Towards Autonomous Data Annotation and System-Agnostic Robotic Grasping Benchmarking with 3D-Printed Fixtures

Wout Boerdijk[1,2], Maximilian Durner[1,2], Ryo Sakagami[1], Peter Lehner[1] and Rudolph Triebel[1,3]

***Abstract*— The interaction of robots with their environment requires robust object-centric perception capabilities, typically achieved using learning-based methods trained on synthetic data. However, real-world deployment demands evaluating these capabilities in relevant environments, often involving extensive manual annotation for a quantitative analysis. Additionally, standardized evaluations for robotic tasks, such as grasping, need reproducible object scene configurations and performance benchmarks. We propose a solution to both problems by temporarily employing 3D-printed components, so-called *fixtures*, which can be designed for any rigid object. Once the scene is set up and object poses are extracted, the fixtures are removed, leaving the natural scene without any artificial distractions. The presented approach is seemingly applicable for pre-determined configurations of multiple objects, which enables precise re-building of scenes with consistent object-to-object relations. Our suggested annotation procedure achieves strong pose accuracy solely on RGB images without any manual involvement. We evaluate and show the usability of the proposed fixtures for automated real-world data annotation to fine-tune a detector and for benchmarking object pose estimation algorithms for robotic grasping. Code and fixture meshes for 3D printing are available at https://github.com/DLR-RM/fixture_generation.**

Fig. 1: Removable fixtures (black structures with tag, fading from left to right) enable direct object pose estimation for real-world data (colored overlays) for both single objects and object configurations.

I. INTRODUCTION

International robotic research is evaluated with different robots, sensors and scenarios, making unbiased comparisons across systems often a challenging task. Several research areas try to compensate this by introducing isolated benchmarks on datasets (e.g., perception) or in simulation (e.g., path-planning, reinforcement learning). However, these efforts only partially address the complexity of robotics, including module interactions (e.g., imperfect inputs) and real-world challenges (e.g., sensor noise). For better comparability, Calli et al. [1] presented the *YCB Object and Model Set* with the intention of "*to be used for benchmarking in robotic grasping and manipulation research*". Their work and follow-ups [2], [3] have greatly stimulated algorithm comparisons, leading to valuable advances in the fields around object manipulation. Yet, a thoroughly comparable evaluation of complete object manipulation pipelines, with a focus on perception and its impact, remains challenging. One of the primary difficulties is the precise reproducibility of object scenes across different research facilities. Relative object placement plays a crucial role in the performance of perception algorithms – whether objects are cluttered or spaced apart can significantly affect their success. Small variations in object poses can greatly influence the behavior of grasping strategies. Hence, precise re-building of scenes is also essential for testing individual modules in isolated fashion with ground truth inputs or integrated with input noise.

Beyond benchmarking, a critical work block in realizing actual robotic applications is validating pipeline modules within the current set-up. Obtaining 6D poses of custom objects without affecting the task set-up is essential, yet generating accurate ground truth labels is time-consuming and often almost impossible due to the lack of a widely applicable approach requiring minimal manual effort.

This work addresses both usages, robotic benchmarking and real-world validation, and proposes an autonomous method for real-world 6D pose annotations by leveraging 3D-printable fixtures (see Fig. 1). These fit tightly on any rigid object and incorporate a fiducial marker for autonomous pose derivation solely from RGB imagery. Before executing a grasping pipeline, the fixtures are removed to reveal the natural scene again. The design of such fixtures can be carried out automatically for almost any rigid object in *BlenderProc* [4] and satisfies the following important constraints: (1) fixtures are uniquely connected to the object such that the relative pose between fixture and object is completely defined; (2) they are detachable without changing the object

[1]Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Wessling, Germany <first>.<second>@dlr.de
[2]Department of Computer Science, Technical University of Munich (TUM), 85748 Garching, Germany
[3]Department of Informatics, Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe

 DOI: 10.1109/ICRA55743.2025.11127593

pose; and (3) the underlying concept is transferable to whole scenes, taking inter-object relationships into account.

In summary, our core contributions are as follows:

- We propose a framework to autonomously derive object poses of real-world recordings, supporting both individual objects and inter-object relations, as well as rebuilding of scenes irrespective to the environment.
- We investigate the annotation quality of our method and carry out a detailed evaluation of potential error sources and their contribution.
- We showcase the applicability of fixtures by designing a set of ten scene fixtures for YCB objects, and present a robotic-aided data collection for fine-tuning as well as a reproducible object pose estimation benchmark for robotic grasping.

