Physics-aware Simulation for Object Detection and Pose Estimation

Chaitanya Mitash, Kostas E. Bekris and Abdeslam Boularias
Department of Computer Science,
Rutgers University
{ cm1074,kb572,ab1544}@cs.rutgers.edu

Abstract— **This work proposes a fully autonomous process to train Convolutional Neural Networks (`CNNs`) for object detection and pose estimation in setups for robotic manipulation. The application involves detection of objects placed in a clutter and in tight environments, such as a shelf. In particular, given access to 3D object models, several aspects of the environment are simulated and the models are placed in physically realistic poses with respect to their environment to generate a labeled synthetic dataset. To further improve object detection, the network self-trains over real images that are labeled using a multi-view pose estimation process. Results show that the proposed process outperforms popular training processes relying on synthetic data generation and manual annotation.**

I. INTRODUCTION

Object detection and pose estimation are frequently the initial step of any robotic manipulation task. The state of the art techniques for solving such visual recognition problems are based on supervised training of Convolutional Neural Networks (`CNNs`). Desirable results are typically obtained by training `CNNs` using datasets that involve a very large number of labeled images (e.g., ImageNet [1], and `MS-COCO` [2]). Creating such large datasets requires intensive human labor. Furthermore, as these datasets are general-purpose, one needs to create new datasets for specific object categories and environmental setups.

The recent Amazon Picking Challenge (`APC`) [3] has reinforced this realization and has led into the development of datasets specifically for the detection of objects inside shelving units. These datasets are created either with human annotation [4], [5] or by constraining scenes to single objects and performing background subtraction [6]. An increasingly popular approach to avoid manual labeling is to use synthetic datasets generated by rendering 3D CAD models of objects with different viewpoints. Synthetic datasets have been used to train `CNNs` for object detection [7] and viewpoint estimation [8]. One major challenge in using synthetic data is the inherent difference between virtual training examples and real testing data. There is a considerable interest in studying the impact of texture, lighting, and shape to address this disparity [9]. One issue with synthetic images generated from rendering engines is that they display objects in poses that are not physically realistic. Moreover, occlusions are usually treated in a rather naive manner, i.e., by applying cropping, or pasting rectangular patches, which again results in unrealistic scenes [7] [8] [10].

This work proposes an automated system for generating and labeling datasets for training `CNNs`. In particular, the two main contributions of this work are:

- a simulator that uses the information from camera calibration, shelf or table localization to setup an environment, performs physics simulation to place objects at realistic configurations and renders images of scenes to generate a synthetic dataset to train an object detector,
- and a lifelong self-learning system that uses the object detector trained with our simulator to perform a robust multi-view pose estimation with a robotic manipulator, and use the results to correctly label real images in all the different views. The key insight behind this system is the fact that the robot can often find a good view that allows the detector to accurately label the object and estimate its pose. The object's predicted label is then used to label images of the same scene taken from more difficult views.

Please refer to [12] for an extended version of this work. For transparency, the software and data for the proposed system, are publicly available at `http://www.cs.rutgers.edu/~cm1074/PHYSIM.html`

II. TECHNICAL DETAILS

The problem statement that we consider is: given a discrete set of sensing configurations of the manipulator and a list of known objects that might appear in the scenes, our objective is to generate a labeled dataset that mimics the data from sensor. Quality of the dataset will be evaluated by using it to train a `CNN` based object detector and testing it's performance on data received from the sensor itself. We approach the problem in two broad steps of physics-aware simulation and real-world adaptation.

