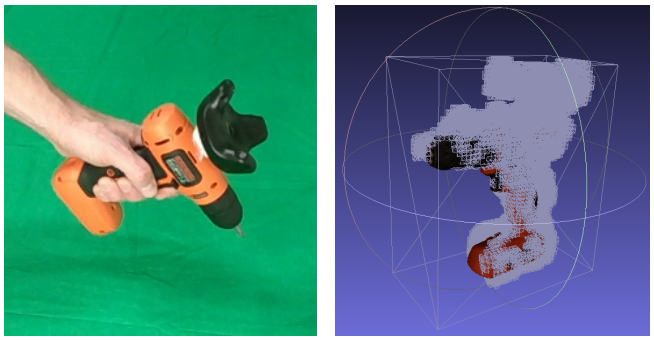
Tool	Task	ADD_t (%)			E_{rot}	E_{tra}
		2 cm	5 cm	10 cm	(deg)	(cm)
glue gun	frame	8.0	53.3	77.1	11.8	5.0
	densewave	**10.6**	61.9	88.6	5.0	3.6
	sparsewave	8.3	66.0	91.0	5.0	3.4
	average	9.0	60.4	85.6	7.3	4.0
grout float	round	9.2	74.4	**98.7**	**3.9**	2.7
	sweep	9.3	**82.7**	98.1	4.3	**2.2**
	average	9.3	78.6	98.4	4.1	2.5
roller	press	4.3	50.5	86.3	8.7	3.7
glue gun 2	lshape	0.0	9.0	41.9	38.5	9.9
glue gun 3	lshape	0.1	4.7	30.0	40.3	10.2
glue gun 4	lshape	1.2	23.4	52.6	20.9	8.4
heat gun	heating	0.0	13.2	56.3	14.3	7.0
power drill	down	5.6	59.8	87.0	8.0	3.8
soldering iron	soldering	0.5	12.8	41.4	35.6	9.0
average	-	3.3	34.7	64.4	19.8	6.5

To estimate the rotation and translation errors, we needed to convert our ground truth annotations to have the same reference coordinate system as the 3D model. Therefore, we aligned the 3D model mesh from the YCB Object dataset with our tracing mesh using ICP in MeshLab (see Fig. 4b). The average rotation and translation errors of CosyPose were $E_{\text{rot}} = 43.6°$ and $E_{\text{tra}} = 4.9$ cm, respectively.

However, the comparison with the DOPE results (see Table IX) is not straightforward. Most importantly, the performace of CosyPose may be affected by the presence of the HTC Vive tracker in the *ImitrobTest* dataset. In the case of DOPE, the tracker was present both in training and testing, whereas CosyPose was trained using a 3D model of the tool without the tracker. In addition, while DOPE was trained on a single object (power drill), CosyPose was trained on a set of multiple objects (YCB Object dataset), leading to possible false positives. Also, the DOPE results were filtered to exclude detections farther than one meter from the reference pose, but this affected less than one percent of frames.

A.15 Robustness to tracker position

To evaluate the impact of the HTC Vive tracker position on the tool on the accuracy of the 6D object pose estimator, we have recorded the same object with two different tracker positions: glue gun 3 has the tracker mounted on the top, while glue gun 4 has the tracker mounted on its left side (see Fig. 5). We have trained and evaluated the 6D object pose estimator



(a) power drill (b) 3D model + aligned meshes

Fig. 4. Alignment of the power drill tool meshes. a) The power drill in the *ImitrobTrain* dataset. b) The mesh of the 3D model from the YCB Object dataset [9] aligned to the tool tracing mesh using ICP in MeshLab. The geometric transformation describes the difference between the 3D model mesh origin and the HTC Vive tracker origin.