II. RELATED WORK

A. 6D Pose Annotation Methods for Real-World Image Data

Labeling methods can broadly be categorized into *manual* and *(semi-) autonomous*.

For manual labeling, one or multiple human annotators create all required annotations, often utilizing off-the-shelf labeling tools (e.g., [5]) and resulting in a very time-consuming procedure: The annotation of a single object for the *Pascal VOC* dataset [6] lasted on average 61 seconds and for the *MS COCO* benchmark [7] even 79 seconds. In the *Youtube-VOS* dataset [8], merely every fifth frame was labeled, which is a common strategy to reduce annotation workload. Additionally, human annotations are arguably subjective: To decrease incorporated human biases, some datasets (e.g. [6], [9]) average multiple independent annotations for a single instance.

Semi-autonomous annotation comprises methods aiding human annotators to a certain extend. These workflows involve labeling a small subset of data and refining a neural network to generate annotations for a larger set [10], [11]. Other techniques, like *DEXTR* [9] and *SAM* [12], create instance masks from simple user inputs such as points or scribbles. *SAM* even allows auto-generated annotations, with users only needing to add class labels. Pairing such tools with open-set detectors like [13] can further automate 2D image annotations.

Manual strategies are mostly limited to 2D ground truth, while semi-autonomous approaches enable 6D pose annotations, a much more difficult task. To reduce annotation efforts, a sequence of images is recorded that allows object poses to be determined once and propagated over frames. *HOPE* [14] annotates objects via manually identifying point correspondences between images and 3D textured object models, and utilize a PnP with RANSAC approach for RGB imagery, while employing alignment by procrustes for RGB-D annotations. [15] take a similar approach for the task of car pose estimation: Users select a set of keypoints to initialize an EPnP algorithm which estimates the car pose by minimizing the re-projection error. In *TUD-L/TYO-L* [3], CAD models are manually aligned, and poses are propagated using ICP by reversing frames. In the YCB-Video dataset [2], for every sequence the first frame was manually annotated and then refined via the objects' Signed Distance Function representation in the depth frame.

To automate the recording process several approaches utilize a turn-table with attached markers for camera registration which allows the optimization over multiple frames. In *T-LESS* [16], ground truth is derived by creating a 3D model from such RGB-D image sequences, manually fitting CAD models, and refining misalignments. Their reported average error is less than 9 mm in object depth. Similar concepts are applied for the *Linemod Occluded* [17] dataset and the *ITODD* [18].

Besides the turn-table, other approaches move the camera around the object which also results in varying backgrounds. The *HomebrewedDB* [19] determines the poses by applying an edge-based ICP on RGB images initialized via [20] (reported error under 2 mm). In [21] over 10,000 frames of 25 objects from the Amazon Picking Challenge with a camera mounted on a robotic arm were collected and annotated. In [22], a robot arm moves the camera across different hemispheres around the scene, with poses annotated using a depth-based ICP method across all scene views.

Opposingly, our proposed method does not require any manual involvement, specific setup like a turn table, or post-processing for object pose annotation.

B. Robotic Benchmarking

Another related field covered by our proposed method is benchmarking of object manipulation related capabilities. Given the challenges of reproducibility in the real world, testing in simulation offers an effective alternative. Recently, several simulation benchmarks validating reinforcement learning methods have been introduced [23], [24]. However, such approaches face drawbacks such as sensor noise, lighting and object attribute variations creating a sim-to-real gap that complicates performance assessment in real-world environments.

Hence, there is a high demand for real-world robotic benchmarks. Multiple concepts for different tasks such as assembly [25], [26], grasping [27], [28], [29], or rearranging [30] have been proposed. Nevertheless, reproducibility can only be achieved by a subset, namely FurnitureBench [25], GRASPA [27], NIST Assembly [26], SceneReplica [28] and the BURG-Toolkit [31]. The NIST benchmark provides a fixed set of assembly boards. GRASPA reproduces scenes by creating a fiducial marker grid with the 2D projected objects' outlines. Both SceneReplica and BURG obtain pre-defined scenes by guiding the user's object placement via rendered overlays in the RGB image. This requires extrinsic calibration since the transformation from table surface to camera is necessary for physically plausible object placement in simulation.

In contrast, our method is not restricted to a known camera pose since the object pose is estimated directly via the marker on the fixture.

