FruitBin: a large-scale fruit bin picking dataset tunable over occlusion, camera pose and scenes for 6D pose estimation

Guillaume Duret, Mahmoud Ali, Nicolas Cazin, Alexandre Chapin, Florence Zara, Emmanuel Dellandréa, Jan Peters, Liming Chen

FruitBin: a large-scale fruit bin picking dataset tunable over occlusion, camera pose and scenes for 6D pose estimation

Guillaume Duret[1,3]
guillaume.duret@ec-lyon.fr

Mahmoud Ali[1]
mahmoud.ali@ec-lyon.fr

Nicolas Cazin[1]
nicolas.cazin@ec-lyon.fr

Alexandre Chapin[1]
alexandre.chapin@ec-lyon.fr

Florence Zara[2]
florence.zara@liris.cnrs.fr

Emmanuel Dellandrea[1]
emmanuel.dellandrea@ec-lyon.fr

Jan Peters[3]
jan.peters@tu-darmstadt.de

Liming Chen[1]
liming.chen@ec-lyon.fr

[1]**Univ Lyon, Centrale Lyon, CNRS, INSA Lyon, UCBL, LIRIS, UMR5205, F-69130 Ecully, France**
[2]Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, UMR5205, F-69622 Villeurbanne, France
[3]Intelligent Autonomous Systems Lab, Technical University of Darmstadt, 64289 Darmstadt, Germany

Abstract

Bin picking is a widely spread application in industries and its automation through robots generally requires object instance-level segmentation and 6D pose estimation. State-of-the-art computer vision algorithms for these tasks are deep learning-based and require large datasets of diversified annotated images at the instance level, which are prohibitively expensive to acquire. In this paper, we make use of PickSim, a newly developed Gazebo-based dynamically configurable open-source pipeline, and introduce a dataset of simulated data, namely FruitBin, for the challenging task of fruit bin picking. FruitBin contains more than 1M images and 40M instance-level 6D pose annotations over both symmetric and asymmetric fruits with or without texture. Rich annotations and metadata (including 6D pose, segmentation mask, point cloud, 2D and 3D bounding boxes, occlusion rate) allow the tuning of the proposed dataset for benchmarking the robustness of object instance segmentation and 6D pose estimation models (with respect to variations in terms of lighting, texture, occlusion, camera pose and scenes). We further propose three scenarios presenting significant challenges of 6D pose estimation models: new scene generalization; new camera viewpoint generalization; and occlusion robustness. We show the results of these three scenarios for two 6D pose estimation baselines making use of RGB or RGBD images. To the best of our knowledge, FruitBin is the first dataset for the challenging task of fruit bin picking and the biggest large-scale dataset for 6D pose estimation with the most comprehensive challenges, tunable over scenes, camera poses and occlusions.

Submitted to the 37th Conference on Neural Information Processing Systems (NeurIPS 2023) Track on Datasets and Benchmarks. Do not distribute.

Figure 1: Example of 6 different points of view of a single scene of the FruitBin dataset

1 Introduction

Bin picking refers to the process of extracting objects from a bin or container. It is commonly used in various industries, such as manufacturing, logistics, and warehouse. Its automation through robotic systems involves using sensors, computer vision, and robotic arms to identify and retrieve objects from an unstructured and cluttered environment. State-of-the-art solutions are data-driven and imply robot perception to segment object instances and estimate their 6D pose (16; 10). As a result, they generally require large-scale datasets of diversified object instance-level annotations which are prohibitively expensive to acquire.

State-of-the-art benchmarks for 6D pose estimation are computer vision oriented. Therefore, they are not defined within robotic learning software or provide only a partial robotic environment, thereby hindering the latter stage of seamless robot learning for manipulation which further involves learning interactions between robots and objects (5). Furthermore, almost all of them only depict tabletop scenes with rigid objects and don't consider bin picking scenarios where objects typically occur in multiple instances and feature severe occlusions and clutter.

In this paper, we present FruitBin, a large-scale dataset of simulated data suitable for robot learning and specifically designed for the challenging task of fruit bin picking as example can be seen in the figures 1 or 3. And Table 1 compares FruitBin with the state of the art. It builds upon PickSim (7), a recently released open-source bin picking simulation pipeline that offers dynamic configuration capabilities within Gazebo (18), an open-source, 3D robotics simulation software widely used in the field of robotics research and development. FruitBin considers delicate objects, *e.g.*, apricot, banana, whose manipulation requires learning the appropriate force through haptic feedback and paves the way to robotic manipulation of deformable objects. It comprises over 1 million images and 40 million instance-level 6D pose annotations, encompassing both symmetric and asymmetric fruits with and without texture. The dataset includes comprehensive annotations and metadata, such as 6D pose, segmentation masks, point clouds, 2D and 3D bounding boxes, and occlusion rates. This rich set of annotations allows fine-tuning the proposed dataset in order to benchmark the robustness of object instance segmentation and 6D pose estimation models against variations in lighting, texture, occlusion, camera pose, and scene complexity. Furthermore, we propose three scenarios that present significant challenges for 6D pose estimation models: new scene generalization; new camera viewpoint generalization; and occlusion robustness. We evaluated the performance of two 6D pose estimation baselines, using either RGB or RGBD images, on these three scenarios. To the best of our knowledge, FruitBin is the first dataset specifically tailored for the demanding task of fruit bin picking and represents the largest-scale dataset for 6D pose estimation, offering comprehensive challenges that can be adjusted in terms of scenes, camera poses, and occlusions.

