ESKO6d - A Binocular and RGB-D Dataset of Stored Kitchen Objects with 6d Poses*

Jesse Richter-Klug[1] and Constantin Wellhausen[1] and Udo Frese[1]

***Abstract*— We present a new dataset with the goal of advancing the state-of-the-art in object pose estimation especially for stored porcelain and glass crockery in kitchen scenes. Specifically the ESKO6d (EASE Stored Kitchen Objects with 6d poses) dataset features texture-less, glossy or glassy ordinary used objects which were naturally stored in a cupboard, drawer or dishwasher. There is a large degree of occlusion being the specific challenge in these scenes. Each scene was recorded in video sequences by two cameras (RGB-D (Kinect) and binocular) within multiple setup stages. The dataset contains an RGB-D image or binocular RGB image plus stereo-matched depth image as well as 6d pose ground truth and instance segmentation. Our dataset contains twelve stored object scenes with a combined amount of 47 video sequences captured by each camera, resulting in over 17k annotated Kinect images and more than 42k annotated stereo images showing around 50 different objects. The ground truth annotation is precise to $3.5\,mm$ ADD (details see paper). The dataset can be accessed under `http://www.informatik.uni-bremen.de/esko6d-dataset`.**

Besides the concrete dataset we propose a method of ground truth pose measurement based on an external 3d tracking system that allows on the one hand to precisely measure the object's pose inside a tight packed storage and on the other hand to obtain the object pose in several images with just one manual measurement.

I. Introduction

Datasets have been the driving force in computer vision research for over two decades now. They make scientific results comparable and relieve the individual researcher from having to record and label own data for test and evaluation. If a method involves machine learning also training data is needed and indeed, the availability of large labeled datasets, such as Imagenet [1], is commonly quoted as one of the reasons causing the recent deep learning revolution [2].

General computer vision has the goal to understand images in their full natural diversity and hence datasets in that area often feature photographs which are annotated e.g. with an object class, 2D bounding boxes or per pixel classes. Vision for robotics in contrast provides sensor information a robot can act on, in particular 6d object poses (position + orientation).

In this paper we present such a dataset (Fig. 1) that is motivated by the long term vision of a general household robot [3], [4]. Such a robot must be able to fetch household objects from cupboards, drawers and out of dishwashers. These scenes are special for object detection and pose estimation, because objects are

- tightly packed, even stacked and thus highly occluded,
- texture-less (crockery), transparent (glasses) and/or reflective (metal bowls),
- placed following a grouping scheme that, although not strictly defined, offers context for the recognition,
- sequentially placed, i.e. there is information about a previous situation relative to which something has changed.

Fig. 1. The proposed ESKO6d-dataset features kitchen objects in cupboards and dishwashers with 6d ground-truth pose. This setup is motivated by the vision of a household robot and involves extreme occlusion in particular for stacked objects. It also involves the grouping of objects typical for cupboards that may be exploited as semantic context information.

In particular we conjecture that often occlusion is so high that without the context of neighbouring objects and/or a previous situation there is probably not enough information to recognize the object from its visible part alone. The ESKO6d-dataset wants to foster addressing these challenges.

If ground truth is used for benchmarking an algorithm, the ground truth error needs to be much smaller than the algorithm's error, otherwise the benchmark is inadequate. We therefore quantify the ground-truth accuracy.

6d poses are usually annotated by manually aligning a wireframe model of the object with the image [5]. This is very laborious and difficult for large occlusion. We instead propose here a different procedure where both the camera(s) and the objects are localized with a 3d tracking system, the objects indirectly by localizing tiny markings. This way, a new object is localized once and after that the ground-truth

*The research reported in this paper has been (partially) supported by the German Research Foundation DFG, as part of Collaborative Research Center (Sonderforschungsbereich) 1320 EASE - Everyday Activity Science and Engineering, University of Bremen (http://www.ease-crc.org/). The research was conducted in subproject R02 'Multi-cue perception supported by background knowledge'.