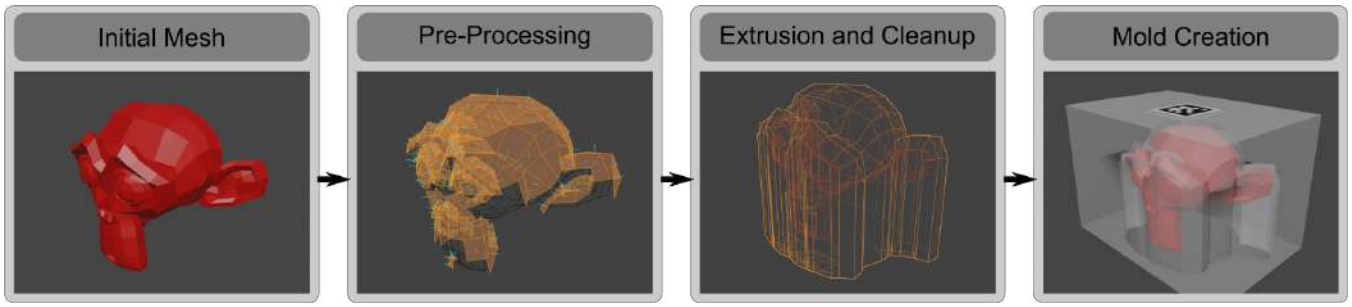

Fig. 2: Top-down fixture creation process: An initial mesh (left) is pre-processed (turquoise lines show the positive normals, second image) and extruded (third image; the initial mesh's edges are marked in red). Finally, a fixture with a tag position is created (right). Note that this is an idealized visualization, and the fixture can be modified further to e.g. reduce material.

III. METHODOLOGY

The core idea of our method is to automatically obtain a static transformation from a detectable fiducial marker to the object(s) of interest. We do so by designing a 3D-printable structure which can be rigidly connected to the object(s) by different means, which we refer to as *fixture*. While it is possible to design such a fixture in a CAD program, we propose to automatize the process with BlenderProc [4]. In this work, three different types of fixtures are presented depending on the topology of the object: *top-down* fixtures for almost any kind of rigid object, *sliders* for box-like objects and *cylindric toppers* for cylindric shapes (see Fig. 3). Fixtures can be extended from single-object fitting to capturing a complete scene comprised of multiple objects, thereby defining inter-object relationships, i.e. the relative pose between the items (visualized in Fig. 4). This enables large-scale labeled data collection as well as reproducible object pose estimation benchmarks for robotic grasping, as highlighted in Sec. VI.

A. Top-Down Fixture Creation

A simple downward extrusion would suffice for many artificially generated meshes. Yet, in practice, models can be derived from scanning real-world objects, resulting in non-watertight shapes, duplicated faces, invalid normals and similar artifacts. This creates the need for a more sophisticated approach, which is visualized in Fig. 2.

First, the initial mesh is adapted by a solidifying operator, which enlarges the mesh along its face normals. This ensures a proper fit of the fixture by compensating for the limitation of printer's resolution and accuracy as well as possible deviations of the object mesh to the objective reality.

Second, further pre-processing is applied to cope with scanned artifacts and computation time. Aside duplicate vertex removal the mesh is reduced to the part encapsulated by the fixture with a standard boolean operator, hereby reducing runtime drastically. Faces completely lying under the object surface – detectable by the fact that all their vertices hit the inside during up-projection – are removed, since these would cause unwanted intersections.

Third, the main algorithm identifies vertices to be down-projected by selecting edges sharing only a single face, indicating that these are at the outer bottom boundary of the mesh. Vertices are down-projected until a satisfactory height of the fixture is reached, a new edge is added between every down-projected vertex pair as well as an additional face for every quadruple. The mesh is then closed at the bottom part, resulting in a watertight object. Optional post-processing cleans up loose parts and removes undesired inside geometry.

Finally, the fixture is created by a simple boolean difference operation between an encapsulating cube and the extruded mesh. Last but not least, the place for the fiducial marker is denoted by a small cut-out on the top side so that the object-to-tag transformation can be derived from the model; a process which is similarly done for all other fixtures.

B. Slider and Cylindric Fixture Creation

Slider and cylindrical fixtures are parameterized by the bounding box of the object, thus can be generated automatically. The former is a cut-out rectangle with one open side, allowing to be slid around the bottom part of an object. A platform for the fiducial marker is connected on one of the three sides of the fixture. The latter is a rotation-invariant mold which can be placed in top-down fashion on cylindrical objects. The cut-out is determined by the diameter of the object, and the marker is placed atop the fixture.