In summary, our contributions are as follows:

- release of FruitBin: the bigest large-scale dataset for fruit bin picking with 40M 6D pose annotations over 1M images, depicting comprehensive challenges for 6D pose estimation;
- rich annotations for object instance-level segmentation and 6D pose estimation, including 2D and 3D bounding boxes, instance, segmentation masks, and occlusion rates;
- proposition of three benchmark scenarios to test robustness of 6D pose estimation models in terms of occlusion, camera pose, and scene variety.
- baseline results over these three scenarios for two 6D pose estimation baselines making use of RGB and RGB-D images, respectively.

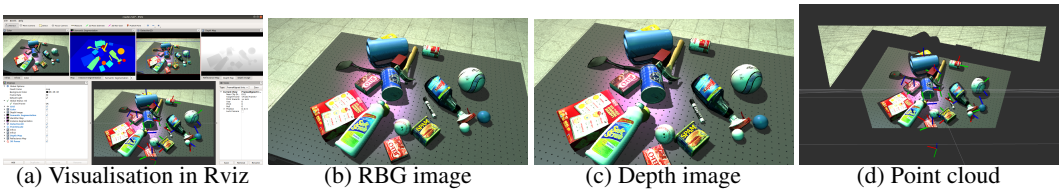
In the following, Section 2 present previous work in 6D pose datasets. Section 3 presents the generation of FruitBin with the software PickSim. Section 4 describes FruitBin with the specific statistics of the large dataset and sub-datasets for the 3 scenarios. Section 5 gathers the result of baselines of 6D estimation models through the 3 scenarios. Section 6 presents some limitations, and Section 7 concludes this article with some perspectives.

2 Related work

Because data famine is the major roadblock for learning-based computer vision tasks and even the more for robot learning. State-of-the-art datasets for 6D pose estimation (6; 11) is quickly over-viewed in this section.

6D pose datasets. State of the art has featured a number of benchmarks for 6D pose estimation. Table 1 lists these datasets and compares them in terms of different characteristics, including data nature (real or synthetic), size (number of samples, number of scenes, number of 6D pose annotations), as well as challenges, *e.g.*, occlusion, clutter, multiple instances, *etc.*

The proposed FruitBin dataset offers high-resolution images and Table 1 demonstrates that it distinguishes itself by incorporating all of the challenges into a single dataset. Notably, it stands out among the limited number of datasets that have significantly increased the sample size, ranging from 2 to 1000 times more samples than existing 6D pose estimation datasets. This expansion in dataset size is crucial as it addresses a critical challenge faced by deep learning models—generalizing to unknown scenes. By incorporating a larger number of diverse scenes, we believe FruitBin enables models to overcome this limitation. With over **40 million** 6D pose annotations, FruitBin not only surpasses other datasets in terms of the number of scenes but also provides the largest number of annotations available.



(a) Visualisation in Rviz (b) RBG image (c) Depth image (d) Point cloud

(e) Depth map (f) Instance map (g) Semantic map (h) 2D Bounding boxes

Figure 2: Examples of annotations

3 Raw data generation process of FruitBin using PickSim

This section introduces the generation of Fruitbin through the use of PickSim (7). This recent pipeline offers extensive annotation generation capabilities as illustrated in the figure 2 and facilitates the

Dataset	Data	Labels	#Samples	#Scenes	#6D pose	Challenges
LINEMOD (12)	Real	Manual	18k	15	15k	C+TL
O-LINEMOD (1)	Real	Semi-auto	1214	15	120k	C+TL+O
APC (26)	Real	Semi-auto	10k	12	~240k	C+L
T-LESS (13)	Real	Semi-auto	49k	20	47k	C+TL+O+MI
YCB-V (29)	Real-S.	Semi-auto	133k	92	613k	O+C
FAT (24)	Synth.	Auto	60k	3	205k	C
BIN-P (17)	Real-S.	Semi-auto	206k	12	20M	SC+SO+MI+BP
CAMERA (28)	Real-S.	Semi-auto	308k	31	4M	O+C
ObjectSynth (14)	Synth.	Auto	600k	6	21M	O+C
HomebrewedDB (15)	Synth.	Auto	17.4k	13	56k	O+C+L
GraspNet-1B (3)	Real	Semi-Auto	97k	190	970k	O+C
RobotP (30)	Synth.	Semi-Auto	4k	-	-	O+TL+L
HOPE (25)	Real	Manual	2k	50	~30k	O+C+MI+L
MetaGraspNet (8)	Real-S.	Auto	217k	6.4k	3M	O+C+MI
SynPick (23)	Synth.	Auto	503k	300	10M	O
StereOBJ-1M (19)	Real	Semi-Auto	396k	183	1.5M	O+L
DoPose (9)	Real	Semi-Auto	3k	301	11k	O+C+BP
FruitBin	Synth.	Auto	**1M**	**70k**	**40M**	**SO+SC+MI+BP+TL+L**

