unity

Drone Pose Estimation with Unity Simulation

AUTHORS

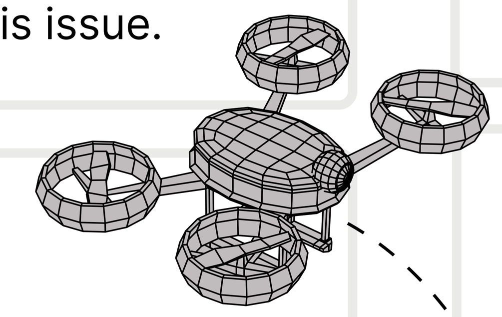
Salehe Erfanian Ebadi (salehe.erfanianebadi@unity3d.com), **Souranil Sen** (souranil@unity3d.com), **Jonathan Leban** (jonathan.leban@unity3d.com), **Priyesh Wani** (priyesh@unity3d.com)

Problem Statement

In this work we demonstrate how the Unity game engine can be used to generate synthetic data for training a pose estimation model, perform path planning, and navigation.

① Motivation

- New applications and ways of navigation have emerged due to the low cost of drones with monocular cameras and advances in computer vision techniques.
- To prove that synthetic data can be used to train a state-of-the-art monocular pose estimation model, we built an end-to-end pipeline for generating a synthetic dataset with Unity.
- By using a synthetic dataset, we had to consider the Sim2Real[1,2] gap. We added multiple visual variations using domain randomizers[3] to address this issue.



② Materials

- Generating a labeled dataset manually for computer vision tasks is extremely difficult and costly.
- We leveraged the Unity game engine combined with the Unity Perception package[4] to generate labeled ground truth data.

Citations

[1] Yu, W., Kumar, V.C., Turk, G. and Liu, C.K., 2019. Sim-to-real transfer for biped locomotion. arXiv preprint arXiv:1903.01390.

[2] Peng, X.B., Andrychowicz, M., Zaremba, W. and Abbeel, P., 2018, May. Sim-to-real transfer of robotic control with dynamics randomization. In 2018 IEEE international conference on robotics and automation (ICRA) (pp. 3803-3810). IEEE.

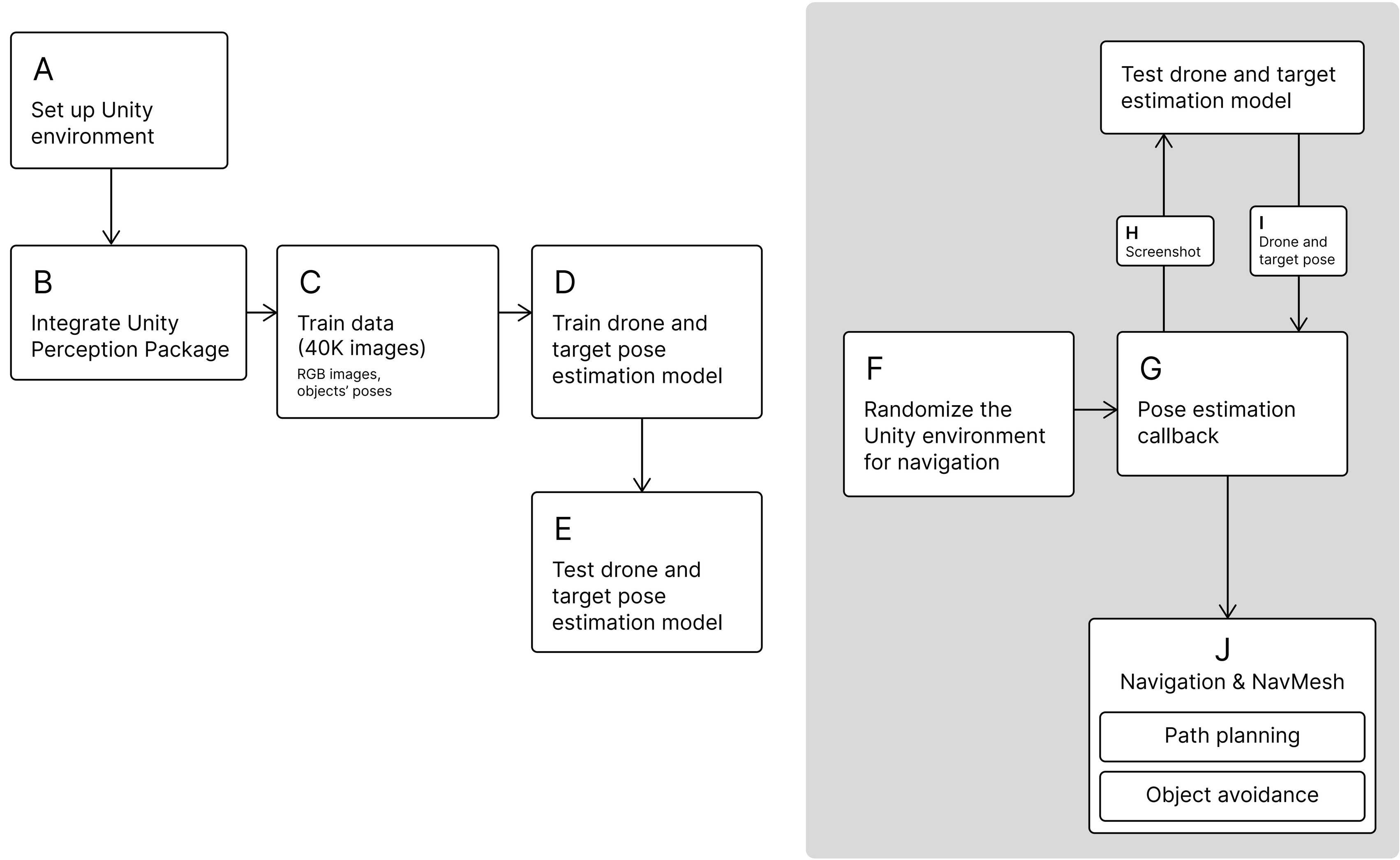
[3] Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W. and Abbeel, P., 2017, September. Domain randomization for transferring deep neural networks from simulation to the real world. In 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS) (pp. 23-30). IEEE.

