Falling Things: A Synthetic Dataset for
3D Object Detection and Pose Estimation

Jonathan Tremblay
NVIDIA
jtremblay@nvidia.com

Thang To
NVIDIA
thangt@nvidia.com

Stan Birchfield
NVIDIA
sbirchfield@nvidia.com

Abstract

We present a new dataset, called Falling Things (FAT), for advancing the state-of-the-art in object detection and 3D pose estimation in the context of robotics.[1] By synthetically combining object models and backgrounds of complex composition and high graphical quality, we are able to generate photorealistic images with accurate 3D pose annotations for all objects in all images. Our dataset contains 60k annotated photos of 21 household objects taken from the YCB dataset [2]. For each image, we provide the 3D poses, per-pixel class segmentation, and 2D/3D bounding box coordinates for all objects. To facilitate testing different input modalities, we provide mono and stereo RGB images, along with registered dense depth images. We describe in detail the generation process and statistical analysis of the data.

1. Introduction

Robotic manipulation of household objects in everyday environments requires accurate detection and pose estimation of multiple object categories. This presents a two-fold challenge for developing robotic perception algorithms. First, acquiring ground truth data is time-consuming, error-prone, and potentially expensive (depending upon the technique used). This limits the *evaluation* of algorithms, particularly with respect to new object categories or environmental conditions. Secondly, existing techniques for acquiring real world data [15, 5, 14] do not scale. As a result, they are not capable of generating the large datasets that are needed for training deep neural networks.

We propose to overcome such limitations by using synthetically generated data. Synthetic data has been gaining traction in recent years as an efficient means of both training and evaluating DNNs for computer vision problems for which collecting ground truth data is laborious, *e.g.*, stereo [16], optical/scene flow [10], or semantic segmentation [4].

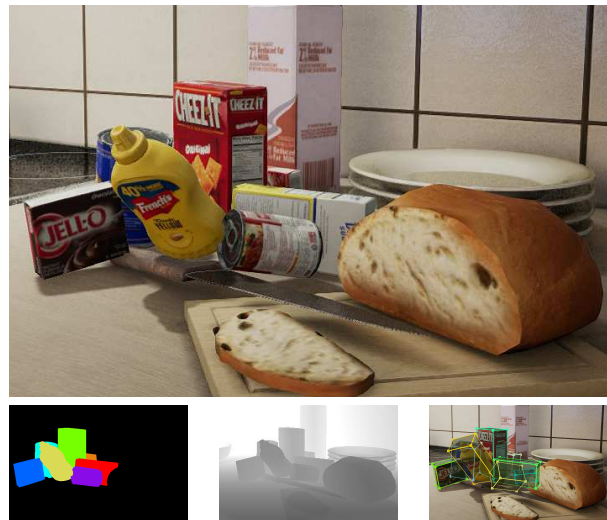[1]The dataset can be downloaded from http://research.nvidia.com/publication/2018-06_Falling-Things

Figure 1. The Falling Things (FAT) dataset was generated by placing 3D household object models (*e.g.*, mustard bottle, soup can, gelatin box, etc.) in virtual environments. Each snapshot consists of a stereo pair of RGB images (only one of which is shown, top), pixelwise segmentation of the objects (bottom left), depth (bottom center), 2D/3D bounding box coordinates (bottom right), and 3D poses of all objects (not shown).

We believe that 3D detection and pose estimation fall within this category and are thus a natural fit for synthetic data due to the difficulty of acquiring accurate ground truth.

In this paper we introduce the *Falling Things (FAT)* dataset, consisting of a large number (61,500) of snapshots for training and evaluating robotics scene understanding algorithms in household environments. Specifically, as shown in Fig. 1, each snapshot consists of a stereo pair of image frames with corresponding depth images, along with the 3D poses, 2D/3D bounding boxes, projected 3D bounding boxes, and pixelwise segmentation of the known objects in the scene. Unlike previous datasets, we aim for photorealistic images of real-world objects, leveraging the popular YCB object models [2] placed in different virtual environments.