[1]All authors are with Faculty of Mathematics and Computer Science, University of Bremen, 28359 Bremen, Germany `jesse@uni-bremen.de`

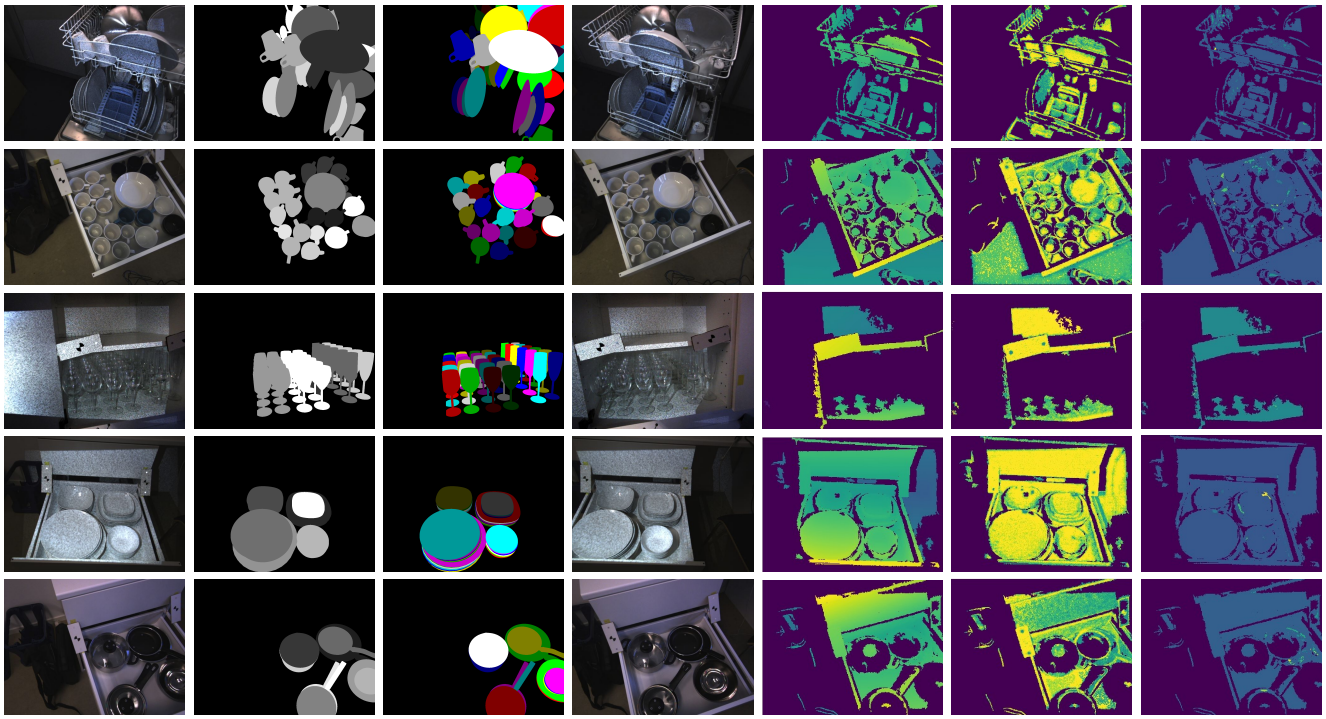


Fig. 2. Sample binocular (rc_visard) data included in the ESKO6d dataset. From left to right we show left-eye RGB image, class segmentation, instance segmentation, right-eye RGB image, disparity image as well as the disparity confidence and error image. Note: The shown segmentation images correspond to the left-eye image, the segmentation images corresponding to the right-eye image are omitted for clarity.

pose relative to the camera is available without further work, even if the camera is moved or other objects are added or removed as long as the object itself is not touched.

This paper extends the master's thesis [6] contributing:

- A dataset with tightly packed kitchen objects in cupboards, drawers and dishwashers, taken with both a binocular and two RGB-D-cameras with ground truth pose relative to the camera and instance segmentation.
- An investigation on the error of the ground truth.
- A procedure to efficiently annotate such a dataset.

The paper is organized as follows: After related work we describe details of the dataset and the annotation procedure, followed by an analysis of the ground truth accuracy.

