QuetzalC++ Team - Results for Final Tier 1

L. Oyuki Rojas-Perez[1] and Jose Martinez-Carranza[1,2]

***Abstract*— ADR and Alpha pilot are international competitions inspired by the dron racer league. The aim of these competitions is to promote the development of artificial intelligence algorithms so that a drone runs through a complex track as quickly as possible. This has provided the scientific community with breakthroughs in environmental awareness, location, safe navigation, planning, as well as precise flight control to successfully execute the manoeuvres necessary to run a runway. Now, Game of Drones NeurIPS 2019 competition proposes to take a step closer to the drone racer league, to compete on the same runway against an adversary, and for it designs three levels of difficulty, 1) planning, 2) perception and 3) fully autonomous. We present our results of the participation on the first tier of teh competition using a trajectory planner.**

I. INTRODUCTION

Drone racing has become a popular sport due to it shows the human ability to control a vehicle in a highly dynamic environment using only visual information. In drone racing, each vehicle is controlled remotely by a pilot, who receives a live transmission from a camera onboard the vehicle. The vehicle must travel a complex track successfully as quickly as possible, and for this, the pilot requires years of training. This sport has motivated autonomous drone racing competitions such as "IROS - Autonomous Drone Racing" [1], "AlphaPilot Innovation Challenge" [2] and recently "Game of Drones" a NeurIPS 2019 competition [3].

Autonomous drone races capture some of the central problems highlighted in robotics and artificial intelligence. Since the development of a fully autonomous drone is difficult due to the challenges that include dynamic modelling, visual perception, trajectory generation, optimal control, location and mapping. Also, the processing is required to be on board using embedded systems in order to avoid the loss of information caused by communication interference. Multiple studies have focused directly on designing a strategy that will allow flying an entire trajectory, applying classic methods based on state machines, visual control and location [4]. One of the problems they encounter in the application of classic methods for drone racing is the failures that are susceptible to changes in appearance, such as variable lighting and inconsistent computational overload.

Deep Learning networks is one of the most used tools today. Recent work has shown that deep networks can allow the development of drones that can detect the doors of the circuit with greater robustness due to lighting changes and even to overlap between doors [5], [6], [7], [8], [9]. However, these techniques are not covered scenarios where other dynamic objects could appear; for example, other drones sailing on the same circuit.

Currently, in autonomous drone racing, the vehicle flies through orange square doors in a predefined sequence using onboard processing. The vehicle that flies through more doors wins the race, and if two drones arrive at the same door or complete the track, the fastest time defines the winner. In these races, several aspects of the real world of drone racing are ignored such as performing the circuit against an adversary, since it is difficult to predict the manoeuvres or movements that the vehicle will perform. For this reason Game of Drones offered a drone race against an opponent on the same track.

Game of Drones present three tiers, 1) planning, 2) perception and 3) fully autonomy, combined planning and perception. In this report, we show the solution proposed for the first tier, using planning to complete the tack. To present our proposal, the document has been organised as follows: the section II describes the related work; section III describes the first trial of Game of Drones competition; section III-B.2 describes the trajectory implement for solution; IV describes our experiments; Finally, our conclusions are discussed in the section V.

II. RELATED WORK

Drone racing is a popular sport; in them, aerial vehicles are remotely controlled to travel a complex track. The pilot shows his ability to control a highly dynamic vehicle using only visual information transmits the drone to FPV (First Person View) glasses. This led to the world's first autonomous drone racing competition, held at the International Conference on Intelligent Robots and Systems IROS 2016 in Korea, Japan [4].

The problem of visual navigation was attacked in the first edition of the autonomous drone racing. Participants developed algorithms based on visual-based navigation to traverse a complex circuit of orange gates. To complete the circuit, some works used RGB cameras to identify the gates they had to cross, and for that, they used algorithms for colour segmentation and corner detection, the controller was based on a reactive control approach [10]. On the other hand, some authors used visual information combined with a location system based on LIDAR, allowing the drone to plan their flight route [5], [11]. However, the door detection method was not efficient, since detection based on colour segmentation is sensitive to lighting conditions. Therefore, the algorithm parameters must be manually adjusted before

[1] Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Puebla, México. Department of Computer Science at INAOE. Email addresses: {oyukirojas, carranza}@inaoep.mx

[2]University of Bristol, Bristol, UK.

each execution, and even with the correct parameters, the drone is easily confused when two doors overlap.