To better understand the contribution of our dataset, con-

sult Table 1. To our knowledge, only two datasets exist with accurate ground truth poses of multiple objects with significant occlusion (T-LESS [6], YCB-Video [15]). Yet neither of these datasets contain extreme lighting variations or multiple modalities. Our FAT dataset thus extends upon these existing solutions in both quantity and variety.

2. Falling Things Dataset

Description. All data were generated by a custom plugin we developed for Unreal Engine 4 (UE4). By leveraging asynchronous, multithreaded sequential frame grabbing, the plugin generates data at a rate of 50–100 Hz, which is significantly faster than either the default UE4 screenshot function or the publicly available UnrealCV tool [12].

We selected three virtual environments within UE4: a kitchen, sun temple, and forest. These environments were chosen for their high-fidelity modeling and quality, as well as for the variety of indoor and outdoor scenes. For each environment we manually selected five specific locations covering a variety of terrain and lighting conditions (*e.g.*, on a kitchen counter or tile floor, next to a rock, above a grassy field, and so forth). Together, these yielded 15 different locations consisting of a variety of 3D backgrounds, lighting conditions, and shadows.

We selected 21 household objects[2] from the YCB [2] dataset.[3] Since these models are not all aligned, translation was applied to each of the downloaded models to center the coordinate frame at the object centroid, and rotation was applied to align the coordinate axes with those of the object, taking product labeling into account. For each run, some of these object models were placed at random positions and orientations within a vertical cylinder of radius 5 cm and height of 10 cm placed at a fixation point. The objects' initial positions are sampled within this volume to avoid initial penetration. The objects were then allowed to fall under the force of gravity, as well as to collide with one another and with the surfaces in the scene. While the objects fell, the virtual camera system was rapidly teleported to random azimuths, elevations, and distances with respect to the fixation point to collect data. Azimuth ranged from $-120°$ to $+120°$ (to avoid collision with the wall, when present), elevation from $5°$ to $85°$, and distance from 0.5 m to 1.5 m.

Our virtual camera system consists of a pair of stereo RGBD cameras. This design decision allows the dataset to support at least three different sensor modalities. Whereas single RGBD sensors are commonly used within robotics, stereo sensors have the potential to yield higher quality output with fewer distortions, and a monocular RGB camera has obvious advantages in terms of cost, simplicity, and availability. By supporting all of these options, the dataset

allows researchers to use their modality of choice, as well as to explore the important topic of comparing across modalities. In our system, the baseline separating the left and right cameras is 6 cm; and each camera's horizontal field of view is 64 degrees, leading to a focal length of 768.2 pixels with an image resolution of 960×540.

The dataset (consisting of 61,500 unique frames) is divided into two parts:

- *Single objects.* The first part of the dataset was generated by dropping each object model in isolation $\sim$5 times at each of the 15 locations. For each run, $\sim$20 image frames were generated by taking screenshots at a rate of 10 Hz while the object fell for $\sim$2 s. Therefore, this part of the dataset consists of 1500 (100×15) unique images for each of the 21 objects, thus totaling 31,500 frames.
- *Mixed objects.* The second part of the dataset was generated in the same manner except with a random number of objects sampled uniformly from 2 to 10. By sampling the objects with replacement, we allow multiple instances of the same category in an image, unlike many previous datasets. For each location we generated 2,000 images, thus yielding 30,000 frames.

Testing. There are several natural ways to split the FAT dataset for training and testing. One approach would be to hold out one location per scene as the test sets, and leave the other data for training. Another approach would be to hold out one environment for testing, leaving the others for training. Finally, the single object images could be used for training, while using the remaining images for testing—that is, assuming that occlusion is artificially introduced in the data augmentation process during training. Further details regarding training/testing methodology can be found with the dataset.

Statistics. Fig. 2 displays the total number of occurrences of each object class in the FAT dataset, using opacity of the color bars to indicate the percentage of the object that is visible (i.e., non-occluded). That is, solid color bars show the number of occurrences for which the object is at least 75% visible, whereas lighter color bars indicate the occurrences for which the visibility is between 25% and 75%. (Occurrences with visibility less than 25% are not shown.) As can be seen, smaller objects (such as the scissors or marker) are occluded more often than larger objects (such as the cracker box).