II. Related Work

Among the datasets which aim at understanding the content of photographs there has been a transition from Pascal VOC [7] over ImageNet [1] to MS COCO [8]. Pascal VOC has 20 classes, 12k images, labels, bounding boxes and pixelwise segmentation and has been accompanied by a challenge from 2005-12. ImageNet has 1k classes (reduced from the originally annotated 20k), 14M images and labels and was accompanied by a challenge from 2012-17. MS COCO has 171 classes, only 330k images but pixelwise instance segmentation and human bodypart keypoints.

In the kitchen domain the EPIC-KITCHENS dataset [9] covers 55 h of human activity in 32 kitchens around the world with first-person video, activity labels and object 2d bounding boxes. This dataset aims at kitchen activities on a general human-level scale not at metrical object poses. [10] presents another human activity dataset about table setting.

For 6d object pose estimation commonly used datasets are LINEMOD [11], Occluded-LINEMOD [12], YCB-Video [13], T-LESS [14] and Rutgers [5], all RGB-D. Occluded-LINEMOD has 10k images, 20 objects and 3 lighting conditions and features office and toy objects (e.g. camera, duck, punch) on a table with moderate to large occlusion. Ground truth has been obtained by placing the test object in the middle of markers on the table. YCB [15] is originally a set of physical objects intended for reproducible grasping research. The YCB-Video dataset shows RGB-D videos of 21 of these objects on a table with 134k images. Ground truth has been obtained by manually aligning a model in the first depth image and then tracking the camera by aligning all objects in the point-cloud simultaneously. While this is efficient, it carries the danger of depth data or aligning problems affecting both the ground truth and an algorithm under evaluation in a correlated way hiding the problem. The Rutgers dataset targets the Amazon Picking Challenge [16] where objects are picked from warehouse shelves (24 objects, 10k images), with manually annotated poses. The scenes are related to our cupboard scenes, but the objects differ (boxes and bags vs. crockery and glasses) and there is no stacking or grouping. [17] evaluates on an unpublished dataset with YCB objects in extreme lighting conditions (e.g. sunny window). The T-LESS dataset features texture-less industry-relevant objects, which are partly composed of the same parts. This dataset's test images also

Fig. 3. Overview of all ESKO6d objects where often several instances of one type (mostly stacked) are in the scene. The numbers inside the images indicate the maximum possible instance occurrence of the shown object per scene. Objects marked with a 'x' occure inside a dishwasher scene.

include clutter and occlusion. Ground truth poses have been obtained by manually aligning the CAD object models with the scene models as well as using a turntable with markers to track the camera pose.

To our knowledge T-LESS is the only dataset, that makes a statement about its ground-truth accuracy. T-LESS measures the distance between the captured depth and the (according to the ground truth poses) rendered depth which results in 5 mm and 9 mm (depending on the sensor) average absolute difference. The authors state this is near the accuracy of the sensors.

As 6d annotation is laborious, researchers have tried learning from rendered images, where exact ground truth is readily available. [18] uses a mixture of few real and many rendered images while [19] and [17] use purely rendered images. An example for photorealistic rendering is the Falling Things dataset [20] which has been generated by the NVIDIA deep learning data synthesizer (NDDS) [21]. Despite this success all papers use real datasets for testing.

III. The ESKO6d Dataset

The dataset divides into multiple subsets, where each is available as binocular and RGB-D variant (for sample data see Fig. 2):

Firstly, we provide the recorded video sequences with the purpose to use for evaluation showing different stored object scenes, where the objects are incrementally placed. The placement was performed by students not involved in the project to obtain a reasonable degree of realism. The incremental placement is an intended feature: The content of e.g. a cupboard does not change rapidly and remembering a previous state could help for recognition. Such an approach seems promising and can be evaluated with our dataset. For each stage of placement a sequence with camera motion is provided. Objects occlude each other to a large degree and are stacked. This includes:

1) The **main dataset**, that is showing all **objects stored** in **cupboards and drawers**.
2) A **scene** with a subgroup of **objects** (3(a), 3(b)-2, 3(d)-1, 3(r)-2) **stored** in a **dishwasher**.

