https://www.imavs.org/pdf/imav.2018.41

Obstacle Avoidance for UAVs via Imitation Learning from Human Data with 3D-Convolutional Neural Networks

Bumsoo Park and Hyondong Oh*
School of Mechanical, Aerospace, and Nuclear Engineering, UNIST, Ulsan, South Korea

ABSTRACT

In this paper, we present a novel framework that can be applied for the obstacle avoidance of unmanned aerial vehicles (UAVs) based on deep learning methods with supervision from actual human flight control data. Through imitation learning from human flight control data, the UAV is expected to learn how to avoid obstacles without any given rules, and at the same time learn the intuition that humans possess to efficiently deal with unexpected situations. One critical limitation for UAVs is that the number and size of sensors that can be attached is restrictive, hence a monocular camera will be used as the only sensor of the UAV. The simulation is conducted within a simulated environment using Gazebo and ROS (Robot Operating System), where the visual input from the camera and human control input regarding the direction of the UAV are utilized for the training process. The trained model is then validated in terms of how well it imitates a human and how capable it is to avoid obstacles.

1 INTRODUCTION

The recent upsurge of demand for mobile robots with higher levels of autonomy has posed several challenges especially when it comes to the ability for autonomous navigation within a given environment. Among the requirements for fully autonomous navigation of an unmanned agent, obstacle avoidance is considered one of the most fundamental factors, as the agent should be capable of avoiding obstacles during its navigation. Autonomous navigation is usually implemented through path planning methods, where an agent is assigned a certain path given that it has information on the surrounding environment. In general, geometrical constraints such as obstacles between the start and goal point are considered throughout the path planning stage, facilitating obstacle avoidance for the autonomous agent. However, real-time path planning may be unavailable under certain circumstances where the environment is uncertain, or where the path cannot be generated due to issues such as the inability to obtain a map of the surrounding environment. In such cases, local path planning methods take place instead of global path planning methods [1], where obstacle avoidance is considered as the main criterion. Hence, we will focus on the investigation of obstacle avoidance methodologies as it is one of the most fundamental requirements for a mobile agent to exhibit fully autonomous behaviours.

Conventional real-time obstacle avoidance techniques employ rule-based methodologies, where certain rules are set for the agent according to the sensor input. Such researches include obstacle avoidance methods with the usage of ultrasonic sensors [2, 3], or optical flow-based obstacle avoidance methods for monocular camera usage [4, 5]. These methods, however, may also lose its robustness as the agent can undesirably encounter unexpected situations which are not considered in the rule-based methods. Recent researches include data-driven or machine learning-based approaches that use

*contact email: h.oh@unist.ac.kr

apprenticeship learning methods [6], yet the algorithms that are used for the supervision of the machine learning model are still developed upon human-made rules. Hence, in some occasions, learning directly from human demonstrations may provide better results depending on the task, especially when there are constraints on the usage of sensors. In fact, there have been multiple attempts regarding imitation learning from human experts. One of the most primitive researches was proposed by Pomerleau [7], where a single layer of a neural network was used to map the images to the steering angles, enabling the agent to stay on the road. Other work regarding how to teach an aircraft to fly using human data [8] has been introduced by Sammut, Hurst, Kedzier, and Michie. In this work, decision trees were used to train and design the autopilot for the aircraft. One representative example of imitation learning from human data was presented by Ross, Gordon, and Bagnell [9], using the DAGGER (Dataset Aggregation) algorithm. Here, a policy is determined online in an iterative manner, using the aggregated data from the algorithm. This algorithm was also used for vision-based autonomous navigation in forest trails [10]. Kim and Chen [11] presented an imitation learning framework for drones to track and follow a certain target based on neural networks.

Under the main assumption that there underlies a certain relationship or function between visual inputs and actions of a human when it comes to obstacle avoidance, we focused on teaching an agent so that it can map the image inputs to the heading directions for obstacle avoidance. Through sufficient acquisition of camera data and the corresponding control inputs generated from human experts, our neural network is likely to learn the underlying rule for obstacle avoidance. In this paper, we present a framework for obstacle avoidance with deep learning-based imitation learning methods. 3D-CNN models [12] will be utilized for the imitation learning framework, as obstacle avoidance with a monocular camera is considered a sequential task.