Fig. 3 shows statistical distributions of various parameters of one of the YCB objects (the mustard bottle) in the FAT dataset (from left camera perspective). Shown are the yaw, pitch, and roll of the object with respect to the front. The modes in yaw (at 0° and 180° and, to a lesser extent, at $\pm$90°) are due to the fact that after falling, the object likely lies on either its front or back side. Similarly, the angle of the camera with respect to the resting surface biases the

---

[2]We used the same subset of 21 objects as in [15].

[3]http://www.ycbbenchmarks.com/object-models

Dataset	# objects	# frames	description	depth	stereo	3D pose	full rotation	occlusion	extreme lighting	segmentation	bbox coords
EPFL multi-view [11]	20	2k	cars	✗	✗	✓−	✗	✗	✗	✗	✗
UW RGBD [9]	300	250k	household	✓	✗	✓−	✓	✗	✗	✗	✗
LINEMOD [5]	15	18k	household	✓	✗	✓	✓	✗	✗	✗	✗
Object tracking [3]	4	6k	household	✓	✗	✓	✓	✗	✗	✗	✗
Pascal3D+ [14]	12	30k	various	✗	✗	✓	✗	✓	✓	✗	✗
Brachmann et al. [1]	20	10k	various	✓	✗	✓	✓	✓	✓	✗	✗
Occlusion [1, 7]	8	1k	household	✓	✗	✓	✓	✓	✗	✗	✗
Krull et al. [8]	3	3k	handheld	✓	✗	✓	✓	✓	✗	✗	✗
Rutgers APC [13]	24	10k	warehouse	✓	✗	✓	✗	✗	✗	✗	✗
T-LESS [6]	30	10k	industrial	✓	✗	✓+	✓	✓	✗	✗	✗
YCB-Video [15]	21	134k	household	✓	✗	✓+	✓	✓	✗	✓	✓
FAT (ours)	21	60k	household	✓	✓	✓+	✓	✓	✓	✓	✓

Table 1. Datasets for object detection and pose estimation. Under 3D pose, ✓+ means that the pose of all known objects in the scene are provided, ✓ means only the pose of a single object is provided, and ✓− means that the provided poses are approximate. Note that segmentation and bounding boxes can be determined for any dataset by overlaying models according to ground truth.
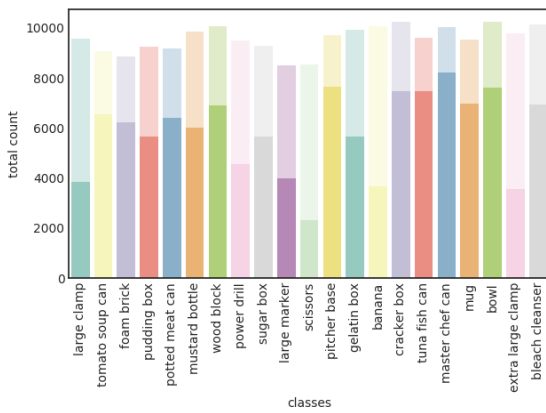

Figure 2. Total appearance count of the 21 YCB objects in the FAT dataset. Light color bars indicate object visibility greater than 25%, while solid bars indicate visibility greater than 75%. (100% visible means *not occluded*, 0% visible means *fully occluded*.)

pitch angle toward neutrality. In contrast, the roll is uniform.

Also shown in the figure is the distribution of the distance to the camera, which extends slightly beyond the intended range of 0.5 m to 1.5 m because oftentimes objects roll or slide after impact. The final row shows the visibility indicating that, while the single objects are fully visible, significant occlusions occur; and the distribution of the centroid within the image, which is approximately a broad Gaussian centered near the center of the image. Similar distributions were observed for other YCB objects.