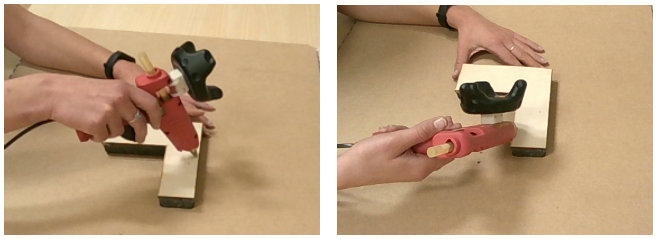
TABLE IX
COMPARISON OF MODEL-FREE ESTIMATOR DOPE [1] AND MODEL-BASED ESTIMATOR COSYPOSE [2] ON THE POWER DRILL TOOL. THE PERCENTAGE OF FRAMES WHERE THE TOOL WAS DETECTED (DETECTIONS) AND AVERAGE ROTATION (E_{rot}) AND TRANSLATION (E_{tra}) ERRORS.

6D object pose estimation method	Detections (% frames)	E_{rot} (deg)	E_{tra} (cm)
DOPE	**99.3%**	**8.0**	**3.8**
CosyPose	34.0%	43.6	4.9

DOPE using four different configurations: a) training and testing on glue gun 3, b) training and testing on glue gun 4 (these two configurations are reported also in the main paper), c) training on glue gun 3 and testing on glue gun 4, and d) training on glue gun 4 and testing on glue gun 3. Table X shows the resulting ADD_5 accuracy and average translation and rotation errors.

In the glue gun 3 to glue gun 3 and glue gun 4 to glue gun 4 configurations, the pose estimator performed better on glue gun 4 than on glue gun 3. This may be related to smaller occlusions of the tool when the tracker is mounted on its left side rather than on the top, considering that the side camera (C2) is on the right-hand side in out setup. However, in the glue gun 3 to glue gun 4 and glue gun 4 to glue gun 3 configurations, where the position of the tracker changed between training and testing, the estimator was not able to correctly predict the pose of the tool.

These experiments indicate that the selected tracker position can affect the 6D object pose estimator performance and that a transfer between different tracker positions is a challenging problem. For a real application, the HTC Vive tracker could be replaced by a smaller or concealed tracking device, e.g. based on an inertial measuring unit. However, this engineering task is beyond the scope of this paper, as the main goal of the *Imitrob* dataset is benchmarking 6D object pose estimation methods on hand-held tool manipulation tasks rather than deployment to the end-user.



(a) glue gun 3 (b) glue gun 4

Fig. 5. The same object with different HTC Vive tracker positions: a) glue gun 3 has the tracker mounted on the top; b) glue gun 4 has the tracker mounted on the left side.

TABLE X
ROBUSTNESS TO TRACKER POSITION (SEE SEC. A.15). 5 CM ADD PASS RATE ACCURACY (ADD_5) AND AVERAGE ROTATION (E_{rot}) AND TRANSLATION (E_{tra}) ERRORS FOR DIFFERENT COMBINATIONS OF TRAINING AND TESTING ON GLUE GUN 3 (TRACKER MOUNTED ON THE TOP) AND GLUE GUN 4 (TRACKER MOUNTED ON THE LEFT).

Training tool	Testing tool	ADD_5 (%)	E_{rot} (deg)	E_{tra} (cm)
glue gun 3	glue gun 3	4.7	40.3	10.2
glue gun 4	glue gun 4	**23.4**	**20.9**	**8.4**
glue gun 3	glue gun 4	0.0	148.6	18.3
glue gun 4	glue gun 3	0.0	148.1	19.8

B. DATASET DOCUMENTATION AND INTENDED USES

Dataset documentation: The *Imitrob* dataset documentation, metadata and download instructions are available at: http://imitrob.ciirc.cvut.cz/imitrobdataset.php

Supplementary code: The GitHub repository for the supplementary code (including example usage of the DOPE [1] method) is at: https://github.com/imitrob/imitrob_dataset_code

Intended uses: The dataset is primarily intended for benchmarking 6D pose estimation methods in manipulation tasks with hand-held objects and evaluating their ability to generalize with respect to various conditions. It can be also used to evaluate the effect of different data augmentation methods. Another usage is the methodology for data acquisition and 6D pose estimator training for new tools and tasks and a guideline for collecting more extensive datasets and benchmarking 6D object pose estimators on various tasks with hand-held tools, e.g. in imitation learning, grasping, virtual or augmented reality, etc. In general, we hope that the presented dataset will trigger further development of 6D object pose estimation methods and their usage in various industrial tasks based on the required accuracy.