[4] Unity Perception Package https://github.com/Unity-Technologies/com.unity.perception

[5] Leban, J., Trang, A., Wolf, S., Platin, J., Navarro, A., Ganguly, S. and Gibson, S., 2021. Teaching robots to see with Unity. https://blogs.unity3d.com/2021/03/02/teaching-robots-to-see-withunity

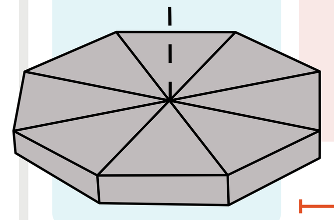
[6] Unity robotics Hub repository: https://github.com/Unity-Technologies/Unity-Robotics-Hub

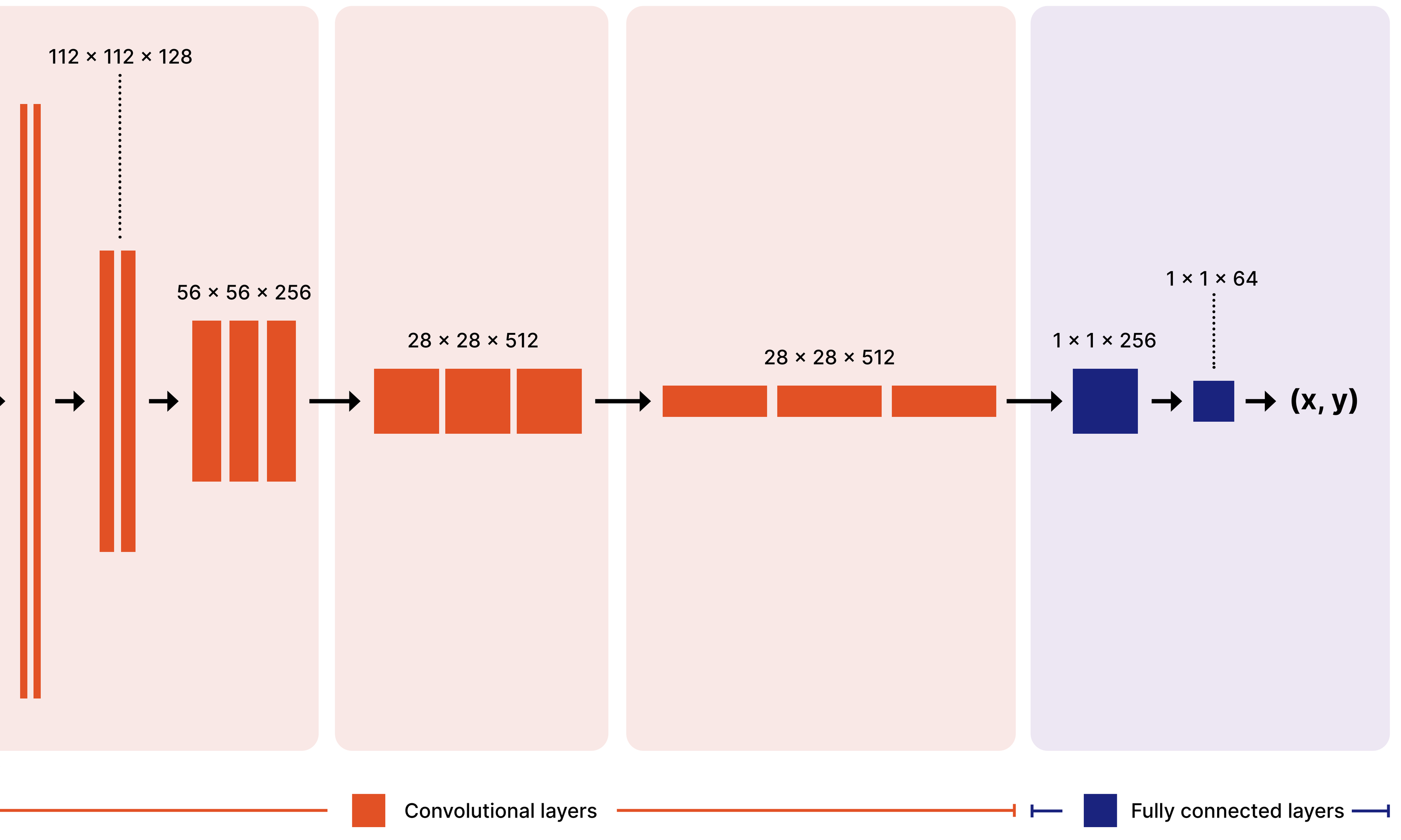
③ Methodology



- Each block represents a step in the drone pose estimation and navigation pipeline.
- Using the rendering engine capabilities and domain randomization we generate a diverse and labeled synthetic dataset from an exocentric view to train a multi-object monocular pose estimation model to predict the pose of a drone and a landing target.
- The estimated position of the drone and the target are sent to the Unity NavMesh module for path planning and navigation[5].

Architecture of the neural network implemented to predict the position of the drone and the target





④ Results

	MSE drone (m)	**MSE target** (m)	**Inference time** (sec)
Medium scenario	0.08059	0.08053	0.00187
Hard scenario	0.09031	0.08809	0.00205

We evaluate the performance of our deep learning model using the Mean Square Error (MSE) in meters. The size of the drone and the cube is 5 cm. Thus, the network has only a 5% error on the prediction of objects' location.

The difficulty of a scenario depends on the elements we randomize during training. In addition to the randomization of the objects' pose, the walls' texture, the lights' pose, the objects' color, and the number of distractor objects, the hard scenario contains the randomization of the camera's pose.

