

CSCI 5561 PROJECT PROPOSAL

AUTONOMOUS NAVIGATION AND COLLISION AVOIDANCE FOR DRONES

February 17, 2024

Marcus Berg
Mason Hawyer
Yuanhao Cai

University of Minnesota
Professor: Junaed Sattar

1 Motivation

Autonomous navigation is a challenging task for a robot. And the problem only becomes more complex when an agent is able to move in multiple degrees of freedom, and the sensor input is noisy. Classical Algorithms split this problem into two sub-tasks Mapping and Planning. These algorithms attempt to use partial observations to distill the world into a discrete map and plan within that map. These added levels of abstractions incur compounded errors making the problem even harder to solve. Despite significant progress in classical algorithms for mapping and planning, the challenges persist, especially when dealing with real-world scenarios where uncertainties and dynamic environments are prevalent. As robots and autonomous systems become more sophisticated, the need for robust solutions to navigate through complex and unpredictable surroundings becomes even more pronounced. One emerging area that holds promise in addressing these challenges is the application of machine learning, particularly deep learning techniques, in autonomous navigation. Machine learning approaches leverage the power of neural networks to learn directly from data, allowing robots to adapt and make decisions based on the continuous stream of sensor inputs without relying on predefined maps or plans.

In the context of drones, autonomous navigation becomes particularly critical. Drones are versatile tools used in various applications, from surveillance and search and rescue to package delivery. Efficient and reliable navigation is essential for drones to operate seamlessly in dynamic and changing environments. The ability to navigate autonomously allows drones to respond to unforeseen obstacles, adjust their paths in real time, and optimize routes for efficiency. Traditional algorithms may struggle to keep up with the agility and adaptability required in real-world drone applications.

In response to these challenges, contemporary approaches aim to leverage data of demonstrated flight plans to train a policy to generate trajectories from visual input. Collecting data for training autonomous navigation models, especially for drones, can indeed be a challenging and time-consuming task. In real-world scenarios, obtaining sufficient data that covers diverse environments and possible situations requires significant expertise and resources. This is where simulation environments play a crucial role in generating the necessary data to train machine learning models for autonomous navigation.

However, a significant concern in training models in simulation is the "sim-to-real" gap, where the distribution of data seen in simulation differs from the distribution of data seen in the real world. Noise in real-world data can further contribute to this distributional mismatch. One approach to mitigate this challenge involves focusing on depth data rather than RGB data during the training process. Depth data provides information about the distance of objects in the environment, which is essential for drone navigation. Unlike RGB data, depth data is less sensitive to variations in lighting conditions and surface textures, making it more robust in different environments. By relying on depth information, the model becomes less dependent on the visual appearance of objects, allowing for improved generalization across diverse scenarios. In the simulation environment, generating realistic depth data involves simulating the noise and uncertainties associated with real-world depth sensors. One effective method is to render both left and right images, mimicking the stereo vision setup commonly used in drones. Simulating the noise in depth data can be achieved by applying algorithms like Semi-Global Matching (SGM) to calculate a disparity map from the rendered stereo images. This disparity map introduces realistic disparities and errors, simulating the imperfections found in real-world depth sensors. In conclusion, focusing on depth data rather than RGB data in the training process for autonomous navigation models, particularly for drones, offers a promising solution to mitigate the "sim-to-real" gap. Depth data provides crucial information about the spatial relationships and distances between objects in the environment, making it less sensitive to variations in lighting and surface textures compared to RGB data.

We plan to implement the paper Learning High-Speed Flight in the Wild [1]. They developed a deep learning model that can generate multiple collision-avoiding paths that can be fed into a drone

controller in real-time. They trained the model from depth data generated in simulation and the model was tested in real-world unseen environments and was able to successfully fly to a desired goal location at high speeds, 5m/s to 10m/s.

2 Related Work

In [3] a robust efficient motion planning is developed. They use a kinodynamic path-searching method to find a safe path optimized over time. The path searching is based on heuristic search using A* and linear quadratic minimum-time control. To smooth the trajectory and clearance of obstacles B-spline optimization is used, which is a method that adjust control points for desired properties. Lastly, an iterative time adjustment method is used to avoid non-conservative trajectories.

In [2], a navigation guidance based solution for tracking and avoiding non-cooperative agents or obstacles is proposed, where the authors extend the traditional Proportional Navigation Guidance (PNG) for collision avoidance of multiples obstacles. This approach does not need changing mode between pursuit and avoidance and does not rely on the optimization calculation.

3 Baseline Method

The paper Learning High-Speed Flight in the Wild [1], is the baseline method that we want to implement. In the paper, they achieve a policy that can autonomously fly quadrotors through various complex environments at high speeds.

The training data for this model is generated through simulation, where scenes are randomly initialized to represent various environments, including forests, urban settings, and fully random geometries. To generate depth data, the authors employ a stereo vision setup in simulation, capturing both left and right images from the drone’s perspective. Subsequently, the Semi-Global Matching (SGM) algorithm is applied to calculate a noisy depth map. This process mimics the real-world challenges faced by drones, where depth perception is critical for navigation. For trajectory generation, the authors utilize privileged information about the scene, including the locations of obstacles. They employ Metropolis-Hastings sampling to generate a base trajectory. This algorithm is given a specified heading to guide exploration in a particular direction. The base trajectory is then perturbed to create a diverse set of trajectories that represent the distribution of collision-free paths toward a goal. A cost is calculated for each trajectory based on how close the trajectory is to an obstacle. Importantly, during the perturbation process, trajectories that collide with obstacles or deviate from the desired direction are removed. This ensures that the final set of trajectories accurately represents feasible and goal-oriented paths for the drone to navigate within the given environment.

The model’s architecture involves a convolutional neural network (CNN) designed to map depth data into an embedding. This embedded representation is subsequently processed through multiple 1D convolutions, ultimately generating a proposed trajectory for the autonomous navigation task. Each proposed trajectory is accompanied by a predicted cost. In the training phase, the model utilizes a Relaxed Winner-Takes-All (R-WTA) loss for supervised learning, focusing on the three trajectories with the lowest cost identified by the expert. This loss function accommodates multi-hypotheses prediction by minimizing the difference between the expert trajectories and network trajectories based on their position components. To make this loss function differentiable, a small relaxation value is introduced, facilitating optimization through gradient descent. Additionally, collision cost predictions are trained through supervised learning, aligning them with ground-truth collision costs generated by the closest trajectory generated by the expert. The overall training loss for each sample is a combination of the R-WTA loss and the squared differences between predicted collision costs and ground-truth values.

Once the model is trained, selecting a singular trajectory from the multiple proposals the model generates is done by a set of heuristics. These heuristics aim to minimize the cost and maintain coherence with the drone’s current path. The chosen trajectory is determined by carefully considering these factors, ensuring a safe and consistent navigation strategy. The final trajectory selected is then fed to a model predictive controller (MPC) to fly the drone.

4 Proposal Method

Our plan is to implement that data generation pipeline and the model developed in the paper Learning High-Speed Flight in the Wild [1]. However, we want to also explore explaining the methods developed in that paper. Specifically, our areas of interest for exploration include introducing dynamic elements into scenes to assess their impact on the method’s performance. We also intend to investigate alternative applications for the heading value provided to the model, possibly directing it towards specific regions of interest within an image. Additionally, we plan to experiment with the model’s architecture, taking into account recent advancements in transformers and diffusion policies.

What we will specifically implement: the random scene creation; the stereo depth rendering with a block matching algorithm; the Metropolis-Hasting sampling trajectory generation in simulation with privileged data; the deep learning model and its loss in PyTorch; the control algorithm that flies a drone with the trajectory generated by the model, if there is time otherwise it will just be a point following the path; and a template matching algorithm to find regions of interest in the image to calculate a desired heading that is fed into the model.

References

- [1] Antonio Loquercio et al. “Learning high-speed flight in the wild”. In: *Science Robotics* 6.59 (2021), eabg5810.
- [2] Cristino de Souza, Pedro Castillo, and Boris Vidolov. “Reactive drone pursuit and obstacle avoidance based in parallel navigation”. In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2020, pp. 1–6.
- [3] Boyu Zhou et al. “Robust and efficient quadrotor trajectory generation for fast autonomous flight”. In: *IEEE Robotics and Automation Letters* 4.4 (2019), pp. 3529–3536.