A general description of the background will be given in Section 2. The methodology for the proposed imitation learning framework followed by details on how the neural network model is trained are discussed in Section 3. In Section 4, we will discuss the training results of the neural network, and validate the performance of the obstacle avoidance capabilities by applying the trained neural network model to a simulated drone. Finally, Section 5 includes the conclusion and discussion for this work.

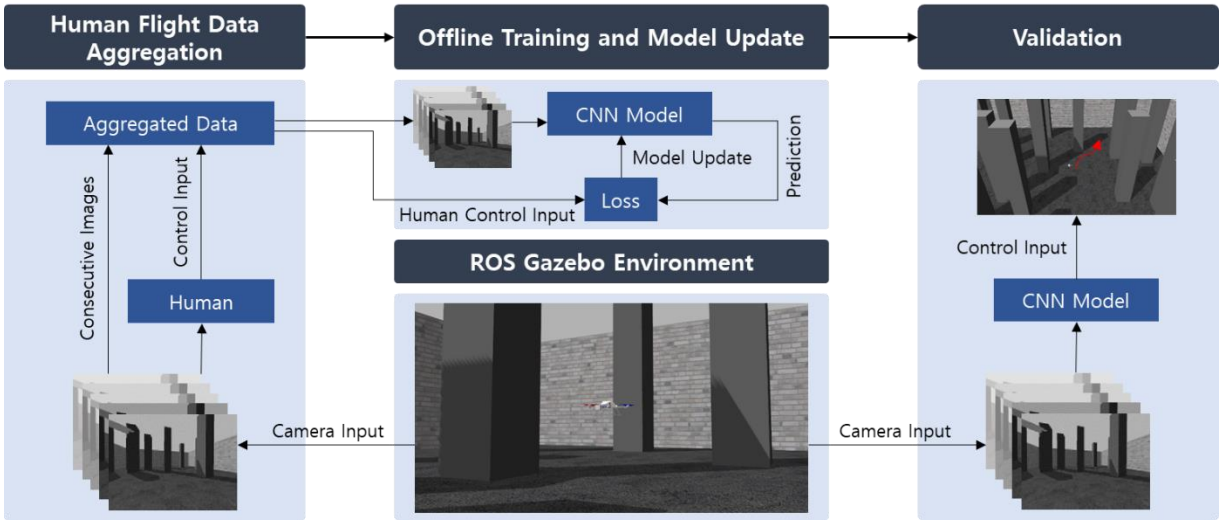

Figure 1 – Proposed imitation learning framework

2 BACKGROUND ON NEURAL NETWORKS

The advent of deep learning technologies has brought about substantial changes throughout various fields of technology. Based on multiple layers of artificial neural networks, deep learning is well known for its ability to approximate complex nonlinear functions, performing complicated classification or regression tasks.

Unlike fully connected layers, Convolutional Neural Networks (CNN) [13] share weights by using a convolution filter among its layers. This enables the neural network to not only reduce the number of weights, but also automatically extract local features from the input. Such characteristics grant CNNs a great advantage over other fully connected neural networks especially when it comes to image classification.

For conventional CNNs, a convolution operation is conducted for each layer with a 2D kernel granting CNNs considerable capabilities regarding image processing. At the same time, CNNs possess the same limitations that other feedforward neural networks have; feedforward neural networks are unable to consider sequential data. Although attempts to combine CNNs with RNNs (Recurrent Neural Networks) [14] [15], have proved to be efficient tasks such as in video classification and scene labelling, 3D-CNNs have also been renowned for its capability to extract spatiotemporal features from a given sequence of data. 3D-CNNs utilize a 3D kernel for the convolution operation, overcoming the previous limitations of 2D-CNNs. As obstacle avoidance is considered a sequential task, 3D-CNNs will be utilized in this study.