Table 1: Comparison of 6D pose datasets with their different challenges (O: Occlusion, C: Clutter, SO: Severe Occlusion, SC: Severe Clutter, MI:Multiple Instance, BP: Bin Pickings, TL: Texture Less, L: Light).

utilization of the dataset for robotics learning. In order to create task-specific datasets for robotics, it is crucial to develop and train tasks such as vision and manipulation within the same environment. Utilizing robotic software, such as Gazebo, for generating computer vision synthetic data offers numerous advantages. Firstly, it seamlessly integrates physical engines, resulting in more realistic outcomes. Secondly, it facilitates the effortless integration of robots and sensors with native robot control capabilities. Lastly, it unlocks the potential for creating datasets specifically tailored to robotic tasks with various open-source libraries like MoveIt (4). Additionally, PickSim offers easy-to-use setup files for domain randomization, dataset recording, and generation. Each step within the pipeline can be executed effortlessly using a simple command, with parameters conveniently set in user-friendly JSON or YAML files. In this section, we present the generation of FruitBin using the PickSim (7) pipeline into four key steps.

Pre-processing. Raw meshes can be provided to enable mesh domain randomization, encompassing textures, object properties, and more. For FruitBin, PickSim receives eight raw meshes representing the fruits we aim to simulate, namely Apple, Apricot, Banana, Kiwi, Lemon, Orange, Peach, and Pear. No randomization is applied to the mesh or textures in order to preserve the distinct characteristics of each fruit. Through this automated process, sdf files are generated, which are essential for robotic software simulation and contain key metadata such as category ID for the future dataset recording.

Scene randomization. PickSim (7) enables domain randomization (2; 20; 21). This functionality is used to generate diverse scenes for fruit bin picking. Through configuration files, users can easily customize the number of objects, cameras, and lighting conditions without writing additional code, simplifying the creation of randomized Gazebo world files. In the FruitBin dataset, scene randomization encompasses the bin, fruits, and lighting. The bin undergoes randomization with rotations and color variations, while the lighting setup includes randomized positions, intensities, and colors. This deliberate setup ensures significant lighting diversity. To maintain statistical consistency, the number of instances for each category is randomly set between 0 and 30, ensuring a consistently full bin. Examples of these scenes can be seen in Figure 3.

Camera randomization. The final aspect of randomization involves the camera settings, where we employ an orbiter sampler included in PickSim to introduce randomness in the distance (ranging from 0.55m to 1m) between the camera and the orbiter center, as well as different angles to ensure

Figure 3: 30 first scenes for a unique point of view

optimal viewpoints of the scene. This streamlined setup enables the generation of fully randomized scenes that are both physically realistic and ideal for fruit bin-picking scenarios. The impact of these camera parameters can be observed in Figure 1, showcasing five different viewpoints of a scene.

Record data. Simulations in Gazebo can be seamlessly launched using the generated world files. These simulations produce datasets with recorded annotations, including instance and semantic segmentation, bounding boxes, occlusion rates, 6D pose estimations, depth maps, point clouds, and normals.

4 FruitBin: a large scale dataset

4.1 Data and Statistics

For FruitBin with eight fruits, the randomization process is performed 10,000 times with 15 cameras, resulting in 150,000 data frames. This process is repeated seven times. Assembling the 7 parts represent FruiBin with over 1 million frames across 70,000 scenes and 105 camera viewpoints. The dataset is organized systematically, preserving information in metafiles. By dividing the dataset into seven parts, sub-datasets can be created for scene generalization, camera generalization, and occlusion robustness. A comprehensive dataset comparison can be found in Section 2.

To streamline the organization of data for learning 6D pose estimation, we have restructured the data based on features and categories. FruitBin incorporates metadata to store essential information such as scene ID, camera ID, category ID, and instance IDs. These unique IDs enable direct access to instance or category-based annotations such as masks, 2D/3D bounding boxes, 6D poses, and occlusion rates. This occlusion rate offers a notable advantage by enabling visible instance counting in images and facilitating data filtering based on different occlusion rates, as described in Section 4. Figure 4 depicts data statistics and provides insights into the distribution of 6D poses among different fruit categories. Notably, the complete randomization of scenes results in a Gaussian distribution of instance numbers in the images, and ensures a balanced representation of each fruit category.

4.2 A tunable large-scale dataset for fruit bin picking

Developing large-scale, diverse, and accurately annotated datasets for 6D pose estimation is a demanding and time-consuming task. The availability of comprehensive and well-annotated datasets plays a crucial role in advancing the state-of-the-art in 6D pose estimation. However, each dataset, as outlined in Table 1, presents unique challenges that necessitate specific datasets. These challenges include bin picking, scene diversity, point of view diversity, occlusion, and multi-instances, among others. Given that 6D pose estimation finds applications in various contexts, it becomes necessary to fine-tune different 6D models to cater to specific requirements. The suitability of FruitBin as a tunable dataset stems from its abundant annotations and extensive scale. Specifically, the large-scale dataset allows generating sub-datasets tailored to specific purposes or ablation studies. The tunable

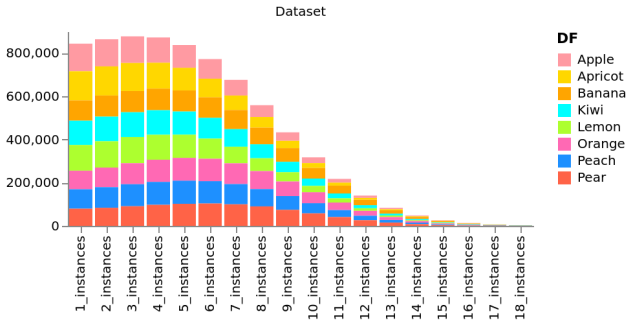


Figure 4: Statistics of the whole dataset with the number of instances for each category in the frames

capability of FruitBin is exemplified through its application in addressing three primary challenges in 6D pose estimation: scene generalization, camera point-of-view generalization, and occlusion robustness.

