Learning Pose from Quadrotor Motion

Seoungjun (Elijah) Lee, Raymond Bjorkman, and Ranjani Narayanan

I. Abstract

This work aims to solve a 3D pose estimation problem by using images that are taken from the viewpoint of a quadrotor. The quadrotor motion is leveraged to capture different angles and positions of the object and the outputs of the network are shown to be agnostic to variations in the contrast, lighting, and other environmental conditions. We propose a network with an architecture developed through empirical analysis and evaluate the performance comparatively.

II. Introduction

The need for accurate pose estimation of objects is important for a variety of common problems in robotics, including the task of manipulation, exploration, and localization. Traditionally, solving this problem has required expensive camera setups or the inclusion of key points. However, recent research has leveraged the advances in deep neural networks to predict the pose of an object directly from a 2D image. Our project seeks to adapt the approach in [1] for the task of pose estimation of an object from a quadrotor's video feed. The contribution of this work is that we aim to estimate the pose of an object from the viewpoint of a quadrotor so the proposed approach is different from the conventional method where the dataset is limited to the viewpoint from the ground.

III. Related Work

Based on the work in [2], domain randomization achieves an error of 1.5cm on 3D pose estimation. Since many robotic tasks require high precision, this work attempts to improve the accuracy of pose estimation based on domain randomization. Much work has already been conducted in 6D pose estimation using data from the RGB-D sensors. However, in an attempt to cut the expense of high-precision sensors and to leverage the simplicity of 2D image data we deploy various tried and tested techniques to improve the accuracy for this purpose. Previous works have involved addition of fiducial markers [3]–[5] which may be undesirable due to external modification on the objects to be detected. Due to a lack of labeled datasets, supervised learning using Deep CNNs is hardly deployed in the real world [6], [7]. Edge detectors [8], [9], image templates [10] and predefined feature points [11] have been used for model based pose estimation of rigid objects from 2D images. These algorithms rely on specific features like texture of surfaces for point matching, calibration of camera, etc. which we wish to eliminate through our approach.

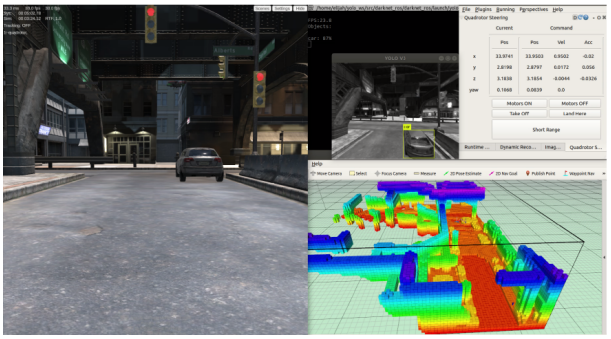

Fig. 1. An instance of data acquisition. **Left**: A quadrotor is flying in Unity simulator. **Top Right**: A raw image with an object (here, a car). The pose of the quadrotor is shown in the console. **Bottom Right**: Mapping for the environment.

IV. Dataset

There are a number of publicly available pose estimation datasets [12]–[14] but we did not find any that were suitable to our chosen application of pose estimation from a quadrotor. To emulate this task we build a custom dataset with images of a symmetrical, cuboidal object in the Unity simulator. We ran the simulation to collect RGB images and used a quadrotor-to-object transformation to obtain the ground truth object pose. The raw image with an object is then transformed into the desired 360 x 640 dimension to be input into the network. The ground truth for pose consists of 3 dimensions defining the object's position (x, y, z) relative to the camera along with its orientation θ on the ground plane. The dataset consists of 3002 images with zero object rotation and 5236 images where the object is rotated and has a non-zero θ (these datasets will henceforth be referred to as "Rot" and "NoRot" respectively). The orientation θ is capped to the range of $-180°$ to $180°$. For obtaining the quadrotor-to-object transformation, we assume that the state estimation of the robot is perfect and use odometry information as the quadrotor pose. An example of the data acquisition setup (to get the raw image data before applying the quadrotor-to-object transformation) is shown in Fig. 1.

V. Method

The entire dataset is randomly split 80% and 20% for training and testing respectively. The images are then normalized at the pixel level and data augmentation techniques such as rotation, translation, and color jitter are applied to boost performance of the network. Using sequential images obtained from the video feed we believe that the initial predictions will be refined as demonstrated in [1]. To check

Fig. 2. Examples of partially occluded images used to train the block detector. Both the orientations and positions are randomized.