Secondly, we also provide auxiliary image sets, which may be used for any purpose:

3) One distraction free video sequence per object observing the object alone from many perspectives. Since objects in a dishwasher are often tilted, we added a second upside down sequence for these.
4) A domain randomization dataset, that consists of 100000 images created with our self-made 3d models (see III-A) by following the approach of [17].

A. Object description

For this dataset objects were selected, which one would naturally find in a common kitchen. All used items can be seen in Fig. 3. The dataset is built on three kinds of items: At first we selected porcelain items like different plates (e.g. 3(a), 3(f)) and cups (e.g. 3(d)). All porcelain items are unicolored, mostly white. Some are black which is more challenging because of low in-object contrast (e.g. 3(j), 3(o)). The second big item pool is built upon glassy items. Most of them are made of glass, e.g. drinking (3(l)), wine (3(b)) and shot glasses (3(q)-1) as well as glass bowls (3(m)-2). Additionally, we included some transparent plastic containers (3(s)). Lastly, the dataset contains metallic objects including pots (e.g. 3(h)), pans (3(c)) and metal bowls (3(r)).

An additional selection criterion was to get multiple pairs of items that mainly differ in size (e.g. 3(i), 3(j) or 3(c)).

For each object there is a handcrafted 3d-model, which includes an approximated material and is also available inside the ESKO6d dataset (`http://www.informatik.uni-bremen.de/esko6d-dataset/models`).

B. Sensor Setup

There are two independent sensor setups: A Kinect that provides RGB-D data and a roboception rc_visard, that is used to provide binocular RGB images with a random-dot projector active in every second image and approximately for every tenth RGB image a depth image based on stereo matching. All stages of all scenes are available in both sensors, of course with different sensor trajectories. The sensors specifications are:

- RGB-D Camera: Kinect Sensor for XBOX ONE (Kinect 2), Color 1920×1080 at 30 Hz with 16 bpp, Depth 512×424 at 30 Hz with 13 bpp; operation range from 0.5 m to 4.5 m (ToF), FOV 70°h./60°v.
- Stereo Camera: roboception rc_visard 160 color
  - Binocular RGB color image: resolution 1280×960, FOV 61°h./48°v., global shutter
  - Stereo-matched depth with RandomDot projector: roboception StereoPlus Module, range from 0.5 m to 3 m, depth resolution 3.8 mm at 0.8 m distance.

C. Data description

The data consists of the following parts.

1) Kinect Data: Each Kinect datum includes an RGB image, a depth image as well as the pose annotation (III-C.3) and all postcalculated images from III-C.4. Kinect data is generated in 30 Hz. The depth images have their own pose since RGB and Depth camera are not synchronized. All images are rectified (ignoring rolling shutter effects).

2) rc_visard binocular data: Each rc_visard datum includes two synchronized and rectified RGB images ('left eye', 'right eye'). The RGB images are sampled in 30 Hz. Every second binocular RGB image includes the projection from the RandomDot projector (which is only active during the exposure of that frame). At a speed of 3 Hz the images during frames with pattern projection are stereo-matched by the camera itself. The result is a depth image as well as a confidence and precision image (all with a resolution of 640×480 pixels). In addition a rc_visard datum includes the pose annotations (III-C.3) and the postcalculated images from III-C.4 for both eyes, too.

3) Pose Annotations: Since the main purpose of the proposed dataset is to aid development of 6d pose object estimation, each image has labels of all classes and 6d poses of every object of the current observed scene (in camera coordinates). In addition we provide per image the camera pose inside the scene.

To reduce the overhead of handling data from different sources inside a detection algorithm we aimed for consistency in the annotation declaration with an existing dataset. Therefore we write all our annotation and data files as similar as possible to the format defined by NVIDIA Deep learning Dataset Synthesizer (NDDS) [21], since to our knowledge this is currently the best tool to generate (additional) synthetic training data.