C. Scene Fixture Creation

Scene fixtures aim to capture inter-relationships between a set of objects, and allow precise re-building of object scenes. To this end, the desired objects are loaded into Blender, and manually placed at desired positions. Depending on the accuracy of the CAD models, physical simulation can be applied for a more realistic result. Fixture creation can then be performed by utilizing methods (or parts thereof) presented in the previous two subsections. Depending on the scene's complexity, it can be more efficient to describe its accompanying fixture without any automation by merely manually selecting different modifiers in Blender. This is mainly due to computational restrictions: the algorithm presented in Sec. III-A is greatly influenced by the number of vertices of the mesh, and for scanned models potential artifacts increase runtime during cleanup. The location of the fiducial marker is determined manually to optimize visibility from different camera angles.

D. Annotation

To detect the fiducial markers in the (rectified) image plane we use the *AprilTag*[1] library. The object pose can then be retrieved given the intrinsic parameters of the camera and the fixed transformation from a marker to the object model established during fixture modeling. While being outside of the scope of this work, higher pose accuracy could be achieved by employing multiple markers per object or utilizing fixed inter-object relationships with connectors, and subsequently modeling and optimizing relative tag poses as proposed in [32].

IV. EXPERIMENTAL SETUP

In the scope of this paper, we perform two main experimental evaluations: (1) We evaluate the annotation quality of our method by comparing actual depth and rendered depth values in Sec. V; and (2) we showcase the applicability of our fixtures for fine-tuning an object detector on automatically derived real-world annotations and for an exemplary robotic grasping benchmark in Sec. VI.

All following experiments are performed on 18 objects[2] of the YCB Video dataset [2]. We carry out robotic applications presented in the scope of this work on the ***S**afe **A**utonomous **R**obotic **A**ssistent (*SARA*)* system[3]. It is a 7-axis robotic arm with a stereo camera mounted at the sixth joint of the robot employed as recording sensor, consisting of a RGB (left) and monochrome (right) Ximea sensor. The right image is solely used for Semi-Global Matching [33] in order to derive depth imagery for the quantitative error evaluation in Sec. V. Images are recorded at a resolution of 2056 x 2464 pixels.

For the subsequent assessment of annotation accuracy, we design 18 single-object molds for every YCB object (visualized in Fig. 3): four cylindrical, seven sliding, and six top-down fixture types. Only for the mug manual design was more sufficient due to its subpar scanned object model (Fig. 3 right). For both applications presented in Sec. VI, we design a set of ten scene fixtures (see Fig. 4) capturing pre-defined configurations of objects of the YCB-Video dataset in two difficulties: *easy* - objects are not allowed to touch each other, and *difficult* - stacked scenes potentially resulting in heavy occlusions. The scenes include three to five objects each and are designed without any specific bias in mind except that each of the 18 objects appears once in both difficulties.

Every fixture is printed on a Prusa MK4 3D printer with a 0.4 nozzle, 0.1mm fast detail print settings and standard PLA filament. Markers are printed on sticky labels with a size of 4 x 4 cm.

V. EVALUATION OF ANNOTATION ACCURACY

For every of the 18 YCB objects we record two different views (approximately 90° rotation between both) and two view points with varying distance / elevation angle, resulting in 144 images in total. For every view four images are recorded: The first two recordings are taken with the fixture attached to evaluate marker detection quality. Before the third take, the fixture is manually removed in order to have a clean RGB and depth map. Finally, the last image is taken after re-attaching the fixture to simultaneously evaluate pose offsets between fixture removal and re-attachment.

TABLE I: Annotation accuracy across two cameras and two configurations. The average mean depth is around 6 mm.

Camera	↔ [m]	∠ [°]	μ_δ	σ_δ	$\mu_{\lvert\delta\rvert}$	$med_{\lvert\delta\rvert}$
Ximea	0.5	80	5.09	13.16	11.20	9.32
	1.0	50	6.37	14.50	12.57	10.49
Manta	0.5	80	5.67	6.26	7.00	5.96
	1.0	50	7.37	5.39	7.97	7.57
Mean			6.23	9.98	9.31	7.51

TABLE II: Contribution of camera noise and fixture removal on the pose error. Both mean translation and rotation errors are below 1 mm and 1°, respectively.