Fig. 3. Sequential images of a cubical object in an urban setup captured from the quadrotor hovering at different heights (z) and angles. Yellow mark indicates the ground truth location.

the validity of this hypothesis, we also shuffle the dataset to disrupt the sequence of images and observe the performance of the network. Given our relatively lightweight approach, we expect that we will have low computational requirements such that training can be carried out on available hardware. Examples of generated data and ground truth are shown in Fig. 2 and Fig. 3.

A. Architecture

As shown in Fig. 4, the architecture for the model we built for this application begins with a backbone network. After this layer is a fully connected layer that yields 4 output values (corresponding to x, y, z and θ respectively). In order to select an appropriate backbone for this algorithm we empirically evaluated the performance of the model with several candidate networks. Squeezenet, Mobilenet, and Mnasnet were all considered for this task because of the relatively few parameters that each of them have.

B. Loss

We deploy L_2 loss for the x, y, and z coordinates in the estimate of pose and use a cosine function for penalizing the estimate of the orientation. A weighting factor λ is used for boosting the penalization due to orientation loss, since cosine values yield comparatively smaller values as opposed to the L_2 loss. The overall loss function is given below:

$$L = (x-\bar{x})^2 + (y-\bar{y})^2 + (z-\bar{z})^2 + \lambda(||\ cos(\theta) - cos(\bar{\theta})\ ||)$$

C. Training and Validation

When training the model on the dataset we opted to use a learning rate of 1e-4, a batch size of 4 images, and 20 epochs of training. A learning rate scheduler was deployed for the 5^{th} and 12^{th} epoch wherein the learning rate is reduced to its $1/10^{th}$. The training and validation loops are visualized below to evaluate the performance of the network with different backbones.

VI. Experiments

For the experiments, we test the performance of the pose estimation network considering (1) dataset with rotation vs. dataset with no rotation (2) domain randomization and (3) each of x, y, z dimension for cartesian loss.

First, as we prepare two datasets, we compare the result of the pose estimator, as visualized in Table I, Fig. 5 and Fig. 6. All the train and validation loss over 20 epochs are shown in the figure. The table shows the validation loss on entire test dataset for each backbone, and domain randomization is done with various contrast, hue, and lighting.

We observe that dataset with no rotation performs better than the dataset with rotation because orientation is subtle to measure and we have limited discretization for the angles. All the loss curves show that our network is learning. The domain randomization does not get much advantage because our object is specific to red object so changing color would worsen the performance and the shown result on the table is expected.

TABLE I

Validation Loss for each dataset

Backbone	Rotation	No Rotation	Domain Randomization
SqueezeNet	51.75	7.00	71.79
MobileNet	77.09	79.18	81.77
MnasNet	66.96	79.66	87.35

VII. Conclusion

We have successfully carried out pose estimation of a monochromatic, cuboidal object using images generated from quadrotor motion. All the images are generated using a sequential format and data augmentation in the form of contrast variations has been deployed. Additionally, the network shows robust performance when there are occluded images and works on both kind of images i.e. with and without rotation. Pose estimated using a Squeezenet backbone gives the most optimal results.

VIII. Supplementary Video Explanation

A verbal explanation of the work documented in this report can be found at the following link: https://drive.google.com/drive/folders/14nkM-iCqPHsuZH6SMp6g_L2FfbJGSGgb?usp=sharing