4) Calculated Data: For easy use we add accessory calculated data by rendering the 3d object models (cf. III-A) given their annotated pose in every image. In that way we receive semantic segmentation and instance segmentation images. Note that all of these additional images only comprise of the annotated poses (and the 3d models). Every pixel which is not pointing onto any object is segmented as background. While the drawer is an object and correctly handled, the dishwasher rack is not modeled and ignored in the segmentation image.

IV. Annotation System

The basic idea for obtaining an object pose relative to the camera is to locate both the object and the camera in the world frame with a 3d tracking system (ART TrackPack with DTrack2, 2 cameras, 2011).

This makes the ground truth independent from the images which are input to an algorithm under evaluation and avoids errors that affect both in a correlated way and are therefore hidden in the evaluation.

If the object is located once, the pose in the world frame stays valid, as long as the object is not moved. This works for camera movement yielding different viewpoints of the same scene without additional annotation effort. Moreover, it even works when objects are added to and removed from the scene as long as the remaining objects are not touched. Hence we build up scenes step by step.

A tracking target is attached to the cameras. Doing the same for every object in the scene is impossible, since it would spoil the images and hinder object placement. We have used two alternatives in the dataset because the more exact of them does not always work (Fig. 4):

- For round objects, e.g. cups and plates, we have produced cardboard lids with retroreflective stickers that are directly tracked (requiring orientation towards the tracking system) and removed after recording the pose.
- Where this is impossible, we locate human recognizable markings on the object by touching with a probe with

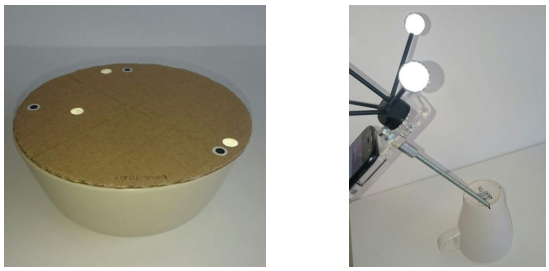


Fig. 4. Alternatives of measuring an object pose: cardboard lid with retroreflective stickers (left) and touching human recognizable markings on the object with a retroreflective marked probe

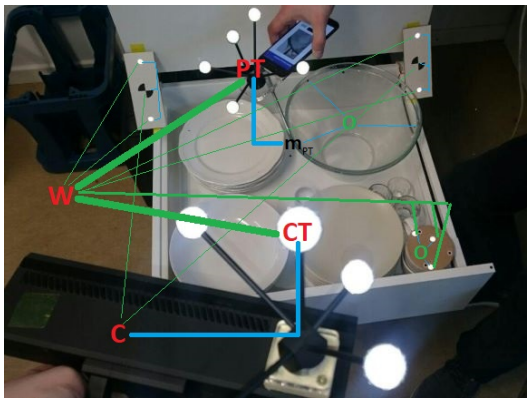


Fig. 5. The **Labeling process**: Camera and objects are located by a 3d tracking system, the objects indirectly with a probe that is in turn tracked. The crashtest markers refine the camera orientation measurement. The **coordinate systems** needed for the ground truth pose of an object relative to the camera: Markers have been located in the object frame (O). They are touched by a probe (m_P) that has a target (PT) that's tracked in the world frame (W). The camera target (CT) is also tracked leading to the camera frame (C). The crashtest markers and respective reflectors are used to refine the camera orientation. Colors indicate if a connection is measured (green) or precalibrated (blue).

a tracking target. The whole setup is attached to a smartphone providing a user interface for the annotator.

A. Coordinate systems and transformations

Figure 5 shows the coordinate systems involved in the process. In the transformation chain

$$T_{C\leftarrow O} = T^{-1}_{CT\leftarrow C} T^{-1}_{W\leftarrow CT} T_{W\leftarrow O}, \tag{1}$$

the transform $T_{CT\leftarrow C}$ is fixed and calibrated (blue) while $T_{W\leftarrow CT}$ is measured (green) by the 3d tracking system. In the reflector-lid mode, $T_{W\leftarrow O}$ is directly available. In the probe mode, it is measured indirectly. Therefore, we probe several markings sequentially and solve for $T_{W\leftarrow O}$ in

$$T_{W\leftarrow O} m^i_O = T^i_{W\leftarrow PT} m_{PT} \quad \forall i \tag{2}$$

where m^i_O is the known position of the i-th marking on the object, m_{PT} the fixed and calibrated probe position and $T^i_{W\leftarrow PT}$ is measured by the tracking system.