Many works use different strategies to solve the drone racing in a real world, for example, in the 2017 edition of the Autonomous Drone Racing [12], participants used waypoint tracking using depth sensors or using a metric monocular SLAM system [13]. Others implemented state machines combined with position and speed estimation. Another solution for this competition was to replace the door detection algorithms based on computer vision algorithms with deep learning. This tool improves the accuracy of the gate detection and is not sensitive to lighting. It is also able to differentiate the front doors when overlap occurs [7].

In other works, they not only use deep learning for door detection and classification. Kaufmann et al. [6], [8], provide robust perception, the proposed deep network detects and obtains the direction of the gate. The prediction speed is incorporate in an extended Kalman filter to the gate location. These two parameters are used with a predictive control algorithm to correct the flight path, as a result significantly improves the speed at the one that the drone goes through a circuit. Cocoma et al. [14] propose the use of the pose-net network to calculate the position of the drone respect to the gate. The output of the network provides a vector with the values of x, y and z in meters.

On the other hand, two works contemplate the presence of an adversary using a game theory [15], [16]. In which they define a series of strategies to win the race, one of them is to hinder the passage of the drone from behind. To carry out these strategies, they use markers to identify the adversary and external location using a motion capture system, so the drones know their and the adversary position, and thus decide the best option to evade or maintain their position. This work can be useful tools for the Game of Drones competition since the simulated Airsim environment allows us to know the drone and the adversary position. However, for the first tier of the Game of Drones competition, we implement a trajectory planning to complete the tack crossing all gates without collision as fast as possible.

III. Game of Drones NeurIPS 2019 Competition

A. Airsim

AirSim [17] is a simulator that offers more realistic scenarios with a dynamic model of the drone closer to reality. Agile and aggressive flight can be simulated in this environment as much as flight missions with adversaries and other entities moving in the environment. Also, Airsim allows to change the weather intensity and the road conditions, for example, rain, road wetness, snow, road snow, falling leaves, road leaves, dust and fog.

B. Qualifier Tier 1

The objective of the first tier of the Game of Drones competition is Finish the track quicker than your opponent, without crashing. The track has 20 gates on different heights and orientations about the z-axis. The gates are rectangular and marked using a "race-tape" checker pattern with different colours on each side, the front side is green, the inner side is blue and the back side is red. For the tier 1 all gates are the same dimensions. There are two reasons for disqualification, timeout if a drone does not finish within the maximal lap time of a track (100 sec) and multiple drone-drone collisions.

1) Data Acquisition: For data acquisition, the organisers provide a API called "airsimneurips" compatible with Python. The airsimneurips API provides us with the following data:

- Drone position.
- Telemetry of the drone.
- Adversary position
- Gate position

2) Trajectory Planner: MoveOnSplineAsync uses ETHZ-ASL's mav_trajectory_generation [18] as the trajectory planning back-end, tracks the references positions and velocities using a pure pursuit tracking controller. Finally the yaw reference is allocated along the tangent of the trajectory. Hence the drone will always look at the direction along which it is flying. MoveOnSplineAsync needs the following parameters:

- Trajectory Tracker Gains (cross track, vel cross track, along track, vel along track, z track, vel z, yaw)
- A list of 3D waypoints.
- Maximum speed.
- Maximum acceleration.
- Position constraint.
- Velocity constraint.
- Acceleration constraint.
- Visualisation of trajectory.
- Colour of trajectory.
- Vehicle name.

IV. Experiments and Results

To complete the task, we use the "moveOnSplineAsync", it is a trajectory planner provided from the airsimneurips API.We modify the waypoints that make up the trajectory to improve drone performance. First, we change the second reference point to the interpolation between Gate 1 and 3 while maintaining the height of Gate 1, since Gates 1,2,3 and 4 are in a straight line. We also added a waypoint after Gate 12 to correctly cross Gate 13. Finally, we added the following waypoints to keep the speed of the drone constant: 1) Between Gate 8 and 9; 2) Between Gate 13 and 14.

Additional, we use different maximum speed and acceleration for Gates 1 to 19 and Gate 20 to 21. We focus on to cross all the gates without time penalties as fast as possible. The table IV shows the progress reducing the time lap.