Author statement: We bear all responsibility in case of violation of right in using our dataset or code. We confirm that we used all the existing assets in accordance to their license.

C. HOSTING, LICENSING, AND MAINTAINANCE PLAN

Hosting: The *Imitrob* dataset is hosted on our in-house servers, which are managed by our dedicated IT department. The dataset and source code are publicly available. The dataset website (http://imitrob.ciirc.cvut.cz/imitrobdataset.php) describes the dataset and provides download links. The source code is hosted on https://github.com/imitrob/imitrob_dataset_code.

Maintainance: The authors will provide important bug fixes to the community as commits to the GitHub repository. The dataset webpage will summarize changes to the code and the dataset. In the unlikely case that our in-house data center stops operating, we will migrate the dataset to another hosting and announce the new links in the GitHub repository.

Licensing: The provided dataset and supplementary code are copyrighted by us and published under the CC BY-NC-SA 4.0 license[1]. To use the code or the dataset, the original work has to be attributed, as specified by the authors on the dataset or code repository websites.

Contributions: Contributions to the dataset and supplementary code are welcome and contributors should contact the authors.

Contact: The contact e-mail address of the manager of the dataset: karla.stepanova@cvut.cz.

D. Datasheet for Dataset Imitrob

Questions from the Datasheets for Datasets (https://arxiv.org/abs/1803.09010) paper, v7.

D.1 Motivation

For what purpose was the dataset created?

The *Imitrob* dataset was created with the aim to enable imitation learning of manipulation tasks purely from visual observations. This includes the ability to recognize 6D pose of the hand-held objects. Current methods are typically trained and tested in different conditions than this kind of tasks, so it is very difficult to estimate how they will perform in manipulation tasks with hand-held tools. As expected, the tested methods showed quite low accuracy in the case of the mainipulation with hand-held tools, especially when generalization to new users, camera viewpoints, or tasks was needed. This motivated the creation of a new dataset, which would enable benchmarking of these methods.

Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?

The dataset was created by Jiri Sedlar, Karla Stepanova, Radoslav Skoviera, Gabriela Sejnova, Jan K. Behrens, and Josef Sivic within CIIRC CTU in Prague (Imitation learning centre http://imitrob.ciirc.cvut.cz) in collaboration with Matus Tuna from Comenius University in Bratislava and Robert Babuska from TU Delft.

Who funded the creation of the dataset?

Jiri Sedlar and Josef Sivic were supported by the European Regional Development Fund under the project IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000468) and the EU Horizon Europe Programme under the project AGIMUS (reg. no. 101070165). Matus Tuna was supported by project VEGA 1/0796/18. Karla Stepanova, Radoslav Skoviera, and Gabriela Sejnova were supported by the Technological Agency of CR under the grant Collaborative workspace of the future (reg. no. FV40319). Gabriela Sejnova was supported by CTU Student Grant Agency (reg. no. SGS21/184/OHK3/3T/37). Radoslav Skoviera, Jan Kristof Behrens, and Robert Babuska were supported by the European Regional Development Fund under the project Robotics for Industry 4.0 (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000470).

D.2 Composition

What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)?

The dataset consists of RGB-D images extracted from 352 video sequences, accompanied by 6D annotation. The videos capture simple manipulation tasks with 9 hand-held tools (glue gun, grout float, roller, glue gun 2, glue gun 3, glue gun 4, heat gun, power drill, and soldering iron) such as applying glue along a given trajectory, polishing a surface, or flattening a cloth.

How many instances are there in total (of each type, if appropriate)?