Error Type	Camera	Translation [mm]		Rotation [°]	
		μ	σ	μ	σ
Detection Accuracy	Ximea	0.33	0.28	0.24	0.19
	Manta	0.22	0.18	0.14	0.16
Removal Error	Ximea	0.76	0.61	0.79	0.66
	Manta	0.94	0.93	0.69	0.40

To assess the quality of annotations derived from the fixtures, we follow previous work ([16], [19]) and report statistics on differences between captured and rendered depth images - specifically, mean and standard deviation as well as the mean and median of the absolute difference. Similarly to the seminal works, we remove depth differences exceeding 5 cm and consider them as outliers. Aside the Ximea sensors mounted on the SARA system, we also evaluate on an Allied Vision Manta GigE RGB camera with a recording resolution of 1504 x 2056 pixels to highlight the sensoric independence of data annotation for our method. As listed in Tab. I, the generated annotations have an average depth offset of around 6 mm depending on distance and elevation, indicating precise object pose annotations and an improvement to the reported 9 mm accuracy of TLESS [16].

An important premise of our method is reliable tag detection and manual removal of the fixture without distorting the object pose for a clean, precisely annotated image of the object; both listed in Tab. II. While the former is mostly influenced by camera noise and is well below half a millimeter, the latter error highlights that the proposed approach allows fixture removal without distorting the object pose more than a millimeter in translation and a single degree in rotation, on average. Note that we remove cylindrical objects from the *Removal Error* evaluation since the z-rotation is not defined.

[1]https://april.eecs.umich.edu/software/apriltag.

[2]During time of research, the objects *004_sugar_box*, *009_gelatin_box* and *040_large_marker* could not be retrieved. Note that given their shape, fixture creation for the first two is similar to other box-like objects (e.g. *003_cracker_box*), while the marker is similar to other cylindric objects (e.g. *002_master_chef_can*).

[3]https://www.dlr.de/rm/en/sara

Fig. 3: Left: The 18 single fixtures for all objects used in this study: four cylindrical (e.g. *002_master_chef_can* on the top right), seven sliding (e.g. *006_mustard_can* on the left) and six top down (e.g. the blue *019_pitcher_base* in the top-middle). Right: Scanned artifacts in the inside of the mug.

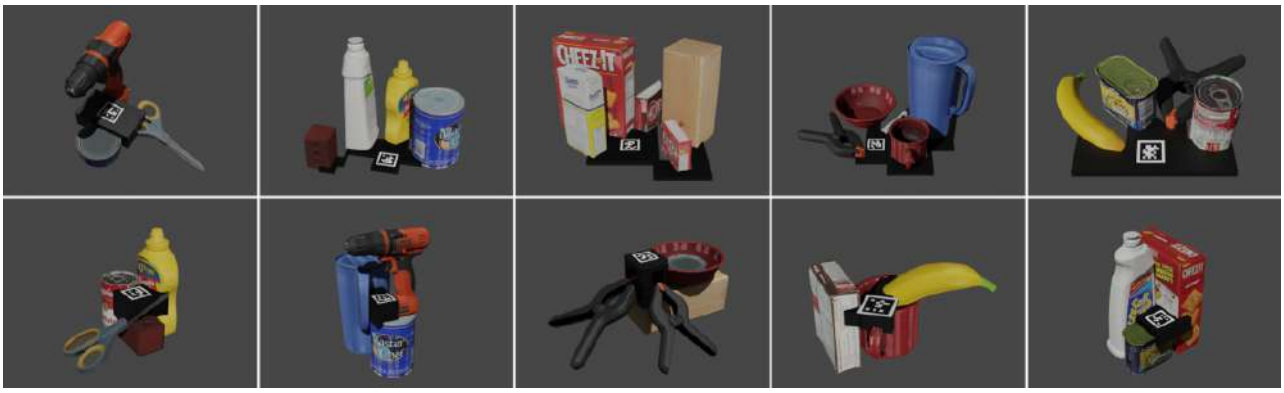

Fig. 4: Five easy (top) and difficult (bottom) scene fixtures. Fixtures allow to handle complicated interactions between objects, such as two clamps leaning on a wood brick (middle image in the bottom row) a banana in a mug (second to last image in the bottom row).

VI. APPLICATIONS

A. Dataset Generation

The proposed fixture annotation framework is well suited to be employed for dataset recording with a robotic agent: After initial object setup and fixture placement, the robot records one image with a fixture on the desired items. Then, the user removes the fixture, and the robot collects various views by following a pre-defined trajectory. The recorded images can be automatically annotated by propagating object poses through the trajectory and then be leveraged as training data for fine-tuning.