To reduce the travel time, even more, we make the following modifications to the trajectory: 1) We moved the waypoint of the first Gate 10 meters before; 2) We move the waypoint of the Gate 4 40 cm to the front; 3) We move the waypoint of the Gate 5 40 cm back; 4) We move the waypoint of the Gate 6 25 cm to the front; 5) We move the waypoint of the Gate 8 and 9 50 cm down; 6) We added a waypoint between gate 9 and 10 to improve the rise of the drone

Max. Speed Gates 1 to 19	Max. Acceleration Gates 1 to 19	Max. Speed Gates 20 to 21	Max. Acceleration Gates 20 to 21	Crossed Gates	Avg.Time Lap (Seconds)	Penalty (Seconds)
85	26	25	20	21/21	81.755	+6
85	26	25	20	21/21	81.050	0
85	26	27	21	21/21	77.51	0
85	26	27	21	21/21	77.52	0
86	26	27	21	21/21	77.393	+6
89	27	27	21	21/21	74.520	+3
90	27	27	21	21/21	74.348	+3
106	28.7	27	21	21/21	72.816	0
106	28.7	27	21	21/21	72.868	0
106	**28.7**	**27**	**21**	**21/21**	**72.858**	**0**

TABLE I

PROGRESS IS REPORTED IN THE TABLE BY REDUCING THE AVERAGE LAP TIME. TEN RUNS WERE PERFORMED FOR EACH PARAMETER MODIFICATION.

Max. Speed Gates 1 to 19	Max. Acceleration Gates 1 to 19	Max. Speed Gates 20 to 21	Max. Acceleration Gates 20 to 21	Crossed Gates	Avg.Time Lap (Seconds)	Penalty (Seconds)
95	82	90	80	21/21	60.689	0
95	84	90	80	21/21	60.634	0
97	85	90	80	21/21	60.521	0
120	98	90	80	21/21	60.466	0
130	37	200	128	21/21	56.605	0
135	45.8	490	157	21/21	54.051	0
135	**46**	**495**	**160**	**21/21**	**53.565**	0

TABLE II

PROGRESS IS REPORTED IN THE TABLE BY REDUCING THE AVERAGE LAP TIME. TEN RUNS WERE PERFORMED FOR EACH WAYPOINT MODIFICATION.

and avoid it to collide with the floor. Finally, a waypoint was added on the Gate 12 so that the drone would cross correctly the Gate 13. With these modifications, we managed to finish the **21 Gates** circuit in **53.565** seconds. The figure IV shows parts of the course completed in 53.565 seconds our best result for Tier 1. The table IV shows the progress reducing the time lap. The displacement of the waypoints was calculated to test and error depending on the speed and maximum acceleration.

V. CONCLUSIONS

Game of Drones competition presents three tiers of different difficulty and level of autonomy. The goal is to finish the race quicker than the opponent. We focus on the first tier complete the complex track against an adversarial as fast as possible. We present the results obtained using the trajectory planner MoveOnSplineAsync. Our strategy was to modify the waypoints of the original trajectory to improve the performance of the drone during flight. The best time reported is 53.565 seconds crossing 21 gates without time penalty.

REFERENCES

[1] Iros 2018 autonomous drone racing competition. *2018*, [Online], Available: https://www.iros2018.org/competitions.

[2] Lockheed martin and drone racing league launch groundbreaking ai innovation challenge. *2018*, [Online], Available: https://news.lockheedmartin.com/2018-09-05-Lockheed-Martin-and-Drone-Racing-League-Launch-Groundbreaking-AI-Innovation-Challenge.

[3] Game of drones neurips 2019 competition. *2019*, [Online], Available: https://www.microsoft.com/en-us/research/academic-program/game-of-drones-competition-at-neurips-2019/.

[4] H. Moon, Y. Sun, J. Baltes, and S. J. Kim. The iros 2016 competitions [competitions]. *IEEE Robotics Automation Magazine*, 24(1):20–29, March 2017.

[5] Sunggoo Jung, Sungwook Cho, Dasol Lee, Hanseob Lee, and David Hyunchul Shim. A direct visual servoing-based framework for the 2016 iros autonomous drone racing challenge. *Journal of Field Robotics*, 35(1):146–166, 2018.

[6] Elia Kaufmann, Antonio Loquercio, Rene Ranftl, Alexey Dosovitskiy, Vladlen Koltun, and Davide Scaramuzza. Deep drone racing: Learning agile flight in dynamic environments. *arXiv preprint arXiv:1806.08548*, 2018.

[7] Sunggoo Jung, Sunyou Hwang, Heemin Shin, and David Hyunchul Shim. Perception, guidance, and navigation for indoor autonomous drone racing using deep learning. *IEEE Robotics and Automation Letters*, 3(3):2539–2544, 2018.

[8] Elia Kaufmann, Mathias Gehrig, Philipp Foehn, René Ranftl, Alexey Dosovitskiy, Vladlen Koltun, and Davide Scaramuzza. Beauty and the beast: Optimal methods meet learning for drone racing. *arXiv preprint arXiv:1810.06224*, 2018.

[9] Hyungpil Moon, Jose Martinez-Carranza, Titus Cieslewski, Matthias Faessler, Davide Falanga, Alessandro Simovic, Davide Scaramuzza, Shuo Li, Michael Ozo, Christophe De Wagter, et al. Challenges and implemented technologies used in autonomous drone racing. *Intelligent Service Robotics*, 12(2):137–148, 2019.

[10] Shuo Li, Michaël MOI Ozo, Christophe De Wagter, and Guido CHE de Croon. Autonomous drone race: A computationally efficient vision-based navigation and control strategy. *arXiv preprint arXiv:1809.05958*, 2018.

[11] Shuo Li, Erik van der Horst, Philipp Duernay, Christophe De Wagter, and Guido CHE de Croon. Visual model-predictive localization for computationally efficient autonomous racing of a 72-gram drone. *arXiv preprint arXiv:1905.10110*, 2019.

[12] Hyungpil Moon, Jose Martinez-Carranza, Titus Cieslewski, Matthias Faessler, Davide Falanga, Alessandro Simovic, Davide Scaramuzza, Shuo Li, Michael Ozo, Christophe De Wagter, Guido de Croon, Sunyou Hwang, Sunggoo Jung, Hyunchul Shim, Haeryang Kim, Minhyuk Park, Tsz-Chiu Au, and Si Jung Kim. Challenges and implemented technologies used in autonomous drone racing. *Intelligent Service Robotics*, 12(2):137–148, Apr 2019.

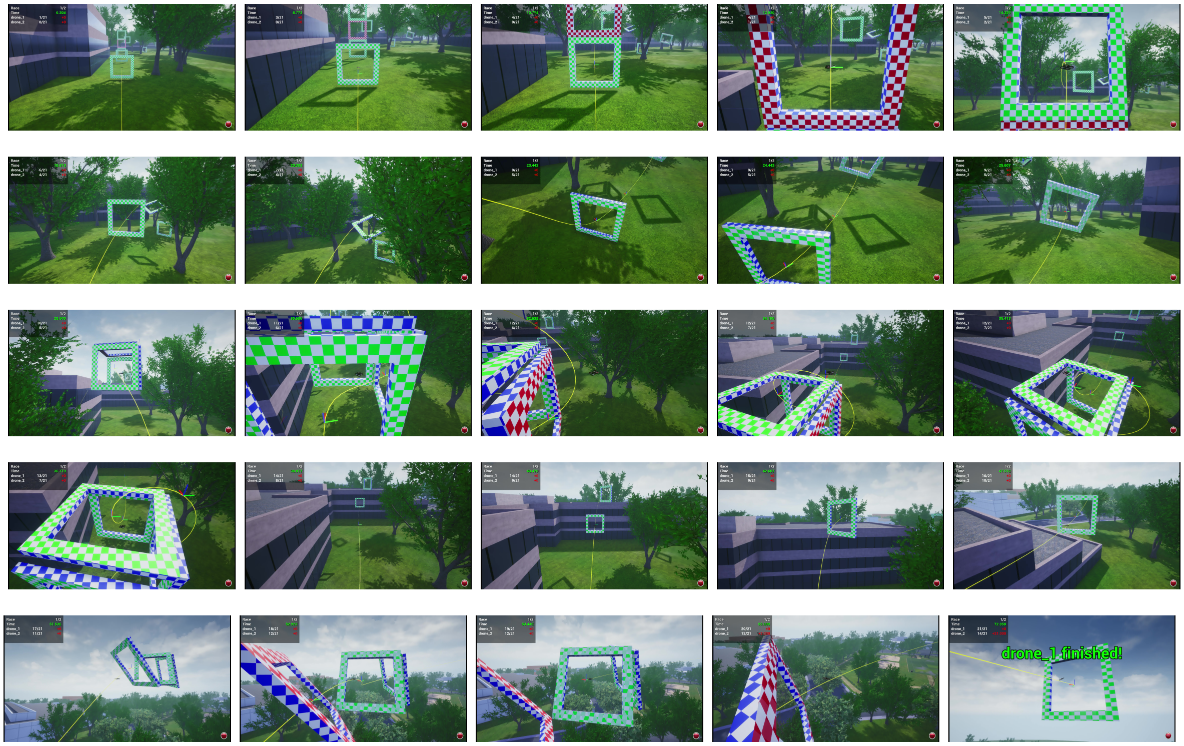

Fig. 1. Examples of the best run. The speed and acceleration used for Gate 1 to 19 is 106 m/s and 28.7 m/s respectively, and for Gate 20 to 21 it is 27 m/s and 21 m/s.