The *Imitrob* dataset contains images extracted from 352 video sequences (208 in the *ImitrobTest* dataset and 144 in the *ImitrobTrain* dataset) of hand-held tool manipulations. The *ImitrobTest* component of the dataset contains 100 332 images and the *ImitrobTrain* component contains 83 778 images.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?

The dataset contains all the possible instances.

What data does each instance consist of?

Each video frame contains the following data:

- 6D pose of the recorded tool (`6DOF/*.json`)
- 2D image coordinates of the tool 3D bounding box vertices and centroid (`BBox/*.json`)
- depth image (`Depth/*.png`)
- RGB image (`Image/*.png`)

In addition, each frame of the *ImitrobTrain* dataset also contains:

- binary mask of the segmented tool and hand (`Mask_thresholding/*.png`)
- RGB image with the segmented tool and hand opaque and the background transparent (`Mask/*.png`)

Each video sequence in the *Imitrob* dataset contains:

- 3D coordinates of the tool bounding box vertices and centroid with respect to the HTC Vive Tracker (`BB_in_tracker`) and intrinsic camera matrices for cameras C1 (`K_C1`) and C2 (`K_C2`) (`parameters.json`)

The training/test component, tool, task, subject, camera, left/right hand or presence/absence of clutter are identified in the name of the video sequence folder.

[1] https://creativecommons.org/licenses/by-nc-sa/4.0/

Is there a label or target associated with each instance?

Yes, each image is annotated with the 6D pose of the tool as well as the video sequence labels, including the identifier of the training/test component, tool, task, subject, camera viewpoint, left/right hand, or presence/absence of clutter.

Is any information missing from individual instances?

The 6D pose for individual data frames was interpolated. When the time difference between consecutive HTC Vive frames was longer than 100 ms, the corresponding camera images were discarded to ensure sufficient accuracy of the ground truth data. Otherwise no information is missing and the data is complete.

Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)?

Yes, the relationships are fully identified by the video sequence labels (see above) and the position of the frame in the sequence.

Are there recommended data splits (e.g., training, development/validation, testing)?

We explicitly state the data splits used for training and testing of the 6D pose estimator. The training set is an (augmented) subset of the *ImitrobTrain* dataset, and the test set is a subset of the *ImitrobTest* dataset.

Are there any errors, sources of noise, or redundancies in the dataset?

Sources of noise include the calibration of the cameras and the HTC Vive controllers and the synchronization between the HTC Vive and the cameras (the HTC Vive data were interpolated to the closest camera frame and if the distance between two consecutive frames was longer than 100 ms, the corresponding camera images were discarded to ensure sufficiently accurate ground truth data).

Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)?

Both the dataset and the supplementary code are self-contained.

Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals' non-public communications)?

N/A.

Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?

N/A.

Does the dataset relate to people?

N/A.

Does the dataset identify any subpopulations (e.g., by age, gender)?

N/A.

Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset?

N/A.

Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)?

N/A.

Any other comments?

N/A.

D.3 Collection process

How was the data associated with each instance acquired?

The directly observable data (RGB-D images) were synchronized with observable HTC Vive data. The parameters of each video sequence setup (such as the tool, task, subject, camera viewpoint, left/right hand, or presence/absence of clutter) were manually annotated and associated with the corresponding data.

What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)?

The visual part of the dataset was collected by two RGB-D cameras, specifically Intel RealSense D-435. The resolution of both RGB and depth images was set to 848x480 and they were recorded at 60 FPS. The 6D pose information was recorded using HTC Vive VR system in standard configuration. An HTC Vive tracker was mounted to the tools to acquire their pose. The cameras and HTC Vive system were calibrated towards a common coordinate system. The calibration of the camera and HTC Vive was validated by the average distances of associated points using a checkerboard pattern. The whole acquisition system was implemented via the Robot Operating System, using Python as the main programming language.

If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?

N/A.

Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?

Only the authors were involved in the collection process.

Over what timeframe was the data collected?

The data was collected in January and February 2021 (glue gun, grout float, and roller) and in March 2023 (glue gun 2, glue gun 3, glue gun 4, heat gun, power drill, soldering iron).