We highlight this by evaluating a Yolov7 [34] detector on the STIOS dataset [35], once trained with synthetic data only, and a second time after fine-tuning it on automatically collected data. To this end, SARA collects 20 images of each fixture scene, resulting in 200 annotated images and 40 annotations per object, as visualized in Fig. 5. While stemming from a different recording sensor, the reported results in Tab. III show that model fine-tuning with data collected in the application environment is beneficial for an algorithm's performance. Particularly notable is the *recall* increase across both sensors, indicating that the detector was able to successfully identify more objects after fine-tuning than before. This is also visible in Fig. 6.

TABLE III: Mean Average Precision (mAP) [%] on STIOS before and after fine-tuning with real-world data.

Data	Sensor	Prec.	Rec.	mAP@0.5	mAP@0.5:0.95
Synth.	rc_visard	0.982	0.742	0.746	0.629
	Zed	0.963	0.934	0.933	0.794
Synth.+ Finet.	rc_visard	0.985	0.938	0.94	0.824
	Zed	0.964	0.95	0.956	0.839

B. Exemplar Benchmarking of an Object Pose Estimation Pipeline for Robotic Grasping

Aside accurate ground truth annotations, fixtures can also capture inter-object relations if designed accordingly. This allows re-building object sets with same relative poses between the objects, and is particularly beneficial for a fair evaluation of Computer Vision algorithms in real-world robotic applications. As an example, we showcase this by benchmarking an object pose estimation pipeline on robotic grasping success. The algorithm consists of the same Yolov7 detector employed in Sec. VI-A trained on the BOP YCB dataset[4] according to the training schedule provided by

[4]https://bop.felk.cvut.cz/datasets/

Fig. 5: Left: Process of data recording on the SARA system, note the removed fixture in the bottom image. Right: Bounding box annotations for various scenes.

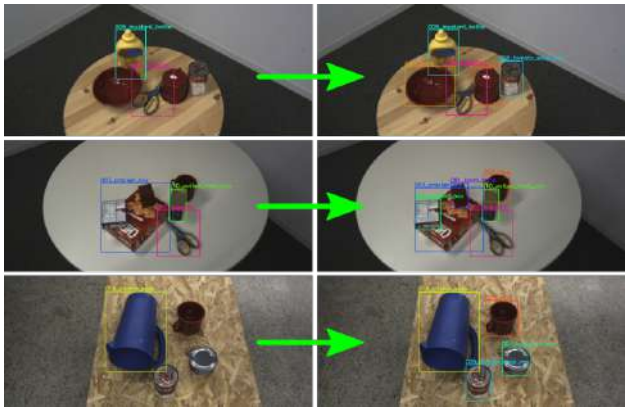

Fig. 6: Qualitative improvements on STIOS data between synthetic training data (left row) and after fine-tuning with real data (right row).

the authors, and a CosyPose [36] pose estimator[5]. For a thoroughly evaluation, we also investigate the influence of the detector by (a) performing pose estimation with ground truth bounding boxes derived from fixture annotations, and (b) listing the number of correctly detected items. For completeness, grasping is also performed given the ground truth annotations derived from the fixtures. The grasps are calculated in advance for each object, and the order of grasping is pre-determined. Results of a single grasping trial are listed in Tab. IV and show a grasping success rate of above 55%, with grasping failures attributed to imprecise pose estimation.

Note that the focus of this study is to highlight the potential of fixtures for reproducible grasping evaluation and benchmarking in a robotic context, rather than the thorough evaluation of the employed algorithms.

[5]https://github.com/ylabbe/cosypose

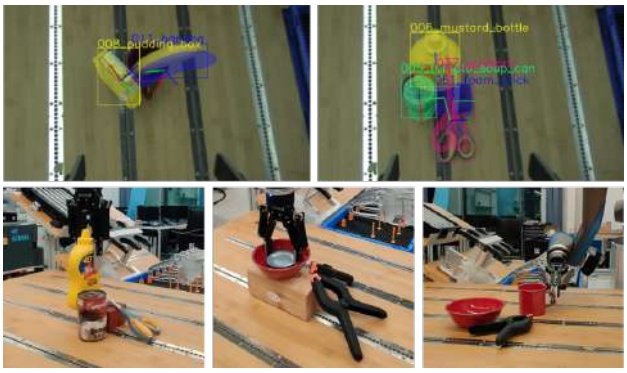

Fig. 7: Top: Two failure cases for grasping stacked scenes with *Cosypose + Yolo*. Left: The mug is not detected, and the rotation of the banana is flipped. Right: The pose of the scissors is shifted. Bottom: Exemplar successful grasps.