3. Conclusion

We have presented a new dataset to accelerate research in object detection and pose estimation, as well as segmentation, depth estimation, and sensor modalities. The proposed dataset focuses on household items from the YCB dataset
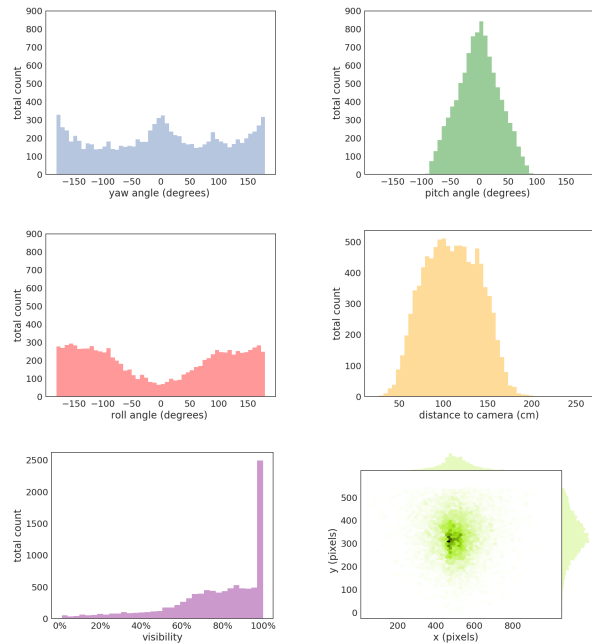

Figure 3. Statistics for one object (mustard bottle) in the FAT dataset. In lexicographic order are the distribution of yaw, pitch, and roll angles; distance to the camera; visibility; and position of the object's centroid within the exported images.

and has been rendered with high fidelity and a wide variety of backgrounds, poses, occlusions, and lighting conditions. Statistics from the dataset confirm this variety quantitatively. We hope that researchers find this dataset useful for exploring robust solutions to open problems such as object detection, pose estimation, depth estimation from monocular and/or stereo cameras, and depth-based segmentation, to advance the field of robotic manipulation.

References

[1] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother. Learning 6D object pose estimation using 3D object coordinates. In *ECCV*, 2014. 3

[2] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar. The YCB object and model set: Towards common benchmarks for manipulation research. In *ICAR*, 2015. 1, 2

[3] C. Choi and H. I. Christensen. RGB-D object tracking: A particle filter approach on GPU. In *IROS*, 2013. 3

[4] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016. 1

[5] 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 *ACCV*, 2012. 1, 3

[6] T. Hodaň, 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 *WACV*, 2017. 2, 3

[7] A. Krull, E. Brachmann, F. Michel, M. Y. Yang, S. Gumhold, and C. Rother. Learning analysis-by-synthesis for 6D pose estimation in RGB-D images. In *ICCV*, 2015. 3

[8] A. Krull, F. Michel, E. Brachmann, S. Gumhold, S. Ihrke, and C. Rother. 6-DOF model based tracking via object coordinate regression. In *ACCV*, 2014. 3

[9] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view RGB-D object dataset. In *ICRA*, 2011. 3

[10] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *arXiv 1512.02134*, 2015. 1

[11] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *CVPR*, 2009. 3

[12] W. Qiu and A. Yuille. UnrealCV: Connecting computer vision to Unreal Engine. In *arXiv 1609.01326*, 2016. 2

[13] C. Rennie, R. Shome, K. E. Bekris, and A. F. D. Souza. A dataset for improved RGBD-based object detection and pose estimation for warehouse pick-and-place. *IEEE Robotics and Automation Letters (RA-L)*, 1(2):1179–1185, July 2016. 3

[14] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond PASCAL: A benchmark for 3D object detection in the wild. In *WACV*, 2014. 1, 3

[15] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In *arXiv 1711.00199*, 2017. 1, 2, 3

[16] Y. Zhang, W. Qiu, Q. Chen, X. Hu, and A. Yuille. Unreal-Stereo: A synthetic dataset for analyzing stereo vision. In *arXiv:1612.04647*, 2016. 1