Figure A. ***Before prediction of the pose and navigation***

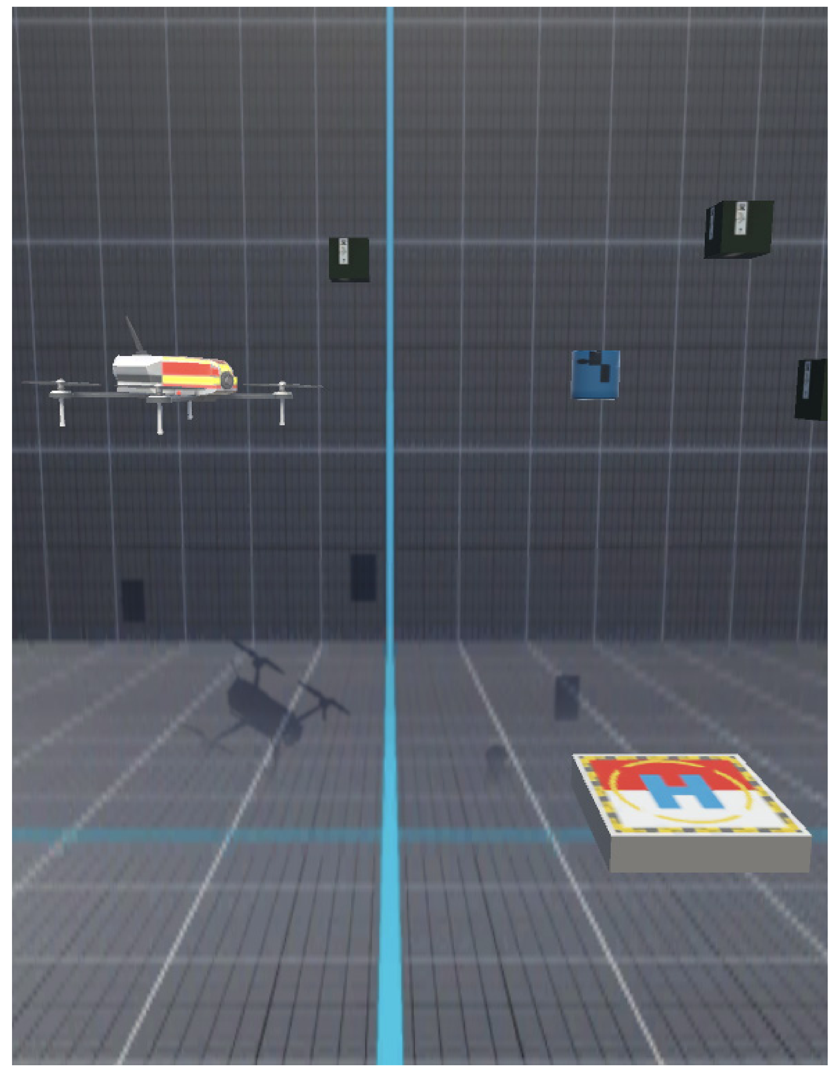
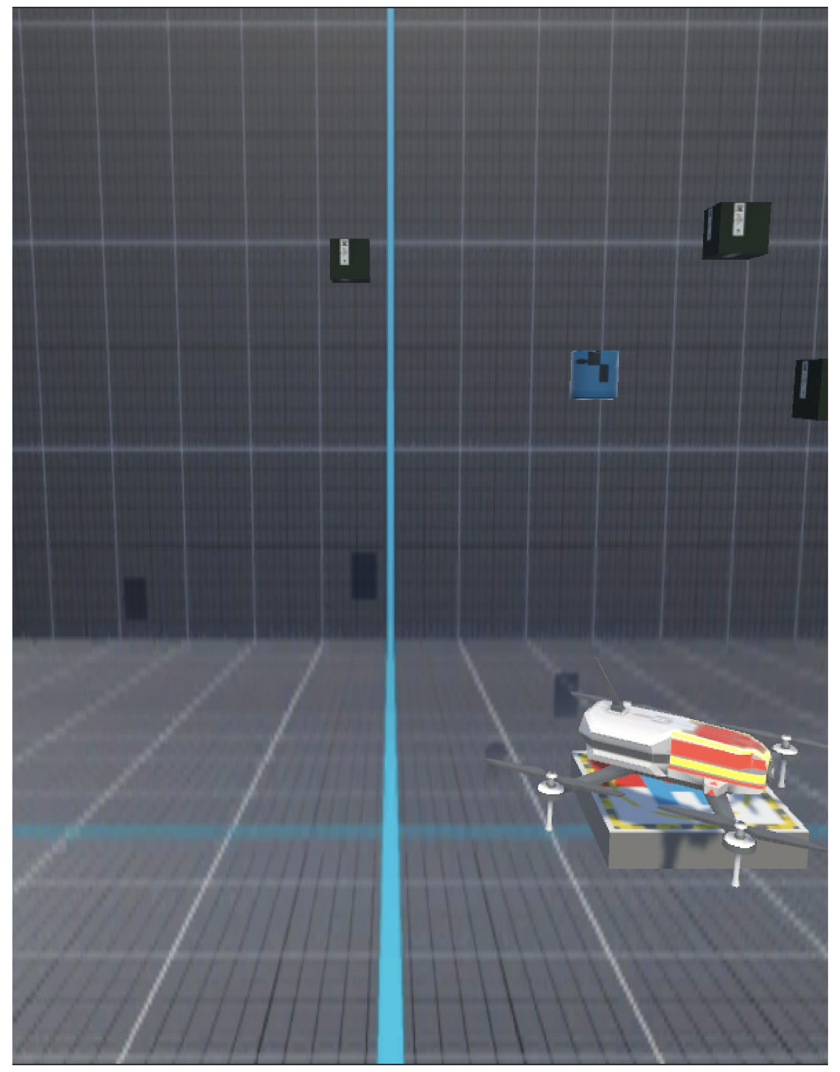


Figure B. ***After prediction, path planning and navigation***



⑤ Conclusion and Future Work

1. Use depth and orientation for estimating 6D pose.
2. Use Unity robotics URDF importer[6], ROS-TCP connector[6] and ROS-TCP endpoint[6] packages to bring a real drone into the simulation to test. The communication between Unity and ROS would be extremely useful for further development on real drones and simulations.
3. Have a Unity → ML pipeline for training and inference with ROS.