B. Calibration

The internal parameters of the rc_visard camera are calibrated by OpenCV's calibration functions [22] and the Kinect2 using the IAI Kinect2 tool [23] which also calibrates the relative pose between Kinect's depth and RGB cameras.

The tracking target poses $T_{CT\leftarrow C}$ of both cameras are calibrated by moving the camera around a checkerboard that is located by additional retroreflective markers and minizing the reprojection error of the checkerboard corners. Similarly m_{PT} is calibrated by moving the probe around a fixed point.

All parameter fitting was performed by Ceres [24].

C. Camera orientation refinement

The tracking system obtains the target's pose from its five retroreflective balls. If we assume a ball-position error of e_B, a target diameter of d and a distance to the object of l, roughly an object position error of $e_B + \frac{l}{d} e_B$ results, where e_B comes from translation and $\frac{l}{d} e_B \approx \frac{64cm}{9cm} e_B \approx 7 e_B$ from rotation.

Thus we have implemented a step to refine the rotation. In the scene two crashtest markers are placed, surrounded by two retroreflective markers each, to locate them by the tracking system (Fig. 5). These marker positions are projected into the image and refined using OpenCV's [22] subpixel corner refinement. Then the camera orientation is obtained from the image positions of these markers while keeping the translation fixed.

D. Time synchronization

There is no hardware synchronization between cameras and tracking system. As moving the camera (slowly) around the scene is much faster than taking still images with a tripod, we have developed a data-driven synchronization.

Therefore we track FAST keypoints using OpenCV's Lukas-Kanade optical flow implementation. These keypoints are converted to world rays using a hypothetical time offset to the tracking data. If the time offset is right, all rays of the same keypoint should intersect. Hence, we define the sum of distances between rays $0.4s$ apart as an error measure. This error is minimized with respect to the time offset.

V. Pose-Annotation Accuracy

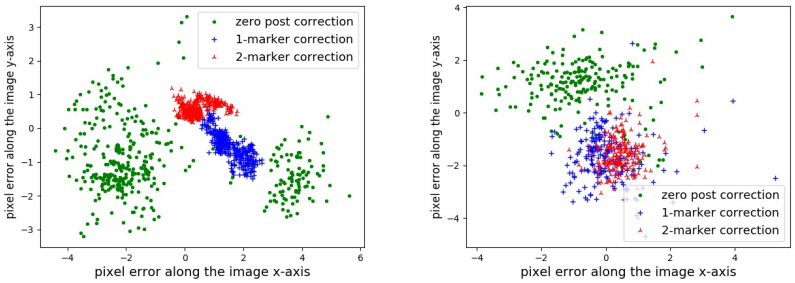
Ground-truth data can only be used for benchmarking if it is significantly preciser than the subject of the benchmark. Thus the precision of the ground-truth must be known. We provide here an indication of the ground-truth precision, by examining the world pose error and the image reprojection error. The first evaluates the errors from object to world in 6d, the second from world to camera but only in 2d.

A. Error Metric

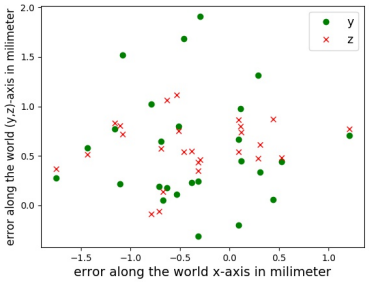
Regarding metric the recent state-of-the-art object pose estimators are evaluated with the ADD error ([11], [13], [17], [25]) or with its variants for symmetrical objects (eg. ADD-S in [11], [13], [25]). Note that despite more than half of the ESKO6d objects being symmetric the pose annotations are not (due to the non-ambiguous markings). Therefore we are using the ADD metric to evaluate our pose-annotation accuracy.

$$ADD(T_1,T_2) = \frac{1}{n}\sum_{i=1}^{n} ||T_1 p_i - T_2 p_i|| = \frac{1}{n}\sum_{i=1}^{n} ||p_i - T_1^{-1} T_2 p_i|| \tag{3}$$

The ADD metric [11] measures the average distance between corresponding object points $p_{1\ldots n}$ in ground truth and estimated pose. It implicitly weights rotation error with the object radius resulting in a single number. If the translation



(a) Sample object point reprojection errors with rc_visard (left) and with Kinect (right).