The first component is a physics-aware simulator that generates realistic synthetic data. The pipeline for the process is depicted in 1. This module has been implemented using the Blender Python API which internally uses Bullet for physics simulation. We start with creating texture mapped 3D CAD models of the known objects and the resting surface such as a shelf or a table in Blender. A RANSAC[13] based approach is used to calibrate the resting surface. Once the resting surface is localized, object

The authors are with the Computer Science Department of Rutgers University in Piscataway, New Jersey, 08854, USA. Email: {cm1074,kb572,ab1544}@rutgers.edu

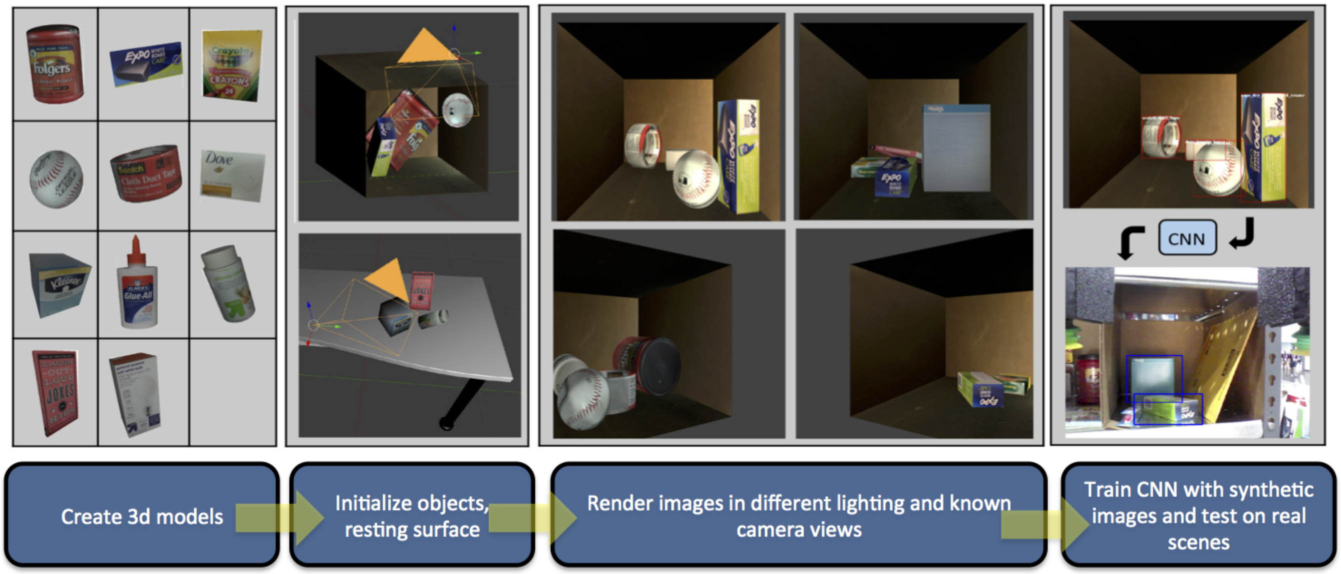
CNN
Create 3d models
Initialize objects, resting surface
Render images in different lighting and known camera views
Train CNN with synthetic images and test on real scenes

Fig. 1: Pipeline for physics aware simulation. The 3D CAD models are generated and loaded in a calibrated environment on the simulator. A subset of the objects is chosen for generating a scene. Objects undergo physics simulation to settle down on the resting surface under the effect of gravity. The scenes are rendered from known camera poses and perspective projection is used to compute 2D bounding boxes for each object. The labeled scenes are used to train Faster-RCNN [11] object detector, which is tested on real world setup.

selection and and initial poses for each scene are chosen uniformly at random within a domain defined by the geometry of the resting surface. Once initialized, the objects fall due to gravity, bounce, and collide with each other and with the resting surfaces. Any inter-penetrations among objects are appropriately treated by the physics engine. The final poses of the objects, when they stabilize, resemble real-world poses. The simulated scene is then rendered from multiple views using the camera poses computed from the known sensing configurations of the robot. The illumination of the scene is approximated by using point light sources which are varied with respect to location, intensity, and color for each rendering. Finally, perspective projection is applied to obtain 2D bounding box labels for each object in the scene. The overlapping portion of the bounding boxes for the object that is further away from the camera is pruned. The synthetic dataset generated from the above process is used to train Faster R-CNN [11] based object detector with deep VGG network architecture [14].