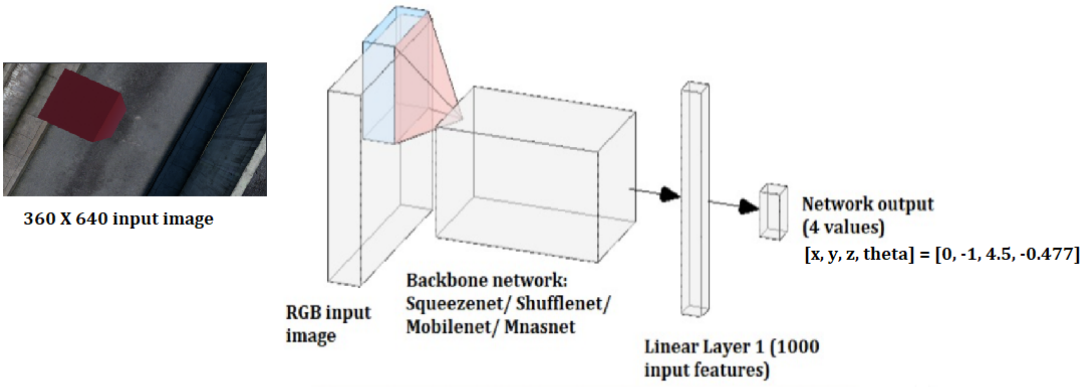

Fig. 4. Overall network for pose estimation

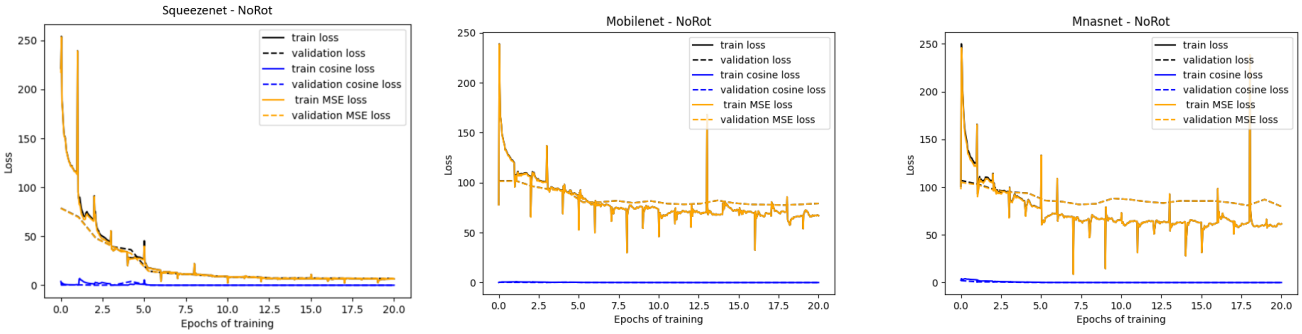

Fig. 5. Train and validation loss curves for all the tested backbones on the NoRot dataset

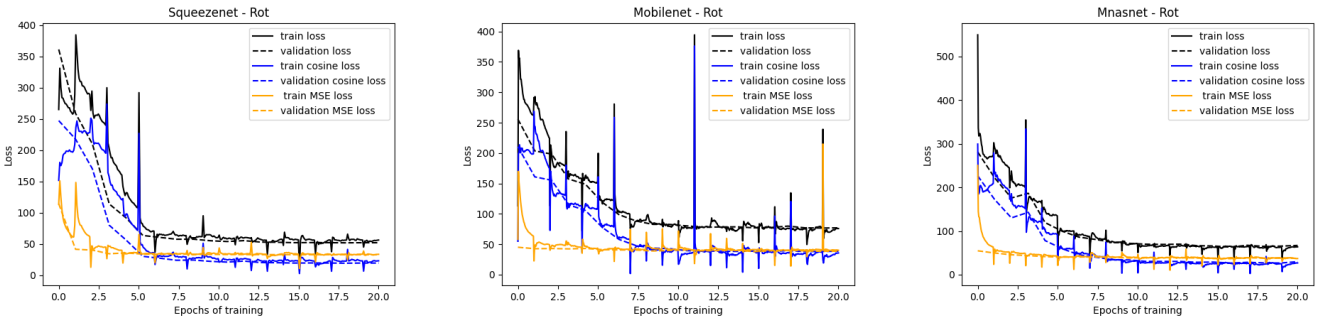

Fig. 6. Train and validation loss curves for all the tested backbones on the Rot dataset

REFERENCES

[1] Xinyi Ren, Jianlan Luo, Eugen Solowjow, Juan Aparicio Ojea, Abhishek Gupta, Aviv Tamar, and Pieter Abbeel. Domain randomization for active pose estimation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7228–7234. IEEE, 2019.

[2] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30. IEEE, 2017.

[3] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407. IEEE, 2011.

[4] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, pages 85–94. IEEE, 1999.

[5] Filippo Bergamasco, Andrea Albarelli, Emanuele Rodola, and Andrea Torsello. Rune-tag: A high accuracy fiducial marker with strong occlusion resilience. In *CVPR 2011*, pages 113–120. IEEE, 2011.

[6] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[7] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 292–301, 2018.

[8] Vincent Lepetit and Pascal Fua. *Monocular model-based 3D tracking of rigid objects*. Now Publishers Inc, 2005.

[9] Tom Drummond and Roberto Cipolla. Real-time visual tracking of complex structures. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):932–946, 2002.