(b) Sample object translation errors along their x-, y- and z-errors labeled with the probe.

TABLE I

RMS ERROR OF THE MEASURED OBJECT POSE.

measuring mode	transl. [mm]	rotation [°]	ADD [mm]
lid with reflectors (*repeatability*)	0.15	0.10	0.41
markings probed (*error*)	1.27	0.32	2.10

error is $\leq e_t$ and the rotation error $\leq e_r$ the ADD is bounded by

$$\leq \frac{1}{n}\sum_{i=1}^{n}(e_t + e_r||p_i||) \leq e_t + re_r, \quad \text{with } r \geq \max_i ||p_i||. \quad (4)$$

Note that this also bounds all ADD variants for symmetrical objects since they may only loose the corresponding object points constraint to allow closer point distances and therefore lower errors.

B. Object pose error

We use the reflector lid method as reference here, because it directly measures $T_{W\leftarrow O}$ with the tracking system, while the other works indirectly. This view is supported by the very good repeatability of that method ($0.15\,mm/0.1°$) and the idea that validating the tracking system itself is beyond the scope of this paper. Also evaluating the precision of the 3d models and how well their reference system matches the one used in annotation is beyond the scope of this paper.

For the probe we measured multiple times the same object from different positions while comparing the outcomes. Figure 6(b) shows translation error distribution and Table I shows the rms result in rotation, translation and converted to ADD by (4) with $r = 150\,mm$ maximal radius.

C. Reprojection / Camera pose error

We evaluate the error in the measured (and refined) camera pose by reprojecting a crashtest marker tracked with the 3d tracking system into the image and taking the residual to the image position obtained from corner detection. This is similar to the refinement procedure (Section IV-C) but of course uses a different marker in the middle of the drawer.

Table II and Figure 6(a) show the result depending on the number of refinement-crashtest-markers in view. Residuals are in the range of a few pixels. The refinement procedure helps a lot for rc_visard but not significantly for Kinect.

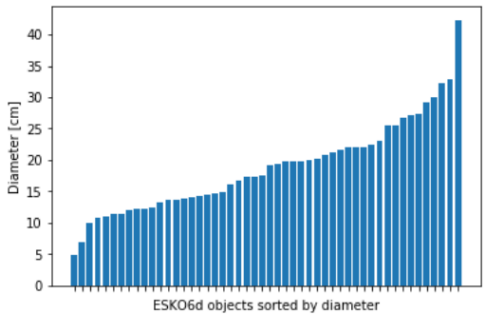


Fig. 6. Object's diameter distribution for ESKO6d

D. Analysis

Finally, we want to assess how much this increases the ADD of $T_{C\leftarrow O}$. We have therefore conducted a simulation study where we added varying amounts of noise to the position of the five balls of the tracking targets and observed to how much reprojection error and ADD that lead. Here the ADD refers to a $r = 150\,mm$ sphere at the average distance we experienced and the reprojection error refers to the sphere center. We took the ADDs corresponding to the reprojection errors from Table II and added the ADD of $T_{W\leftarrow O}$ from Table I. The result is shown as "estimated overall ADD" in Tab. II.

If algorithms target the 0.1 factor of the object's diameters as ADD(-S) accuracy threshold (like eg. [11]), than we can observe a factor > 3 from those to our ground truth for all but the two smallest objects (cf. Fig. 6), in case the two crashtest markers are in view. On account of this we say, with the exception of the two smallest objects (namely 3(q)), our ground truth values are clearly precise enough for evaluation.