Were any ethical review processes conducted (e.g., by an institutional review board)?

N/A.

Does the dataset relate to people?

N/A.

D.4 Preprocessing/cleaning/labeling

Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)?

The data were originally recorded as ROS bag files, from which the individual data instances were extracted, synchronized, interpolated, and saved to separate folders. For the *ImitrobTrain* dataset, the masks were created by automatic segmentation of the RGB images.

Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)?

The original bag files are saved on our internal data storage, but are too big to be easily shareable.

Is the software used to preprocess/clean/label the instances available?

No, we don't provide the raw data and thus neither the code to process it. Dataset manipulation tools (for the already preprocessed and labeled data) are available on the supplementary code GitHub page: https://github.com/imitrob/imitrob_dataset_code.

D.5 Uses

Has the dataset been used for any tasks already?

This is the first use of the dataset.

Is there a repository that links to any or all papers or systems that use the dataset?

There are no papers that use our dataset, yet. Future uses will be added to the dataset/code website.

What (other) tasks could the dataset be used for?

The dataset is primarily intended for benchmarking 6D pose estimation methods in manipulation tasks with hand-held objects and evaluating their ability to generalize with respect to various conditions. It can be also used to evaluate the effect of different data augmentation methods. Another usage is the methodology for data acquisition and 6D pose estimator training for new tools and tasks and a guideline for collecting more extensive datasets and benchmarking 6D object pose estimators on various tasks with hand-held tools, e.g. in imitation learning, grasping, virtual or augmented reality, etc. In general, we hope that the presented dataset will trigger further development of 6D object pose estimation methods and their usage in various industrial tasks based on the required accuracy.

Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?

N/A.

Are there tasks for which the dataset should not be used?

N/A.

D.6 Distribution

Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?

N/A.

How will the dataset will be distributed (e.g., tarball on website, API, GitHub)?

The dataset is available on the dataset website: http://imitrob.ciirc.cvut.cz/imitrobdataset.php

When will the dataset be distributed?

In 2023.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?

The newly provided datasets and benchmarks are copyrighted by us and published under the CC BY-NC-SA 4.0 license[2].

Have any third parties imposed IP-based or other restrictions on the data associated with the instances?

N/A.

Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?

N/A.

D.7 Maintenance

Who is supporting/hosting/maintaining the dataset?

Karla Stepanova at CIIRC CTU in Prague.

How can the owner/curator/manager of the dataset be contacted (e.g., email address)?

Contact e-mail address: karla.stepanova@cvut.cz

[2] https://creativecommons.org/licenses/by-nc-sa/4.0/

Is there an erratum?

Any updates to the code will be visible as commits in the GitHub repository. The dataset website will summarize all changes to the code and the dataset.

Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)?

Any updates to the code will be visible as commits in the GitHub repository. The dataset website will summarize all changes to the code and the dataset.

If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)?

N/A

Will older dataset versions continue to be supported/hosted/maintained?

N/A.

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?

Yes, contributions to the dataset are welcome. Please get in touch with the maintainer of the dataset via e-mail (see above).

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 *CoRL*, 2018.

[2] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "CosyPose: Consistent multi-view multi-object 6D pose estimation," in *ECCV*, 2020.

[3] O. Sorkine-Hornung and M. Rabinovich, "Least-squares rigid motion using SVD," *Computing*, 2017.

[4] M. Borges, A. Symington, B. Coltin, T. Smith, and R. Ventura, "HTC Vive: Analysis and accuracy improvement," in *IROS*, 2018.

[5] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *SIGGRAPH*, 1992.

[6] M. Dawson-Haggerty *et al.*, "Trimesh [computer software]," https://trimsh.org/, 2019.

[7] M. Forte and F. Pitié, "F, B, Alpha Matting," *arXiv:2003.07711*, 2020.

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[9] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The YCB object and model set: Towards common benchmarks for manipulation research," in *ICAR*, 2015.