Given access to an object detector trained with the physics-aware simulator, the self-learning pipeline as depicted in figure 2 precisely labels real world images using a robust multi-view pose estimation. This is based on the idea that the detector performs well on some views, while might be imprecise or fail in other views, but aggregating 3d data over the confident detections and with access to the knowledge of the environment, a 3d segment can be extracted for each object instance in the scene. This combined with the fact that we have 3d models of objects, makes it highly likely to estimate correct 6D pose of objects given enough views and search time. We use Super4PCS [15] to perform model matching. The confident success in pose estimation is then projected back to the multiple views, and used to label real images. These examples are very effective to reduce the confusion in the classifier for novel views. The process also autonomously reconfigures the scene using manipulation actions to apply the labeling process iteratively over time on different scenes, thus generating a labeled dataset which is then used to re-train the object detector. The PRACSYS motion planning library is used for performing the manipulation actions.

III. EVALUATION

We evaluate our system on the benchmark dataset released by Team MIT-Princeton [6] in the APC 2016 framework. The experiments are performed on 148 scenes in the shelf environment with different levels of lighting and clutter. The scenes include 11 objects used in APC with 2220 images and 229 unique object poses. The objects were chosen to represent different geometric shapes, however ignoring the ones which did not have any depth information. The standard Intersection-Over-Union (IoU) metric is used to evaluate the performance in object detection task. For 6D pose estimation success is evaluated as the percentage of predictions with an error in translation less than 5cm and mean error in the rotation less than 15^o. Evaluations for the object detection task can be found in Table 3. We compare our results to the benchmark performance [6], where the training images are real images of single objects labeled by background subtraction. We further demonstrate the importance of placing objects at physically realistic

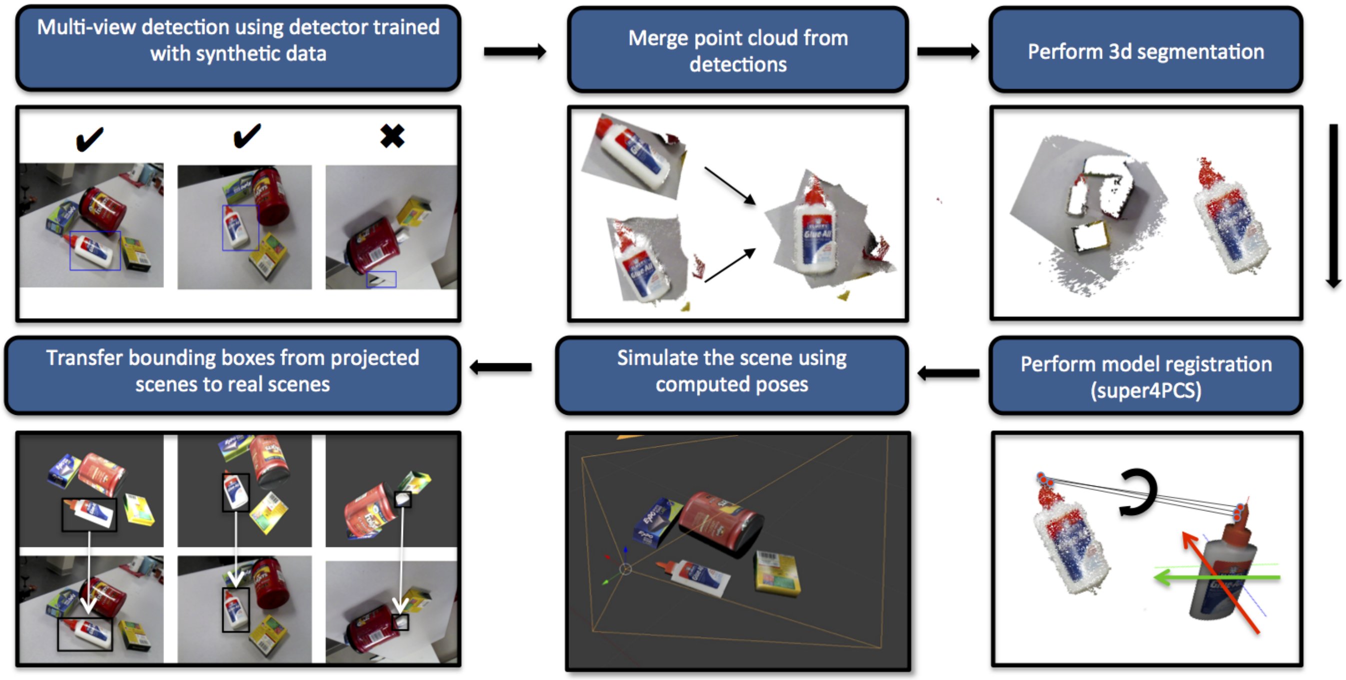
Multi-view detection using detector trained with synthetic data
Merge point cloud from detections
Perform 3d segmentation
Transfer bounding boxes from projected scenes to real scenes
Simulate the scene using computed poses
Perform model registration (super4PCS)