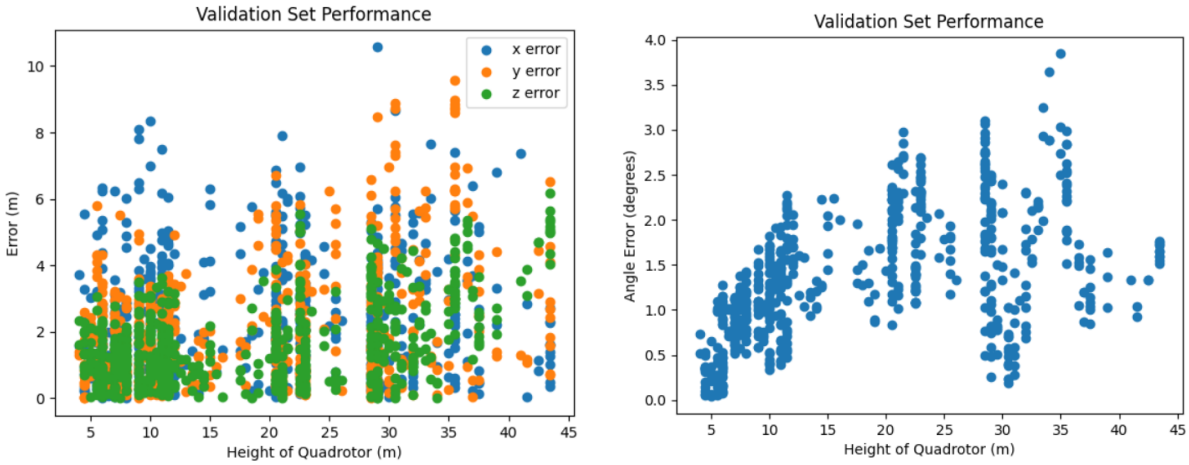

Fig. 7. Error on validation dataset for trained network with Squeezenet backbone. When the quadrotor has a higher elevation, the network tends to make less slightly less accurate predictions.

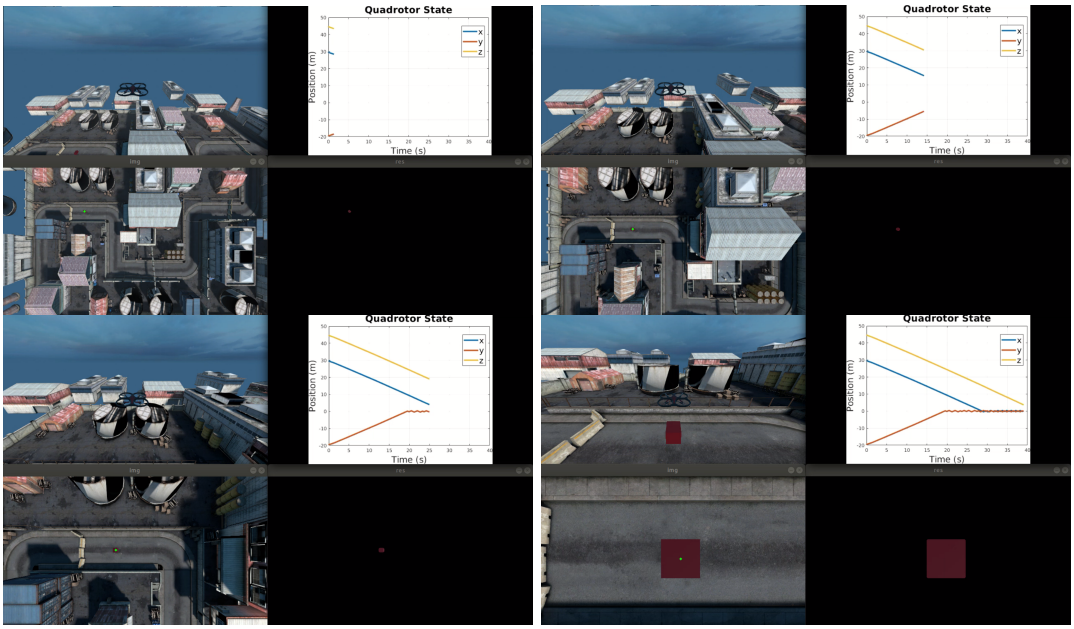

Fig. 8. Visual results for the perception-action loop can be seen in Figure 3. Each quadrant represents a snapshot of the simulation at one of $t = [0, 10, 20, 30]$. Within each snapshot, we show the global view of the simulation (top left), the view from the downward facing quadrotor camera (bottom left), the quadrotor state (top right), and the output of the object detector (bottom right). As can be seen, the block detector successfully detects the red block and is able to estimate the centroid of the block (represented as a green dot on the downward facing view to be used as the ground truth). Then, the quadrotor is given waypoints that move it successively closer to the block, showing that the pose estimation is to be done with various camera viewpoints.

[10] Frédéric Jurie and Michel Dhome. Hyperplane approximation for template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):996–1000, 2002.

[11] Iryna Skrypnyk and David G Lowe. Scene modelling, recognition and tracking with invariant image features. In *Third IEEE and ACM international symposium on mixed and augmented reality*, pages 110–119. IEEE, 2004.

[12] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015.

[13] Alexander Krull, Eric Brachmann, Frank Michel, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Learning analysis-by-synthesis for 6d pose estimation in rgb-d images. In *Proceedings of the IEEE international conference on computer vision*, pages 954–962, 2015.

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.