TABLE IV: Grasping evaluation from grasps generated by CosyPose and CosyPose + Yolov7 versus those derived from fixture annotations.

Easy Scenes	Detected Objects	Grasped Objects
CosyPose	*n/a*	11
CosyPose + Yolov7	17	10
Fixture Poses	18	18
Hard Scenes	Detected Objects	Grasped Objects
CosyPose	*n/a*	10
CosyPose + Yolov7	15	9
Fixture Poses	18	18

VII. CONCLUSION

We presented a toolkit to automatically generate object pose annotations from RGB images by employing 3D-printable structures. Our method operates independently of depth imagery, extrinsic calibration or any other setup constraints, and does not require any form of post-processing. We highlighted the applicability of our method for robot-aided autonomous labeling in order to increase the performance of Computer Vision algorithms on real-world data. A key feature is the possibility of connectivity between different fixtures, enabling accurate re-building of specific object configurations including stacked items, thus leveraging the toolkit for evaluations such as autonomous object grasping outside simulation. We believe that this is an important premise for unbiased benchmarking on arbitrary real-world robotic systems.

ACKNOWLEDGMENT

The authors would like to thank Marcus G. Müller for expertise in Blender, David L. Risch and Konstantin M. Eisebitt for conceptual realizations, and Zoltán C. Márton for enriching discussions. This work was partly funded by the ”Robotics Institute Germany”.

REFERENCES

[1] B. Calli, A. Singh, A. Walsman *et al.*, "The YCB object and Model set: Towards common benchmarks for manipulation research," in *2015 International Conference on Advanced Robotics (ICAR)*, Jul. 2015, pp. 510–517.

[2] Y. Xiang, T. Schmidt, V. Narayanan *et al.*, "PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes," *arXiv:1711.00199 [cs]*, May 2018.

[3] T. Hodan, F. Michel, E. Brachmann *et al.*, "BOP: Benchmark for 6D Object Pose Estimation," *arXiv:1808.08319 [cs]*, Aug. 2018.

[4] M. Denninger, M. Sundermeyer, D. Winkelbauer *et al.*, "BlenderProc," *arXiv:1911.01911 [cs]*, Oct. 2019.

[5] "LabelMe: Image Annotation and Labeling Toolkit," https://labelme.csail.mit.edu, accessed: September 10, 2023.

[6] M. Everingham, L. Van Gool, C. K. I. Williams *et al.*, "The Pascal Visual Object Classes (VOC) Challenge," *Int. Journal of Computer Vision*, vol. 88, pp. 303–338, Jun. 2010.

[7] T.-Y. Lin, M. Maire, S. Belongie *et al.*, "Microsoft COCO: Common Objects in Context," *arXiv:1405.0312 [cs]*, Feb. 2015.

[8] N. Xu, L. Yang, Y. Fan *et al.*, "YouTube-VOS: A Large-Scale Video Object Segmentation Benchmark," *arXiv:1809.03327 [cs]*, Sep. 2018.

[9] R. M. Swan, D. Atha, H. A. Leopold *et al.*, "AI4MARS: A Dataset for Terrain-Aware Autonomous Driving on Mars," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2021, pp. 1982–1991.

[10] B. Adhikari, J. Peltomäki, J. Puura *et al.*, "Faster Bounding Box Annotation for Object Detection in Indoor Scenes," in *7th European on Visual Information Processing Workshop (EUVIP)*, Nov. 2018, pp. 1–6.

[11] P. Voigtlaender, M. Krause, A. Osep *et al.*, "MOTS: Multi-Object Tracking and Segmentation," *arXiv:1902.03604 [cs]*, Apr. 2019.

[12] A. Kirillov, E. Mintun, N. Ravi *et al.*, "Segment anything," *arXiv:2304.02643*, 2023.

[13] S. Liu, Z. Zeng, T. Ren *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[14] S. Tyree, J. Tremblay, T. To *et al.*, "6-DoF Pose Estimation of Household Objects for Robotic Manipulation: An Accessible Dataset and Benchmark," Dec. 2022, arXiv:2203.05701 [cs].