3 APPROACH

3.1 Imitation Learning Framework

For a drone to avoid obstacles during autonomous navigation, we utilize deep learning techniques to approximate the function between the visual inputs and controls. We first start by aggregating image and control input data from human flight demonstrations. During the demonstration, a human expert steers a simulated drone within the Gazebo environment. The accumulated data is then processed so that it can be used for the training of the neural network. Using this training data, the training process of the neural network is done offline. After the training process is finished, the neural network model is capable of imitating human expert decisions. For real-time applicability, a Gazebo-Python communication node is established using ROS (Robot Operating System). Here, the Python node receives images from the Gazebo simulator, where each image is fed through the neural network producing control input predictions for what a human expert may have chosen. Hence the drone is able to mimic expert behaviours, and consequently avoid obstacles. The overall framework is shown in Figure 1.

3.2 Training Data Acquisition and Processing

The training data for the neural network was obtained from the Gazebo simulator, using ROS. A human expert demonstration of obstacle avoidance within the simulated environment was recorded, i.e., control inputs from the human expert flight and the corresponding image data was aggregated into the training dataset. The sampling rate for the images was 15 frames per second, which is identical for the sampling rate of the control inputs. For each control input data, a value of 0, 1, or 2 is saved to indicate whether the human expert has commanded the drone to steer left, right, or move straight, respectively. The drone increases or decreases its heading angle upon left or right control inputs, and incrementally returns its heading angle to 0 when a ‘go straight’ command is given. A total of two hours of flight data from the expert was collected. Considering the fact that the environment did not exhibit much variance with respect to the colour and that a grayscale image contains adequate information for the identification of obstacles, each image from the training dataset was converted into a grayscale

image. After the whole dataset is processed so that it can be used for training, we separate the dataset into a training and test dataset with a ratio of 7:3. We use the training dataset to train the neural network model, and the test dataset to verify the performance of the neural network. An example of the training environment and training image is shown in Figure 2.

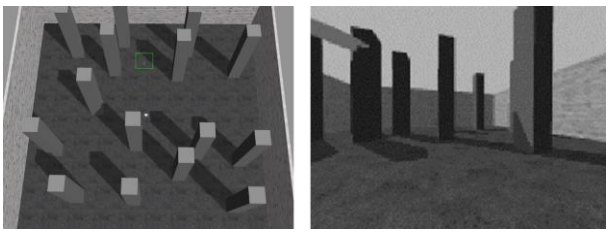

Figure 2 – Training environment (Left) and camera data (Right).

3.3 Neural Network Architecture and Training

As we are dealing with sequences of image data, we used a 3D-CNN architecture for the experiment. A total of 4 convolutional layers and 2 max pooling layers were used, and a fully-connected hidden layer was added at the end of the network. A Softmax output layer is added at the end of the network. The illustration of the CNN structure utilized in this study is shown below in Figure 3. For each layer, the first number indicates the step size of the images used for each batch, the next two numbers indicate the size of the input for each layer, and the last number indicates the number of channels or filters used. The size of the convolution filter was 3x3x3 and a 2x2x2 kernel was used for the pooling layer.

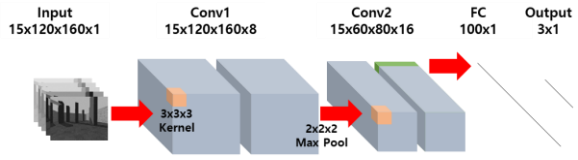

Figure 3 – 3D-CNN architecture for imitation learning

The training process of the 3D-CNN is done offline using the training data above. The 3D-CNN was trained with an epoch of 6,000, and the learning rate was 0.001. We used Adam Optimizer as the optimizer and truncated normal initializer for the initializer.

4 SIMULATION RESULTS

The imitation loss is as shown in Figure 4. The blue lines indicate the training loss whereas the red lines indicate the test loss. The final training loss was 0.006. The test loss was obtained by feedforwarding images from the test dataset, which were not used during the training process. The training accuracy was 96% and the test accuracy was 81%. Despite the minor discrepancy between the prediction values and labels for the test dataset, we can imply that the 3D-CNN is capable of mimicking human decisions given a set of images.

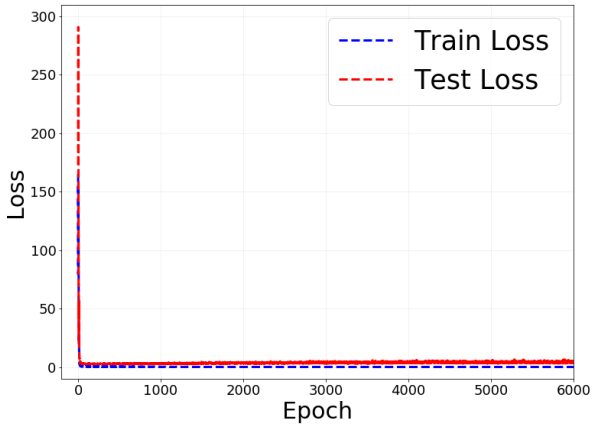

Figure 4 – Training and Test Loss

We applied the 3D-CNN to the drone to verify the effectiveness of the trained model regarding obstacle avoidance, implementing the real-time Gazebo-Python communication node using ROS.

Three unknown environments with randomly placed obstacles were used for the validation of the obstacle avoidance capability. Unlike the rectangle map used for the aggregation of the training data, the test map was designed as a long rectangular figure so that the drone could be tested if it is capable of crossing the entire map without crashing into any obstacles. In addition, the obstacles were arranged in a manner that the

drone has not been able to see during the training phase to test the robustness of the obstacle avoidance capability. This demonstrates the fact that the 3D-CNN has learned a general rule for obstacle avoidance using a monocular camera, instead of learning how to navigate within the training environment only. Figure 5 shows an example of one of the maps used for the validation. The trajectory of the drone is shown in Figure 6. The trajectories show that the drone was able to successfully cross the entire maps in all three test maps, suggesting that the 3D-CNN has learned how to mimic a human expert and avoid obstacles during navigation.