Camera and scene generalization scenarios. To explore scene generalization and point-of-view camera generalization, 2 scenarios of single-instance 6D pose estimation are created. The idea is to sample the FruitBin dataset to generate scenario oriented dataset. The sampling is from the first part of the dataset consisting of 10,000 distinct scenes and 15 identical camera points of view. In the scene scenario, data is sampled from the large dataset, resulting in 60% representing 6000 scenes with all 15 cameras for training, 20% for evaluation, and 20% for testing, with each part containing a different scene. A similar approach is applied to study the camera scenario, with nine initial points of view allocated for training, three for evaluation, and the last three for testing. During the filtering of the dataset, all the dataset samples are category based to already process the data for the purpose of future 6D pose estimation. During this sample, new metadata is recorded such as all the scenes, camera IDs, and all the category-based instance ID corresponding to including their occlusion rate.

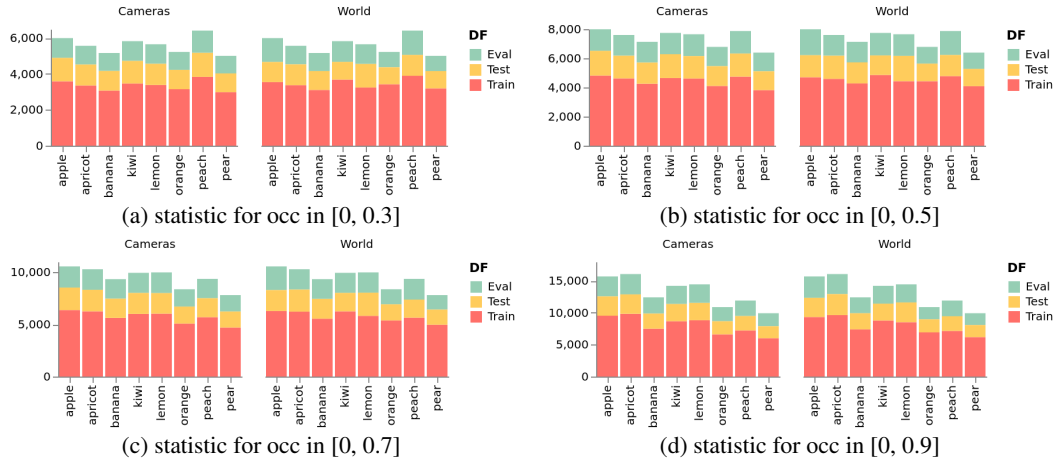
Occlusion robustness scenarios. For the precise study of occlusion robustness, we considered studying as a parameter of the two previous. In practice for the two previous scenarios, 4 levels of difficulty are generated leveraging occlusion rate. During the previous sampling, an occlusion parameter is added; instead of taking all data, a filter will be applied based on the occlusion rate of the object. The first version focuses on objects with occlusion bellow 30%, followed by versions with occlusion going to 50%, 70%, and 90%, respectively, representing the most challenging scenarios. Additional splitting based on occlusion is performed during testing to evaluate occlusion-related performances. For an even more precise study of the impact of the occlusion, the testing part could be split according to a partition for getting an occlusion-aware performance overview as presented in the tables 3.

The figure 5 present the splitting of the different scenario for the 4 occlusion difficulties.

5 Experiments

Baseline Methods In order to estimate the challenges posed by our FruitBin dataset, we conducted a comprehensive assessment using two distinct state-of-the-art 6D pose estimation models employing different data modalities.

The first method, **PVNet** (22), utilizes a RGB image and information from 3D models of the objects as input to extract the 6D pose. This method processes the input image in two stages: firstly, it determines 2D keypoints locations of the objects using a series of convolution and deconvolution blocks, followed by a RANSAC-based voting scheme. The 6D pose is then obtained by solving an uncertainty-driven Perspective-n-Point (PnP) problem given the 2D key points and the 3D model.



(a) statistic for occ in [0, 0.3]

(b) statistic for occ in [0, 0.5]

(c) statistic for occ in [0, 0.7]

(d) statistic for occ in [0, 0.9]

Figure 5: Statistics of the 4 occlusion ranges and two scenarios (scene (world) and camera) with the splitting: training, evaluating and testing.

The second method, **Densefusion** (27), takes as input RGB image and depth information (RGB-D input) along with a semantic mask of the scene. Generally, depth information methods tend to be more robust but require more computational resources. DenseFusion initially extracts a binary mask for each object, which is then used to crop the image and the point cloud within the region of interest (ROI). Each ROI is then used as input to a 2D feature extractor and a point cloud extractor to obtain color and geometry embeddings. These embeddings are then concatenated for each point and fused to extract 'local' and 'global' information, resulting in the final (dense) fused features. The 6D pose is estimated using a pose predictor model that iteratively refines the pose at each step.