Fig. 2: Self-learning pipeline. Detector trained with simulated data is used to detect objects from multiple views. The point cloud aggregated from successful detections undergoes 3d segmentation. Super4PCS [15] is used to estimate 6D pose of the object in world frame. The computed poses with high confidence values are simulated and projected back to the multiple camera views to obtain precise labels over real images.

Training dataset	**succ**
Benchmark (MIT-Princeton) [6]	75%
synthetic data with known pose distribution	69%
synthetic data with uniform pose distribution	31%
physics simulation (Our)	64%
physics simulation, varying illumination (Our)	70%
adding data with multi-view self labeling (Our)	**82%**

Fig. 3: Object detection results on Princeton Shelf&Tote dataset

poses in the simulation and the utility of randomization with respect to the unknown parameters such as illumination.

The utility of our training in localizing highly occluded objects from multiple views, is reflected in the performance on the 6D pose estimation task 4. We compare our system to that of the MIT-Princeton team for APC 2016, where the system uses a semantic segmentation framework [16] trained with a dataset of real images. It is interesting to note that our success in pose estimation task is at par with the success achieved when using ground-truth bounding boxes. This identifies the need for an efficient global reasoning for pose estimation which is generally ignored because their computation complexity.

Object recognition/model matching	**pose succ(%)**
FCN/PCA, ICP [6]	54.6%
ground-truth bounding-box/PCA, ICP	84.8%
RCNN/Super4PCS (Our-training)	75.0%
RCNN/PCA, ICP (Our-training)	**79.4%**

Fig. 4: Pose Estimation results using different object recognition and model matching techniques.

IV. FUTURE WORK

In this work we presented a system to autonomously generate data to train CNNs for object detection and pose estimation in robotics. Even though the physics simulation contributes significantly in the training process, there exists a dataset bias in simulated data with respect to texture and illumination which we tackled by randomization and adding self-labeled real examples. In the future, we would like study how could learning the unknown parameters of the simulation such as illumination and model properties help improve the training, and secondly, how to efficiently use such a simulation for global reasoning in the pose estimation problem.

REFERENCES

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

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

[3] "Official website of Amazon Picking Challenge," 2016.

[4] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel, "Bigbird: A large-scale 3d database of object instances," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 509–516.

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

[6] A. Z. et al., "Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge," *arXiv preprint arXiv:1609.09475*, 2016.

[7] X. Peng, B. Sun, K. Ali, and K. Saenko, "Learning deep object detectors from 3d models," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1278–1286.

[8] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2686–2694.

[9] B. Sun and K. Saenko, "From virtual to reality: Fast adaptation of virtual object detectors to real domains," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[10] Y. Movshovitz-Attias, T. Kanade, and Y. Sheikh, "How useful is photo-realistic rendering for visual learning?" in *Computer Vision–ECCV 2016 Workshops*. Springer, 2016, pp. 202–217.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[12] C. Mitash, K. Bekris, and A. Boularias, "A self-supervised learning system for object detection using physics simulation and multi-view pose estimation," *arXiv:1703.03347*, 2017.

[13] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[15] N. Mellado, D. Aiger, and N. K. Mitra, "Super 4pcs fast global point-cloud registration via smart indexing," *Computer Graphics Forum. Vol. 33. No. 5*, 2014.

[16] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation."