Enhancement in Novel State Representation Learning for Intelligent Navigation Systems

Geng LiangYu   
MTech (IS) Year 2 (PT)  
NUS (ISS) Singapore  
e0384909@u.nus.edu

Han Dongchou Francis  
MTech (IS) Year 2 (PT)  
NUS (ISS) Singaporee0385045@u.nus.edu

Tan Chin Gee  
MTech (IS) Year 2 (PT)  
NUS (ISS) Singapore  
e0384927@u.nus.edu

*Abstract*—This work evaluated the improvement gained from refining an algorithm used to address the simulation-to-real transfer (sim-real) problem in autonomous first-person view (FPV) drone navigation. A deeper ResNet and DenseNet model replaced the original ResNet based feature encoder of the Cross-Modal Variational Auto-Encoder (CMVAE) algorithm used for solving the sim-real problem. Feature encoders with the deeper ResNet and DenseNet architectures demonstrated better navigational performance compared with the original feature encoder.

# INTRODUCTION

*A. Simulation-to-Real Transfer Problem*

The training of autonomous navigation system, as it is practiced now, relied heavily on data intensive methods such as deep learning and reinforcement learning [1] [2]. The data collection challenges of these methods are well-understood, and these problems can generally be addressed by using simulated environments [3] [4]. Collecting data using a simulator is a feasible way of collecting large quantity of data that otherwise would have been more difficult if collected manually. Simulated data, as compared with real-world data, are cheaper and easier to collect which makes it an ideal candidate for data collection.

However, the use of simulated data also poses unique challenges for the development of autonomous navigation system. The drawback of models trained on simulated data is arguably the decay of performance when models are deployed in the real world. This is known as the reality gap or simulation-to-real transfer (sim-real) problem [5] [6].

The crux of the sim-real problem is that a simulated environment, despite the best effort to make it as realistic to the real-world as it can be with current technology, is still not an actual copy of the real-world [7]. There are perceivable visual differences between simulated and real environments, and the complexity of forces and dynamics acting on objects in the real-world is simplified in simulation. Hence, simulated environment cannot fully model all aspects of reality. All these differences reduce the performance of simulation models deployed in the real-world.

Prior works had tried addressing the sim-real problem by concentrating on specific characteristics of the issue. These works typically fall under the approach of (1) creating better simulators that minimizes the differences between simulation and reality, or (2) making simulation models more robust to the effect of the sim-real problem. The first approach is driven by the idea that as the simulation becomes more accurate, so will the transition of simulation models to reality becomes smoother. The second approach abandons the idea of a fully accurate simulation, in favor of methods that aim to generalize simulation models to real-world deployment. The typical methods under this approach are domain randomization and domain adaption.

*B. Simulators improvement*

A focus on improving the fidelity of simulated environments has led to more visually realistic simulators, such as the Microsoft Aerial Informatics and Robotics Simulation (AirSim) [8] and CARLA [9]. This approach of developing ever more accurate simulators will continue unabated in the foreseeable future, bolstered by increasing computing power and advancement in hardware technology. Despite these expected advancements in simulators, it is likely that simulation models will still fail to properly generalize to the real-world due to the slightest discrepancy between these two environments [10].

*C. Domain Randomization*

Domain randomization [11] [12] and domain adaptation [13] [14] [15] are methods that seek to reduce the impact of performance decay that is triggered by transferring simulation models to the real-world. Domain randomization takes place during the simulation training of models, where features that are non-relevant to the task, such as lighting condition and the relative position of elements, are changed randomly. Models are therefore exposed to a multitude of different environments where important aspects of the environment are kept unchanged. Hence, the sim-real problem is minimized in these models because they had learnt to ignore non-informative elements for the real-world task, which may confuse it, through randomization. The obvious shortcoming of the domain randomization approach is the greater efforts and costs required to train models in many different simulated environments.

*D. Domain Adaption*

Some proposed solutions viewed the incorporation of labeled real-world data as an answer to the sim-real problem. Domain adaption focuses on fine-tuning trained simulation models to bridge the reality gap, so that models can work well in the real-world. Labeled data collected from the real-world is often required when constructing the bridge that links simulated models to the real-world. There is ongoing effort to minimize the size of labeled data, and therefore preserve the benefits conferred by the training of models in a simulation. An example solution is the use of Generative Adversarial Network (GAN) [16] in combination with 93 labeled and 186 unlabeled images to successfully deploy simulation-trained visuomotor policies in the real-world [17]. Despite the efforts of these works, the reliance on labeled data does negates some of the benefits gained from using a simulator for model training. A more satisfactory approach that uses only simulation data is needed.

*E. Representation Learning Using Cross-Modal Variational Auto-Encoder*

A solution that eliminates the need for labeled real-world data uses Cross-Modal Variational Auto-Encoder (CMVAE) [18] for representational learning solely within a simulation environment [19]. The CMVAE feature encoder architecture differed from vanilla autoencoders by possessing only one encoder but having one decoder for each data modalities. This work focused on autonomous first-person view (FPV) drone navigation, which is similar in nature to the task of drone racing that only rely on images from a single RGB camera for navigation.

The CMVAE’s approach takes the two different modalities of the (1) video-stream of the drone’s FPV camera and (2) poses of racing gates in the simulated environments, and learnt a representation of the drone’s state that has much fewer dimensions than the raw inputs of both modalities. These state representations can reconstruct the original inputs approximately despite its smaller size. This is termed as the concept of *autoencoding* in unsupervised representation learning [20].

These state representations learnt by the CMVAE’s encoder capture dense informatic latent variables that are most important for decision-making in autonomous navigation of drone through gates with unknown locations. A navigation policy trained with these latent variables retains high real-world performance and addresses the sim-real problem, because it has learnt to focus on relevant features of the real-world to guide its navigation and minimizes the influences of trivial differences between its training and deployment environments on its performance. This effect was attributed to the prevention of overfitting through the regularization brought about by forcing CMVAE to learn the latent variables for multiple data modalities.

After obtaining the state representation, the next step taken was imitation learning for training the navigation policy. Behavior cloning (BC) [21], which is a simplistic imitation learning algorithm, was used to learn the optimal path through the gates that was generated by a minimum jerk trajectory planner [22]. This completes the entire training, and the drone is now ready for real-world deployment.

# RELATED WORKS

*A. Drone Racing*

Drone racing is a sport in which human pilots fly their drone at high speed through racing gates, guided only by live video-stream from the drone’s FPV camera [23]. There are numerous difficult technical challenges in developing autonomous racing drones, and of which this work will address the challenge of optimal real-world navigation of simulation model. Related works that tackled key technical challenges of autonomous drone navigation are presented.

*B. Vision-Based Drone Navigation*

Vision-based autonomous drone navigation is characterized by its primary use of visual sensors such as monocular cameras and stereo cameras [24]. Vision-based drone, like all mobile robots, has a typical architecture that comprises perception systems to detect surroundings, localization and mapping systems to know its states and the environment, path planning systems to chart the route, and motor control systems to move the robot [25]. Due to visual sensor’s ability to offer user a tremendous amount of information, the extraction of the features being observed in relation to the drone for determining positioning information is not a simple task.

Because of limitations to of the payload of the drone, it is can be impractical for some small drones to be outfitted with multiple sensors. Therefore, a more general approach of enhancing the drone’s environmental perception ability, instead of obtaining better pose estimation by multiple sensor data fusion is needed [26]. Monocular cameras are especially suited for applications where compactness and minimum weight are critical. In addition to that, lower price and flexible deployment make them a good option for drones.

*C. Representation Learning of State for Navigation Policy*

State representation learning focuses on extracting abstract and compact features from data in environment where the action of an agent affects said environment [27]. The resultant features need to be encoded with essential information required for the task, and to disregard irrelevant data. Training on a compact set of informationally rich features permits more efficient learning of navigation policies, compared with doing so using the raw sensory data [28].

Adversarial networks were used for the learning of state representations [29]. The GAN-based state representation learning framework is an alternative approach to VAE-based ones. There is a perception in the literature of GAN-based framework being less suitable for state representation learning because they were slower and harder to train compared with their VAE counterparts [30] [31] [32] [33]. Other issues concerning the GAN-based approach were training instability and the requirement for having a priori knowledge of the data [34]. VAE-based framework, however, was more effective at extracting useful representations of data [35].

# SOLUTION

*A. Re-designing the Feature Encoder Architecture*

It is obvious that the use of CMVAE in training an autonomous drone navigation system does not require localization and an explicit map of the environment to navigate, and this approach also does not require any real-world training of the simulation model. Due to its higher utility to the task of autonomous navigation model development, this work has chosen the CMVAE’s approach and examined the extent to which its feature encoder can be refined to yield higher performance during real-world drone deployment. Other structures of the original model were not modified because they were not relevant for this work’s aim of improving representational learning in autonomous FPV drone navigation.

The goal of this work was to enhance the process of learning a latent state representation from different data modalities, as described by [19] in their two-steps approach of training a FPV drone to autonomously navigate a set of racing gates. Using a latent state representation to train navigation policy reduces the impact caused by sim-real transfer. The feature encoder is the key module responsible for generating the latent state representation, and thus this work focused on improving the performance of this module. Said enhancements were implemented by re-designing the feature encoder with state-of-the arts computer vision algorithms ResNet [36] and DenseNet [37].

*B. Microsoft Aerial Informatics & Robotics Simulation*

Training and evaluation of the re-designed architecture was conducted in the open-source AirSim environment, which is a simulator for experimenting with algorithms for autonomous vehicles such as drones and cars. The simulated environment uses the Unreal Engine built of AirSim to create a controllable training and testing platform, that is not only highly realistic in terms of the physics and visuals of the real-world, and also drastically reduces the costs and time involved in the collection of massive amount of training data required by deep learning and reinforcement learning.

AirSim is equipped with a generic drone model that when combined with its pre-loaded RGB camera quicken the pace of experimentation right from the start. Crashing the drone repeatedly in AirSim does not incurs costs, unlike that of training with real drone. Images, data streams and vehicular control in AirSim are communicated programmatically through its Application Programming Interfaces (APIs).

*C. Tello Drone*

A Tello multirotor drone served as the autonomous vehicular platform for testing the trained model in the real environment. This commercially available drone was chosen for its front mounted RGB camera that matched the configuration of the simulated drone and its ease of programming in Python.

# METHOD

*A. State Representation Learning*

The CMVAE architecture forms the foundation of the feature encoder, and also enable it to generate the latent state representation from a combination of RGB images and data on the pose of racing gates relative to the frame of the drone. Feature encoder samples latent state representation from a normal distribution.

Training procedure followed the same steps outlined in [18]. Dataset was comprised of 50K 64 x 64 images that were paired with their respective data of relative gate poses seen in these images. Training and validation were allocated 80% and 20% of the dataset respectively. Training losses recorded included (i) MSE loss between actual and reconstructed images, (ii) MSE loss for gate pose reconstruction, and (iii) Kullback-Leibler (KL) divergence loss for each sample.

*B. Imitation Learning*

Imitation learning proceeded exactly as described in [19]. A minimum jerk trajectory planner in combination with a pure-pursuit path tracking controller generated an expert trajectory. Then the velocity commands that match the expert trajectory were paired with their corresponding FPV images to create the training dataset for imitation learning.

This work used the imitation learning technique, behavior cloning (BC) to train the navigation policy [38 – Osa 2018]. Training of the control network proceeded with trained weights of the feature encoder kept unchanged. A total of three navigation policies were trained for different feature encoder architectures: (i) original, (ii) ResNet, and (iii) DenseNet. Feature encoder weights were not updated during the training of the navigation policy.

# RESULTS

Training in the AirSim took place in the map of an open soccer field. Red colored squares positioned over the entire field served as “gates” for the drone to fly through. Simulated flight assessments were conducted on navigation policies obtained with three different feature encoders. The assessment put each navigation policy through a randomized 50m circular track of eight gates for 10 trials, and under 2 weather conditions: sunny and foggy (Figure x & x). The sunny weather condition was identical to the weather in data used in training. A count of gates that were successfully navigated serves as the performance metric.

Table x shows the average number of successfully navigated gates in the assessment for different latent encoder architectures. When the weather is sunny and thus have the same condition during training, there were no considerable differences in navigation performance among all three feature encoders. Unsurprisingly, navigation performance of all feature encoders suffered under the foggy condition. However, feature encoders with the ResNet and DenseNet architecture demonstrated less degradation of performance compared with the original feature encoder.

# DISCUSSION & CONCLUSION

The focus of this work was the enhancement of a novel approach in tackling the sim-real problems. It is a departure from the typical practices of domain randomization and domain adaptation, by using CMVAE to jointly convert two or more simulated sensory data to highly informatic representation variables with much lower dimensionality. Navigation policies trained with these variables demonstrated fewer performance issues when in the real world.

The re-designed feature encoder architecture achieved better performance in FPV drone navigation compared with the original architecture in simulated flight assessments under varying weather effects.

Future work can explore the use of advance imitation learning techniques and continue to explore the application of the CMVAE framework for robotic manipulation tasks.

##### References

1. O. Walker, F. Vanegas, F. Gonzalez, and S. Koenig, “A Deep Reinforcement Learning Framework for UAV Navigation in Indoor Environments,” in *IEEE Aerospace Conference Proceedings*, vol. 2019-March. IEEE Computer Society, Mar. 2019.
2. C. Wang, J. Wang, X. Zhang, and X. Zhang, “Autonomous navigation of UAV in large-scale unknown complex environment with deep reinforcement learning,” in *Proc. IEEE Global Conf. Signal Inf. Process*. (GlobalSIP), Nov. 2017, pp. 858-862.
3. K. Kang, S. Belkhale, G. Kahn, P. Abbeel, and S. Levine, “Generalization through simulation: Integrating simulated and real data inot deep reinforcement learning for vision-based autonomous flight,” in *IEEE Intl Conf. on Robotics and Automation* (ICRA), 2019.
4. C. Sampedro, A. Rodriguez-Ramos, I. Gil, L. Mejias, and P. Campoy, “Image-based visual servoing controller for multirotor aerial robots using deep reinforcement learning,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 979-986.
5. Jakobi, N., Husbands, P., & Harvey, I. (1995). Noise and the reality gap: The use of simulation in evolutionary robotics. In F. Moran, A. Moreno, J. Merelo, & P. Chacon (Eds.), Advances in artifical life: Proceedings of the Third European Conference on Artifical Life. Berlin: Springer-Verlag.
6. S. James, A. J. Davision, and E. Johns, “Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task,” *CoRR*, vol. abs/1707.02267, 2017.
7. M. Neunert, T. Boaventura, and J. Buchli, “Why off-the-shelf physics simulators fail in evaluating feedback controller performance: A case study for quadrupedal robots,” in Advances in Cooperative Robotics. World Scientific, 2017, pp. 464-472.
8. S. Shah, D. Dey, C. Lovett, and A. Kapoor. AirSim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, pages 621-635. Springer, 2018.
9. A. Dosovitskiy, G. Ros, F. Codvilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In Proceedings of the 1st Annual Confernce on Robot Learning, pages 1-16, 2017.
10. K. Cobbe, O. Klimov, C. Hesse, T. Kim, and J. Schulman. Quantifying generalization in reinforcement learning. CoRR, abs/1812.02341, 2018.
11. J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” *CoRR*, vol. abs/1703.06907, 2017.
12. A. Loquercio, E. Kaufmann, R. Ranftl, A. Dosovitskiy, V. Koltun, and D. Scaramuzza, “Deep drone racing: From simulation to reality with domain randomization,” *arXiv preprint arXiv: 1905.09727*, 2019.
13. G. Csurka. Domain adaptation for visual applications: A comprehensive survey. arxiv: 1702.05374, 2017.
14. Bousmalis, K., Irpan, A., Wohlhart, P., Bai, Y., Kelcey, M., Kalakrishnan, M., Downs, L., Ibarz, J., Pastor, P., Konolige, K., Levine, S., and Vanhoucke, V. (2017). Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping. *ArXiv e-prints*.
15. F. Golemo and A. A. Taiga, “Sim-to-Real Transfer with Neural-Augmented Robot Simulation,” no. CoRL, 2018.
16. Goodfellow, Ian J., Pouget-Anadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron C., and Bengio, Yoshua. Generative adversarial nets. *NIPS*, 2014.
17. Zhang, F., Leitner, J., Ge, Z., Milford, M., Corke, P. Adversarial discriminative sim-to-real transfer of visuo-motor policies. *Int. J. Robot. Res*. 2019, 38, 1229-1245.
18. A. Spurr, J. Song, S. Park, and O. Hilliges, “Cross-modal deep variational hand pose estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 89-98.
19. R. Bonatti, R. Madaan, V. Vineet, S. Scherer, and A. Kapoor, “Learning visuomotor policies for aerial nvigation using cross-modal representations,” *arXiv: 1909.06993v2*, 2020.
20. M. Tachannen, O. Bachem, and M. Lucic, “Recent advances in autoencoder-based representation learning,” *arXiv, abs/1812.05069*, 2018.
21. T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, J. Peters et al., “An algorithmic perspective on imitation learning,” Foundations and Trends in Robotics, vol. 7, no. 1-2, pp. 1-179, 2018.
22. C. Richter, A. Bry, and N. Roy, “Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments,” in Robotics Research. Springer, 2016, pp. 649-666.
23. E. Kaufmann, A. Loquercio, R. Ranftl, A. Dosovitskiy, V. Koltun, and D. Scaramuzza. Deep drone racing: Learning agile flight in dynamic environments. In *CoRL*, 2018.
24. Yuncheng. Lu, Zhucun. Xue, Gui-Song. Xia, and Liangpei. Zhang (2018) A survey on vision-based UAV navigation, Geo-spatial Information Science, 21:1, 21-32.
25. C. Badue, R. Guidolini, R. Vivacqua Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. Paixão, F. Mutz, L. Veronese, T. Oliveira-Santos, and A. F. De Souza, ‘‘Self-driving cars: A survey,’’ 2019, arXiv:19
26. Shen, S., Y. Mulgaonkar, N. Micheal, and V. Kumar. 2014. “Multi-Sensor Fusion for Robust Autonomous Flight in Indoor and Outdoor Environments with a Rotorcraft MAV”. Paper Presented at the Robotics and Automation (ICRA), IEEE International Conference.
27. T. Lesort, N. Diaz-Rodriguez, J. -F. Goudou, and D. Filliat., “State representation learning for control: An overview”, CoRR, vol. abs/1802.04181, 2018.
28. Munk, J., Kober, J., and Babuska, R. (2016). Learning state repreentation for deep actor-critic control. In *Proceedings of the 55th Conference on Decision and Control (CDC)*, pages 4667-4673. IEEE.
29. Xi Chen, Yan Duan, Rein Houthooft, John Schulman, IIya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In NeurIPS, 2016.
30. Z. Lin, K. K. Thekumparampil, G. Fanti, S. Oh. (2020). InfoGAN-CR and ModelCentrality: Self-supervised model training and selection for disentangling GANs, *arXiv: 1906.06034v3*.
31. Yuhuai Wu, Yuri Burda, Ruslan Salakhutdinov, and Roger B. Grosse. On the quantitative analysis of decoder-based generative models. CoRR, abs/1611.04273, 2016.
32. Tian Qi Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. In NeurIPS, 2018.
33. Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In ICML, 2018.
34. I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. Beta-VAE: Learning basic visual concepts with a constrained variational framework. In Proc. ICLR, 2017.
35. Wonkwang Lee, Donggyun Kim, Seunghoon Hong, and Honglak Lee. High-Fidelity Synthesis with Disentangled Representation. arXiv preprint arXiv:2001.04296, 2020.
36. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385, 2015.
37. Gao Huang, Zhuang Liu, Kilian Q. Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. arXiv preprint arXiv:1608.06993, 2016.
38. T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, J. Peters et al., “An algorithmic perspective on imitation learning,” Foundations and Trends in Robotics, vol. 7, no. 1-2, pp. 1–179, 2018.