[15] X. Song, P. Wang, D. Zhou *et al.*, "ApolloCar3D: A Large 3D Car Instance Understanding Benchmark for Autonomous Driving," Nov. 2018, arXiv:1811.12222 [cs].

[16] T. Hodan, P. Haluza, S. Obdrzalek *et al.*, "T-LESS: An RGB-D Dataset for 6D Pose Estimation of Texture-less Objects," Jan. 2017, arXiv:1701.05498 [cs].

[17] S. Hinterstoisser, V. Lepetit, S. Ilic *et al.*, "Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes," in *Computer Vision – ACCV 2012*, Berlin, Heidelberg, 2013, vol. 7724, pp. 548–562.

[18] B. Drost, M. Ulrich, P. Bergmann *et al.*, "Introducing MVTec ITODD — A Dataset for 3D Object Recognition in Industry," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, Venice, Italy, Oct. 2017, pp. 2200–2208.

[19] R. Kaskman, S. Zakharov, I. Shugurov *et al.*, "HomebrewedDB: RGB-D Dataset for 6D Pose Estimation of 3D Objects," Sep. 2019, arXiv:1904.03167 [cs].

[20] B. Drost, M. Ulrich, N. Navab *et al.*, "Model globally, match locally: Efficient and robust 3D object recognition," in *IEEE/CVF on Computer Vision and Pattern Recognition*. San Francisco, CA, USA: IEEE, Jun. 2010, pp. 998–1005.

[21] C. Rennie, R. Shome, K. E. Bekris *et al.*, "A Dataset for Improved RGBD-based Object Detection and Pose Estimation for Warehouse Pick-and-Place," Feb. 2016, arXiv:1509.01277 [cs].

[22] M. Durner, S. Kriegel, S. Riedel *et al.*, "Experience-based optimization of robotic perception," in *IEEE International Conference on Advanced Robotics (ICAR)*, Hongkong, China, July 2017.

[23] T. Yu, D. Quillen, Z. He *et al.*, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on Robot Learning (CORL)*. PMLR, 2020, pp. 1094–1100.

[24] S. James, Z. Ma, D. Rovick Arrojo *et al.*, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters (RA-L)*, 2020.

[25] M. Heo, Y. Lee, D. Lee *et al.*, "Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation," *arXiv preprint arXiv:2305.12821*, 2023.

[26] K. Kimble, K. Van Wyk, J. Falco *et al.*, "Benchmarking protocols for evaluating small parts robotic assembly systems," *IEEE Robotics and Automation Letters*, vol. 5, pp. 883–889, 2020.

[27] F. Bottarel, G. Vezzani, U. Pattacini *et al.*, "Graspa 1.0: Graspa is a robot arm grasping performance benchmark," *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, pp. 836–843, 2020.

[28] N. Khargonkar, S. H. Allu, Y. Lu *et al.*, "Scenereplica: Benchmarking real-world robot manipulation by creating reproducible scenes," *arXiv preprint arXiv:2306.15620*, 2023.

[29] J. Luo, C. Xu, L. Tan *et al.*, "Fmb: a functional manipulation benchmark for generalizable robotic learning," in *CoRL 2023 Workshop on Learning Effective Abstractions for Planning (LEAP)*.

[30] Z. Liu, W. Liu, Y. Qin *et al.*, "Ocrtoc: A cloud-based competition and benchmark for robotic grasping and manipulation," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, pp. 486–493, 2021.

[31] M. Rudorfer, M. Suchi, M. Sridharan *et al.*, "Burg-toolkit: robot grasping experiments in simulation and the real world," *arXiv preprint arXiv:2205.14099*, 2022.

[32] C. Nissler, S. Büttner, Z.-C. Marton *et al.*, "Evaluation and improvement of global pose estimation with multiple apriltags for industrial manipulators," in *IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016, pp. 1–8.

[33] H. Hirschmuller, "Stereo Processing by Semiglobal Matching and Mutual Information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, pp. 328–341, Feb. 2008.

[34] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[35] M. Durner, W. Boerdijk, M. Sundermeyer *et al.*, "Unknown object segmentation from stereo images," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4823–4830.

[36] Y. Labbé, J. Carpentier, M. Aubry *et al.*, "CosyPose: Consistent Multi-view Multi-object 6D Pose Estimation," in *Computer Vision – ECCV 2020*, Cham, 2020, vol. 12362, pp. 574–591.