Both models have been trained and evaluated on existing 6D pose estimation datasets: LINEMOD and YCB-Video, presented in section 2, and have shown state-of-the-art results.

Metrics The baseline models are evaluated using the ADD metric (12) (*average distance*) for non-symmetrical objects and ADD-S (29) (*average closest point distance*) for symmetrical objects. In the case of FruitBin, Apple, Apricot, Kiwi, Lemon, Orange and Peach objects are considered as symmetrical while Banana and Pear are non-symmetrical. In the following, ADD(-S) refer to both metrics.

ADD is the mean distance of the transformed 3d model points using the estimated pose $[\hat{R}|\hat{t}]$ and ground truth pose $[R|t]$. Based on the computed distance, the estimated pose is considered correct if the distance is less than 10% of the model's diameter. The diameter represents the longest distance between 2 points in the object.

Experiments Using the two baselines described in 5 and the metric presented in 5, we train each of them over the 8 scenarios presented in section 4.2. The result are shown in table 2 for Densefusion and PVNet. For instance, in these tables, "Scenes_occ[0.3,0.4]" refers to the "Scene" scenario with a level of occlusion ranging from 30% to 40%.

Densefusion, which relies on depth information, exhibits superior performance compared to PVNet, which solely utilizes RGB images. However, the effectiveness of Densefusion is heavily influenced by object occlusion. It achieves a success rate of 90% or higher when the object's occlusion is bellow 30%. Conversely, as the occlusion in the data increases, up to 90%, the performance of Densefusion declines, reaching a success rate of 67%. It fails to meet the refinement threshold for this level of occlusion for the camera scenario. Additionally, it is important to note that Densefusion employs ground truth segmentation as an input for specific studies on 6D pose estimation. This observation is further substantiated by the results presented in Table 3, which illustrate the performance across various occlusion ranges. The performance remains satisfactory when objects are slightly occluded,

	Apple	Apricot	Banana	Kiwi	Lemon	Orange	Peach	Pear	Avg
Densefusion									
Scenes_occ[0.0,0.3]	0.997	0.993	0.490	0.991	0.996	1.0	1.0	0.674	0.899
Scenes_occ[0.0,0.5]	0.995	0.950	0.526	0.948	0.956	1.0	1.0	0.636	0.882
Scenes_occ[0.0,0.7]	0.981	0.950	0.414	0.894	0.933	0.997	0.998	0.570	0.849
Scenes_occ[0.0,0.9]	0.844	0.713	0.278	0.656	0.726	0.896	0.903	0.306	0.676
Cameras_occ[0.0,0.3]	0.983	0.872	0.588	0.968	0.957	1.0	0.999	0.669	0.888
Cameras_occ[0.0,0.5]	0.978	0.900	0.592	0.974	0.980	0.999	0.999	0.606	0.887
Cameras_occ[0.0,0.7]	0.983	0.922	0.530	0.887	0.864	0.995	0.997	0.553	0.850
Pvnet									
Scenes_occ[0.0,0.3]	0.505	0.422	0.858	0.501	0.486	0.572	0.640	0.762	0.593
Scenes_occ[0.0,0.5]	0.430	0.432	0.880	0.503	0.473	0.572	0.685	0.793	0.596
Scenes_occ[0.0,0.7]	0.533	0.431	0.879	0.475	0.492	0.581	0.649	0.763	0.600
Scenes_occ[0.0,0.9]	0.445	0.363	0.864	0.487	0.481	0.561	0.621	0.761	0.573
Cameras_occ[0.0,0.3]	0.590	0.516	0.952	0.631	0.594	0.701	0.784	0.862	0.704
Cameras_occ[0.0,0.5]	0.606	0.524	0.941	0.611	0.597	0.693	0.819	0.834	0.703
Cameras_occ[0.0,0.7]	0.577	0.475	0.935	0.602	0.588	0.748	0.773	0.810	0.688
Cameras_occ[0.0,0.9]	0.519	0.447	0.939	0.580	0.568	0.673	0.753	0.827	0.663

Table 2: Success rate result of Dnesefusion and Pvnet model trained on the scene and camera scenarios with the different levels of occlusion.