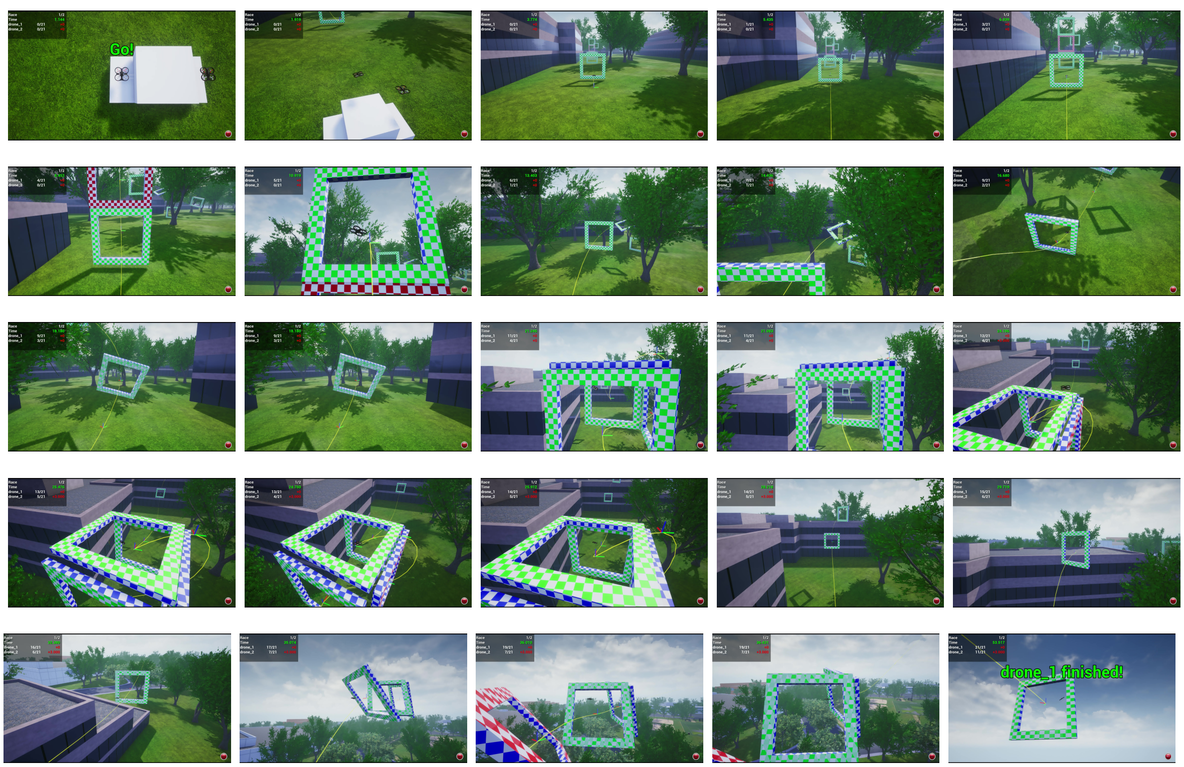

Fig. 2. Examples of the best run. The speed and acceleration used for Gate 1 to 19 is 135 m/s and 46 m/s respectively, and for Gate 20 to 21 it is 495 m/s and 160 m/s.

[13] L. O. Rojas-Perez and J. Martinez-Carranza. Metric monocular slam and colour segmentation for multiple obstacle avoidance in autonomous flight. In *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, pages 234–239, Oct 2017.

[14] José Arturo Cocoma-Ortega and José Martinez-Carranza. A cnn based

drone localisation approach for autonomous drone racing. In *11th International Micro Air Vehicle Competition and Conference*, Madrid, Spain, October 2019.

[15] Zijian Wang, Riccardo Spica, and Mac Schwager. Game theoretic motion planning for multi-robot racing. In *Distributed Autonomous Robotic Systems*, pages 225–238. Springer, 2019.

[16] Riccardo Spica, Davide Falanga, Eric Cristofalo, Eduardo Montijano, Davide Scaramuzza, and Mac Schwager. A real-time game theoretic planner for autonomous two-player drone racing. *arXiv preprint arXiv:1801.02302*, 2018.

[17] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.

[18] Charles Richter, Adam Bry, and Nicholas Roy. Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In *Robotics Research*, pages 649–666. Springer, 2016.