TABLE II

ROOT MEAN SQUARED RESIDUUM OF A TRACKED OBJECT POINT.

points avg. distance	`Kinect` *556.4 mm*		`rc_visard` *778.5 mm*	
no. of crashtest-markers in view	pixel	estimated overall ADD	pixel	estimated overall ADD
0	2.065	3.358 mm	3.024	4.350 mm
1	2.077	3.377 mm	1.626	3.480 mm
2	1.850	3.275 mm	0.815	2.866 mm

This holds especially if one uses $2\,cm$ as accuracy threshold like eg. [25], [17] do.

VI. Conclusions

We have presented a dataset of realistically packed crockery items in kitchen cupboards, drawers and dishwashers annotated with 6d object poses. This dataset is meant to foster household robotics research as it exhibits characteristics that make this situation challenging and interesting. The ground truth poses are accurate to $\approx 3.5\,\text{mm}$ ADD.

The annotation process works by locating markings on the objects with a handheld and tracked smartphone to obtain the object in world pose which is then converted into an object in camera pose by tracking the camera. This way an object needs to be annotated only once, as long as it is not moved, which greatly reduces the annotation effort.

ACKNOWLEDGMENT

We thank Roboception for providing the rc_visard camera.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition 2009*. Ieee, 2009.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[3] M. Beetz, "A roadmap for research in robot planning," *European Network of Excellence in Artificial Intelligence Planning (PLANET). Technical Coordination Unit for Robot Planning.*, 2003. [Online]. Available: http://planet.dfki.de

[4] R. Dillmann, "Teaching and learning of robot tasks via observation of human performance," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 109–116, 2004.

[5] C. Rennie, R. Shome, K. E. Bekris, and A. F. De Souza, "A dataset for improved rgbd-based object detection and pose estimation for warehouse pick-and-place," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 1179–1185, 2016.

[6] C. Wellhausen, "Entwicklung eines teilautomatisierten systems zur bestimmung von ground-truth posen teilweise verdeckter objekte," Master's thesis, Universität Bremen, 2019, (German).

[7] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[9] D. Damen, H. Doughty, G. Maria Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, *et al.*, "Scaling egocentric vision: The epic-kitchens dataset," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 720–736.

[10] C. Mason, M. Meier, F. Ahrens, T. Fehr, M. Herrmann, F. Putze, and T. Schultz, "Human activities data collection and labeling using a think-aloud protocol in a table setting scenario," in *IROS 2018: Workshop on Latest Advances in Big Activity Data Sources for Robotics & New Challenges, Madrid, Spain*, 2018.

[11] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Asian conference on computer vision*. Springer, 2012, pp. 548–562.

[12] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6d object pose estimation using 3d object coordinates," in *European conference on computer vision*. Springer, 2014, pp. 536–551.

[13] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[14] T. Hodan, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, and X. Zabulis, "T-less: An rgb-d dataset for 6d pose estimation of texture-less objects," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 880–888.

[15] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 36–52, 2015.

[16] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, "Analysis and observations from the first amazon picking challenge," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 172–188, 2018.

[17] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," *arXiv preprint arXiv:1809.10790*, 2018.

[18] P. S. Rajpura, H. Bojinov, and R. S. Hegde, "Object detection using deep cnns trained on synthetic images," *arXiv preprint arXiv:1706.06782*, 2017.

[19] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 23–30.

[20] J. Tremblay, T. To, and S. Birchfield, "Falling things: A synthetic dataset for 3d object detection and pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2038–2041.

[21] T. To, J. Tremblay, D. McKay, Y. Yamaguchi, K. Leung, A. Balanon, J. Cheng, and S. Birchfield, "NDDS: NVIDIA deep learning dataset synthesizer," 2018, https://github.com/NVIDIA/Dataset_Synthesizer.

[22] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[23] T. Wiedemeyer, "Iai kinect2," 2014 – 2015, accessed September 21, 2018. [Online]. Available: https://github.com/code-iai/iai_kinect2

[24] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[25] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," *arXiv preprint arXiv:1901.04780*, 2019.