	Apple	Apricot	Banana	Kiwi	Lemon	Orange	Peach	Pear	Avg
Densefusion									
Scenes_occ[0.8, 0.9]	0.547	0.391	0.026	0.258	0.373	0.514	0.592	0.0	0.360
Scenes_occ[0.7, 0.8]	0.679	0.534	0.018	0.378	0.453	0.747	0.733	0.019	0.465
Scenes_occ[0.6, 0.7]	0.868	0.597	0.079	0.520	0.605	0.913	0.899	0.047	0.570
Scenes_occ[0.5, 0.6]	0.912	0.729	0.108	0.593	0.724	0.967	0.950	0.112	0.646
Scenes_occ[0.4, 0.5]	0.964	0.758	0.242	0.793	0.804	1.0	0.993	0.189	0.731
Scenes_occ[0.3, 0.4]	1.0	0.895	0.305	0.803	0.861	1.0	1.0	0.209	0.771
Scenes_occ[0.2, 0.3]	0.996	0.945	0.418	0.884	0.915	0.995	1.0	0.247	0.817
Scenes_occ[0.1, 0.2]	1.0	0.953	0.455	0.959	0.928	1.0	1.0	0.384	0.841
Scenes_occ[0.0, 0.1]	1.0	0.974	0.557	0.973	0.963	1.0	0.999	0.665	0.898
Pvnet									
Scenes_occ[0.8, 0.9]	0.258	0.220	0.896	0.478	0.557	0.558	0.489	0.723	0.522
Scenes_occ[0.7, 0.8]	0.248	0.286	0.898	0.519	0.498*	0.562	0.502	0.791	0.538
Scenes_occ[0.6, 0.7]	0.393	0.347	0.887	0.511	0.504	0.616	0.486	0.698	0.555
Scenes_occ[0.5, 0.6]	0.397	0.305	0.901	0.450	0.448	0.565	0.521	0.687	0.534
Scenes_occ[0.4, 0.5]	0.412	0.345	0.836	0.458	0.457	0.551	0.500	0.750	0.539
Scenes_occ[0.3, 0.4]	0.500	0.359	0.870	0.508	0.485	0.609	0.601	0.790	0.590
Scenes_occ[0.2, 0.3]	0.508	0.427	0.848	0.468	0.493	0.541	0.616	0.800	0.588
Scenes_occ[0.1, 0.2]	0.570	0.385	0.872	0.482	0.477	0.614	0.745	0.803	0.619
Scenes_occ[0.0, 0.1]	0.604	0.443	0.896	0.458	0.478	0.557	0.749	0.805	0.624

Table 3: Precise evaluation of Dense-fusion and PVnet trained on "$Scenes\_occ[0.0, 0.9]$" over the testing partitions of level of occlusion

achieving a success rate of 90% with occlusion levels below 10%. However, the success rate drops significantly to 30% when considering occlusion between 80% and 90%.

On the contrary, PVNet exhibits distinct characteristics. The method relies on keypoints, which reduces its dependence on occlusion. It demonstrates consistent performance across different scenarios, yielding an average result of 64% (tables 2 and 3). Notably, there are variations in performance based on the object, with notable peaks for pear, banana, and peach, which possess more intricate textures. Conversely, other objects in the dataset can be considered texture-less, lacking any distinctive texture specifications. Another important aspect to consider is that PVNet relies on pixel values, while our dataset may exhibit varying distances between the camera and the objects. This can result in a limited number of pixels available for inferring key points and subsequently estimating the 6D pose.

These experiments showcase the significant advantages of our dataset, enabling us to present formidable challenges to state-of-the-art models that typically excel within the upper echelons of standard datasets.

6 Limitation and future work

This work aims to introduce a benchmark for 6D pose estimation by presenting a dataset specifically designed for this purpose. However, it is important to acknowledge that our dataset has limitations in terms of fruit meshes. Given the nature of fruits, where each instance is unique, there is a need to expand the dataset to include category 6D pose estimation. In order to address this, a logical step would be to incorporate new vision annotations into the open-source software PickSim NOCS (28), which is widely utilized for category 6D pose estimation. This addition would enhance the dataset and enable comprehensive evaluation and analysis of category 6D pose estimation methods.

Furthermore, addressing the sim2real gap between our simulator and real fruits is an important aspect to consider. We recognize the need for extensive studies on domain randomization techniques to bridge this gap effectively, particularly in the context of robotics applications. By investigating and refining domain randomization methods, we can enhance the simulation realism and improve the transfer-ability of models trained in the simulator to real-world scenarios. These efforts will contribute to advancing the field of robotics and promoting the seamless integration of simulated environments with real-world applications.

7 Conclusion and discussion

In this paper, we present the largest dataset for fruit bin picking, comprising over 40 million 6D pose annotations and 1 million images. The dataset constitutes significant challenges for 6D pose estimation, as demonstrated in Section 5, including occlusion and texture-less objects. To address different aspects of 6D pose estimation, we have created eight distinct datasets to evaluate scene generalization, camera point of view generalization, and occlusion robustness. While the current baseline model exhibits its own strengths, none of the baselines achieve satisfactory performance across all benchmarks, offering the research community a challenging benchmark to tackle.

This dataset is curated specifically for 6D pose estimation, offering extensive annotations for enhanced accessibility. Beyond its possible applications in 3D reconstruction, Nerf reconstruction, and multi-view 6D pose estimation, it also holds significant value for robotics learning, highlighting one of its key advantages. With its versatility, the dataset bridges the gap between computer vision and robotics, fostering collaboration and innovation across various research areas. Our intention is for this dataset to facilitate the improvement of existing 6D pose estimation models and drive further advancements in the field of 6D pose for robotics.

Acknowledgments and Disclosure of Funding

This work was in part supported by the French Research Agency, l'Agence Nationale de Recherche (ANR), through the projects Learn Real (ANR-18-CHR3-0002-01), Chiron (ANR-20-IADJ-0001-01), Aristotle (ANR-21-FAI1-0009-01). It was granted access to the HPC resources of IDRIS under the allocation 2022-[AD011012172R1] made by GENCI.