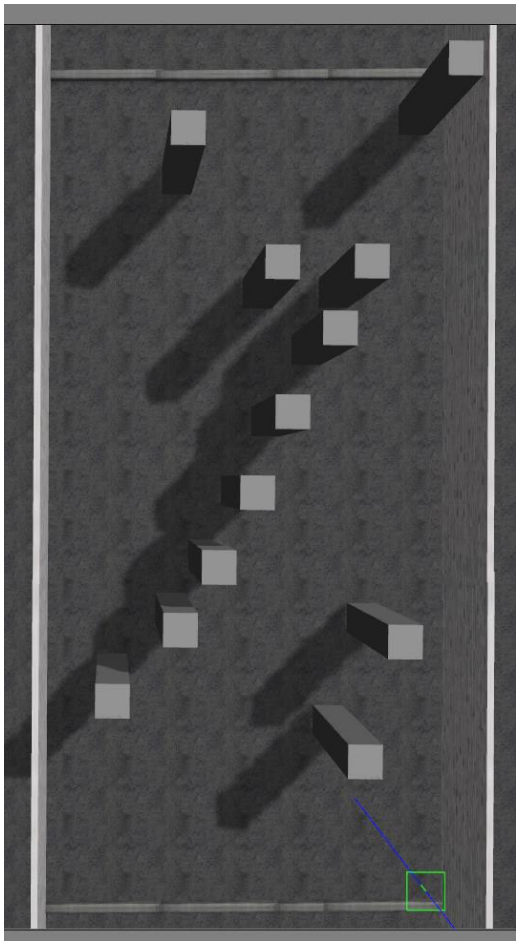

Figure 5 – Test environment example

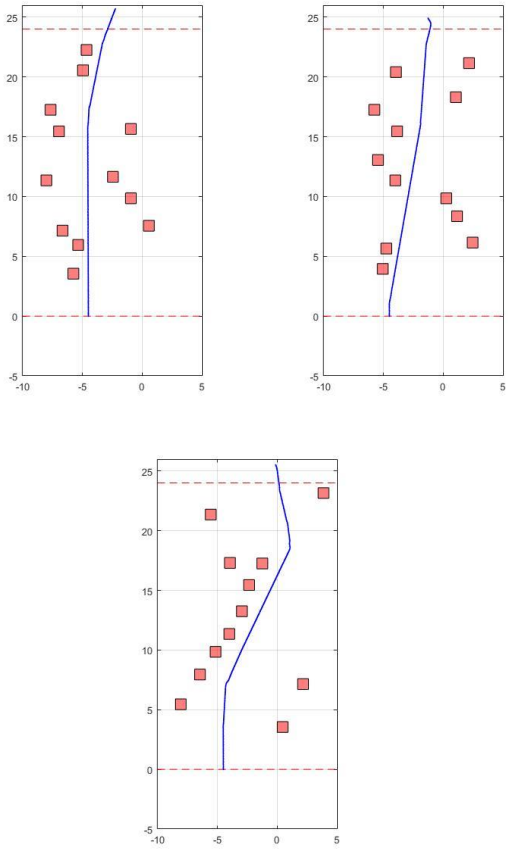

Figure 6 – Drone trajectories within the three test environments

5 CONCLUSION

An imitation learning framework using 3D-CNNs for obstacle avoidance has been investigated in this paper. Through our experiments with a drone in a simulated environment, the 3D-CNN model exhibited human-like behaviour, and at the same time we were able to achieve acceptable results for obstacle avoidance without the utilization of complicated neural network structures, proving the feasibility of applying imitation learning to drones to conduct a certain task.

Considering the fact that the imitation loss dramatically decreased after the first few epochs, the obstacle avoidance task can be considered a rather simple task to train, and thus the applicability to more complicated environments or more sophisticated tasks are yet to be explored.

ACKNOWLEDGEMENTS

This work was supported by the technology innovation project (S2439592) of Small and Medium Business Administration.

REFERENCES

[1] Chu, K., Lee, M., & Sunwoo, M. (2012). Local path planning for off-road autonomous driving with avoidance of static obstacles. IEEE Transactions on Intelligent Transportation Systems, 13(4), 1599-1616.

[2] Borenstein, J., & Koren, Y. (1988). Obstacle avoidance with ultrasonic sensors. IEEE Journal on Robotics and Automation, 4(2), 213-218.

[3] Borenstein, J., & Koren, Y. (1991). The vector field histogram-fast obstacle avoidance for mobile robots. IEEE transactions on robotics and automation, 7(3), 278-288.

[4] Souhila, K., & Karim, A. (2007). Optical flow based robot obstacle avoidance. International Journal of Advanced Robotic Systems, 4(1), 2.

[5] Song, K. T., & Huang, J. H. (2001). Fast optical flow estimation and its application to real-time obstacle avoidance. In Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on (Vol. 3, pp. 2891-2896). IEEE.

[6] Kahn, G., Zhang, T., Levine, S., & Abbeel, P. (2017, May). Plato: Policy learning using adaptive trajectory optimization. In Robotics and Automation (ICRA), 2017 IEEE International Conference on (pp. 3342-3349). IEEE.

[7] Pomerleau, D. A. (1989). Alvinn: An autonomous land vehicle in a neural network. In Advances in neural information processing systems (pp. 305-313).

[8] Sammut, C., Hurst, S., Kedzier, D., & Michie, D. (1992). Learning to fly. In Machine Learning Proceedings 1992 (pp. 385-393).

[9] Ross, S., Gordon, G., & Bagnell, D. (2011, June). A reduction of imitation learning and structured prediction to no-regret online learning. In Proceedings of the fourteenth international conference on artificial intelligence and statistics (pp. 627-635).

[10] Ross, S., Melik-Barkhudarov, N., Shankar, K. S., Wendel, A., Dey, D., Bagnell, J. A., & Hebert, M. (2013, May). Learning monocular reactive uav control in cluttered natural environments. In 2013 IEEE international conference on robotics and automation (pp. 1765-1772). IEEE.

[11] Kim, D. K., & Chen, T. (2015). Deep neural network for real-time autonomous indoor navigation. arXiv preprint arXiv:1511.04668.

[12] Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 4489-4497).

[13] LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010, May). Convolutional networks and applications in vision. In ISCAS (Vol. 2010, pp. 253-256).

[14] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2625-2634).

[15] Pinheiro, P. H., & Collobert, R. (2014). Recurrent convolutional neural networks for scene labeling. In 31st International Conference on Machine Learning (ICML) (No. EPFL-CONF-199822).