References

[1] Eric Brachmann. '6d object pose estimation using 3d object coordinates, 2014. `https://heidata.uni-heidelberg.de/dataset.xhtml?persistentId=doi:10.11588/data/V4MUMX`.

[2] Xiaoyu Chen, Jiachen Hu, Chi Jin, Lihong Li, and Liwei Wang. Understanding domain randomization for sim-to-real transfer. In *International Conference on Learning Representations*, 2022.

[3] Yuhao Chen, E. Zhixuan Zeng, Maximilian Gilles, and Alexander Wong. MetaGraspNet_v0: A Large-Scale Benchmark Dataset for Vision-driven Robotic Grasping via Physics-based Metaverse Synthesis. *ArXiv*, pages 1–7, dec 2021.

[4] D.M. Coleman, Ioan Alexandru Sucan, Sachin Chitta, and Nikolaus Correll. Reducing the barrier to entry of complex robotic software: a moveit! case study. *ArXiv*, abs/1404.3785, 2014.

[5] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. *CoRR*, abs/1910.11215, 2019.

[6] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. BlenderProc. *arXiv*, oct 2019.

[7] Guillaume Duret, Nicolas Cazin, Mahmoud Ali, Florence Zara, Emmanuel Dellandréa, Jan Peters, and Liming Chen. PickSim: A dynamically configurable Gazebo pipeline for robotic manipulation. In *Advancing Robot Manipulation Through Open-Source Ecosystems - 2023 IEEE International Conference on Robotics and Automation (ICRA) Conference Workshop*, Londres, United Kingdom, May 2023. Adam Norton, University of Massachusetts Lowell and Holly Yanco, University of Massachusetts Lowell and Berk Calli, Worcester Polytechnic Institute and Aaron Dollar, Yale University.

[8] Maximilian Gilles, Yuhao Chen, Tim Robin Winter, E. Zhixuan Zeng, and Alexander Wong. MetaGraspNet: A Large-Scale Benchmark Dataset for Scene-Aware Ambidextrous Bin Picking via Physics-based Metaverse Synthesis. In *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, volume 2022-Augus, pages 220–227. IEEE, aug 2022.

[9] Anas Gouda, Abraham Ghanem, and Christopher Reining. DoPose-6D dataset for object segmentation and 6D pose estimation. *ArXiv*, apr 2022.

[10] Matthieu Grard, Emmanuel Dellandréa, and Liming Chen. Deep multicameral decoding for localizing unoccluded object instances from a single rgb image. *International Journal of Computer Vision*, 128, 05 2020.

[11] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh Ti Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S.M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: A scalable dataset generator. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2022-June, pages 3739–3751. IEEE, jun 2022.

[12] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Computer Vision–ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part I 11*, pages 548–562. Springer, 2013.

[13] Tomas Hodan, Pavel Haluza, Stepan Obdrzalek, Jiri Matas, Manolis Lourakis, and Xenophon 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)*, pages 880–888. IEEE, mar 2017.

[14] Tomas Hodan, Vibhav Vineet, Ran Gal, Emanuel Shalev, Jon Hanzelka, Treb Connell, Pedro Urbina, Sudipta N. Sinha, and Brian Guenter. Photorealistic Image Synthesis for Object Instance Detection. In *2019 IEEE International Conference on Image Processing (ICIP)*, volume 2019-Septe, pages 66–70. IEEE, sep 2019.

[15] Roman Kaskman, Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. HomebrewedDB : RGB-D Dataset for 6D Pose Estimation of 3D Objects Technical University of Munich , Germany Siemens Corporate Technology , Germany. *ICCV Workshop*, 2019.

[16] Kilian Kleeberger, Richard Bormann, Werner Kraus, and Marco F Huber. A Survey on Learning-Based Robotic Grasping. *Current Robotics Reports*, 1(4):239–249, 2020.

[17] Kilian Kleeberger, Christian Landgraf, and Marco F. Huber. Large-scale 6D Object Pose Estimation Dataset for Industrial Bin-Picking. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2573–2578. IEEE, nov 2019.

[18] Nathan Koenig and Andrew Howard. Design and use paradigms for Gazebo, an open-source multi-robot simulator. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3:2149–2154, 2004.

[19] Xingyu Liu, Shun Iwase, and Kris M. Kitani. StereOBJ-1M: Large-scale Stereo Image Dataset for 6D Object Pose Estimation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10850–10859. IEEE, oct 2021.

[20] Samarth Mishra, Rameswar Panda, Cheng Perng Phoo, Chun-Fu (Richard) Chen, Leonid Karlinsky, Kate Saenko, Venkatesh Saligrama, and Rogerio S. Feris. Task2sim: Towards effective pre-training and transfer from synthetic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9194–9204, June 2022.

[21] Fabio Muratore, Fabio Ramos, Greg Turk, Wenhao Yu, Michael Gienger, and Jan Peters. Robot learning from randomized simulations: A review. *Frontiers in Robotics and AI*, 9, 2021.

[22] Sida Peng, Yuan Liu, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Pvnet: Pixel-wise voting network for 6dof pose estimation, 2018.

[23] Arul Selvam Periyasamy, Max Schwarz, and Sven Behnke. SynPick: A Dataset for Dynamic Bin Picking Scene Understanding. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, volume 2021-Augus, pages 488–493. IEEE, aug 2021.

[24] Jonathan Tremblay, Thang To, and Stan Birchfield. Falling Things: A Synthetic Dataset for 3D Object Detection and Pose Estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, volume 2018-June, pages 2119–21193. IEEE, jun 2018.

[25] Stephen Tyree, Jonathan Tremblay, Thang To, Jia Cheng, Terry Mosier, Jeffrey Smith, and Stan Birchfield. 6-DoF Pose Estimation of Household Objects for Robotic Manipulation: An Accessible Dataset and

Benchmark. *IEEE International Conference on Intelligent Robots and Systems*, 2022-Octob:13081–13088, 2022.

[26] Rutgers University. Rutgers apc rgb-d dataset, 2016. `https://robotics.cs.rutgers.edu/pracsys/rutgers-apc-rgb-d-dataset/`.

[27] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3338–3347, 2019.

[28] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[29] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes. In *Robotics: Science and Systems XIV*. Robotics: Science and Systems Foundation, jun 2018.

[30] Honglin Yuan, Tim Hoogenkamp, and Remco C. Veltkamp. RobotP: A benchmark dataset for 6D object pose estimation. *Sensors (Switzerland)*, 21(4):1–26, 2021.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplemental material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See section 6
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Acknowledgement and supplemental material.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We used the code of PickSim
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Example of Images from the dataset

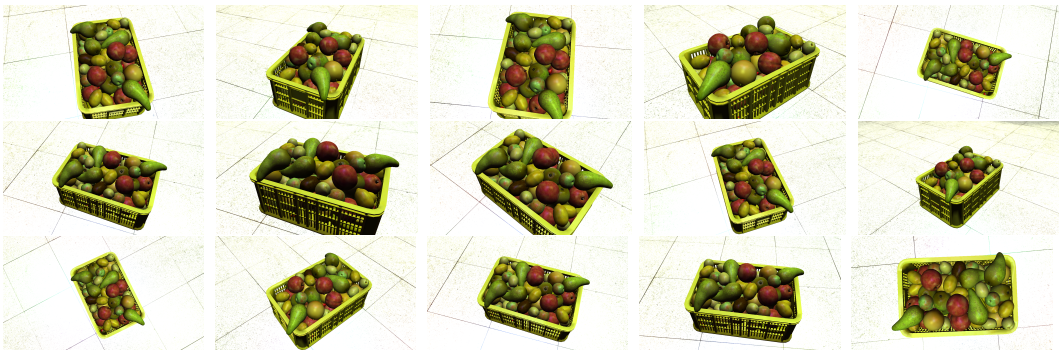


Figure 6: 15 point of view of a single scene of FruitBin

B Appendix

Include extra information in the appendix. This section will often be part of the supplemental material. Please see the call on the NeurIPS website for links to additional guides on dataset publication.

1. Submission introducing new datasets must include the following in the supplementary materials:
   (a) Dataset documentation and intended uses. Recommended documentation frameworks include datasheets for datasets, dataset nutrition labels, data statements for NLP, and accountability frameworks.
   (b) URL to website/platform where the dataset/benchmark can be viewed and downloaded by the reviewers.
   (c) Author statement that they bear all responsibility in case of violation of rights, etc., and confirmation of the data license.
   (d) Hosting, licensing, and maintenance plan. The choice of hosting platform is yours, as long as you ensure access to the data (possibly through a curated interface) and will provide the necessary maintenance.

Figure 7: 31 first scenes for a unique point of view

2. To ensure accessibility, the supplementary materials for datasets must include the following:
   (a) Links to access the dataset and its metadata. This can be hidden upon submission if the dataset is not yet publicly available but must be added in the camera-ready version. In select cases, e.g when the data can only be released at a later date, this can be added afterward. Simulation environments should link to (open source) code repositories.
   (b) The dataset itself should ideally use an open and widely used data format. Provide a detailed explanation on how the dataset can be read. For simulation environments, use existing frameworks or explain how they can be used.
   (c) Long-term preservation: It must be clear that the dataset will be available for a long time, either by uploading to a data repository or by explaining how the authors themselves will ensure this.
   (d) Explicit license: Authors must choose a license, ideally a CC license for datasets, or an open source license for code (e.g. RL environments).
   (e) Add structured metadata to a dataset's meta-data page using Web standards (like schema.org and DCAT): This allows it to be discovered and organized by anyone. If you use an existing data repository, this is often done automatically.
   (f) Highly recommended: a persistent dereferenceable identifier (e.g. a DOI minted by a data repository or a prefix on identifiers.org) for datasets, or a code repository (e.g. GitHub, GitLab,...) for code. If this is not possible or useful, please explain why.
3. For benchmarks, the supplementary materials must ensure that all results are easily reproducible. Where possible, use a reproducibility framework such as the ML reproducibility checklist, or otherwise guarantee that all results can be easily reproduced, i.e. all necessary datasets, code, and evaluation procedures must be accessible and documented.

Figure 8: 31 first scenes for a unique point of view

4. For papers introducing best practices in creating or curating datasets and benchmarks, the above supplementary materials are not required.