Autonomous Cinematography with Teams of Drones

Jesus Capitan[1] and Arturo Torres-Gonzalez[1] and Anibal Ollero[1]

I. INTRODUCTION

Aerial cinematography with drones is becoming quite popular, with many new commercial platforms for both amateurs and professionals. The main reasons for this breakout are their reduced cost, their maneuverability and their ability to produce unique shots, mainly when compared to static cameras or dollies. Moreover, using multiple drones to film the same event concurrently widen the range of possibilities available for the production team, specially in outdoor, large-scale scenarios. For instance, several action points could be covered simultaneously, or multi-camera shots could be carried out, achieving more artistic views.

However, there are multiple issues to consider when operating an multi-drone system for cinematography, which increases considerable the operator's load. To name a few, moving targets have to be tracked, smooth trajectories are needed for the drones, different shots should be assigned to the available drones taking into account their remaining battery, collisions between the drones should be prevented, drones should not access no-fly zones, etc.

There are multiple works addressing camera motion planning for aerial filming [1], [2]. The common idea is to formulate some kind of optimization problem to generate smooth camera trajectories that fulfill aesthetic and cinematographic constraints. There are also end-to-end solutions for aerial cinematographers [3], [4] where high-level commands can be specified. However, the focus of these previous works is on static scenes and single-drone settings. Moreover, there are works filming dynamic targets in outdoor scenarios and coping with obstacle avoidance [5], [6]. There is no much work considering multi-drone shots for cinematography. Recently, some authors have proposed optimization techniques based on receding horizon to film with several drones in indoor settings [7], [8].

In this work, we propose a distributed system for autonomous execution of cinematography missions. We implement a set of flight controllers in order to perform with autonomous drones canonical shots described in the literature. Then, we devise a system that allows the drones to execute concurrent shots in a distributed manner, by means of synchronization events. Our work has been done within the framework of the EU-funded project MULTIDRONE [1], which addresses the problem of building drone teams for autonomous media production. The project studies all aspects in the complete system: how to define *shooting* missions for media production; how to translate them into feasible plans for the drone team, and how to execute those planned missions with multiple drones in parallel.

Our system assumes that there is a central planner that receives shooting missions with shot descriptions by a media director; and it allocates those shots to the available drones so that constraints such as shots' start time and duration, as well as drones' remaining battery, are fulfilled. This paper focuses on the autonomous execution of these shooting missions with the drone team, once the plan has been computed and the shots assigned to the drones. In particular, we propose a distributed scheduler that activates onboard each drone the corresponding shot controllers. An event-based system is used to synchronize shot execution among the drones and ensure proper coordination. Our main contributions are the following:

- We compile a list of canonical shots from the cinematography literature. Then, we implement autonomous flight controllers for all the shots, defining specific parameters to describe each shot. Each controller includes also a component to control the camera gimbal tracking the filmed target.
- We develop a complete architecture for autonomous execution of cinematography missions with a team of drones. The architecture is agnostic to the planner used to assign the shots in a feasible manner, and it integrates shot controllers with onboard shot schedulers and target tracking modules.

Our approach uses an event-based mechanism to synchronize multi-drone shots and to ensure that shots start at the right time. The original plan for shot execution is computed predicting targets' motion. However, there may be inaccuracies on those predicted trajectories (e.g., a target arrives later than expected to shot start position) or contingencies during the mission execution (e.g., a drone fails and runs out of battery). Our method tackles those uncertainties in two manners. First, re-planning is allowed online during mission execution if any emergency occurs. Second, the synchronization mechanism allows us to account for possible delays in the mission execution or mismatches with the original plan. Since drones start shots when they receive the corresponding event, we could plan so that they arrive at the shot start position before the expected time, and use that buffer time to accommodate possible delays in the scene.

Besides, our method considers safety in several manners: (i) drones are not allowed to fly over pre-defined no-fly zones; (ii) a collision avoidance algorithm is used so that drones do not collide with each other; and (iii) there is an

[1]J.C., A.T. and A.O. are with the University of Seville, Seville, Spain [jcapitan,arturotorres,aollero]@us.es

[1]https://multidrone.eu

onboard component for emergency management that triggers specific procedures in case of failures.

II. SYSTEM OVERVIEW

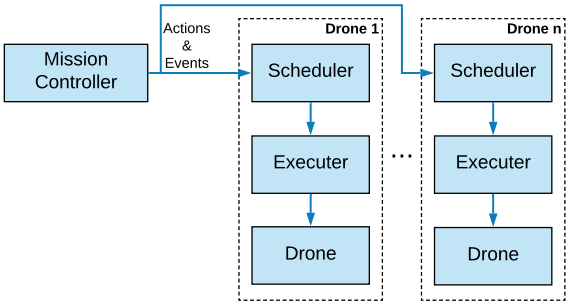

Fig. 1. Overview of our system architecture.

Figure 1 shows the architecture of our system. There is a central entity called *Mission Controller*, which is the one managing the whole planning and execution process for a mission. This module receives from the end-user (i.e., media director) the shooting missions, with descriptions of the desired shots, i.e., their starting events and positions, their duration, etc. For instance, a director could design a shooting mission to film a rowing race; and specify a *lateral* shot from the `START_RACE` event to the end of the race, and a *orbital* shot starting with the `FINISH_LINE` event, i.e., when the boats reach the finish line. The Mission Controller will send these events to the drones as they occur. Depending on the event, this occurrence may be detected automatically by the Mission Controller or indicated by the director.

When the Mission Controller receives a shooting mission, it uses another centralized module implementing the *Planner* to compute a feasible plan to execute. This module is out of the scope of this paper and would produce a list of *actions* for each drone. Basically, *shooting* and *navigation* actions; the former to execute shots, the latter to navigate from the end of a shot to the start position of the next one. Once the plan is computed, the Mission Controller sends it to the drones, and it is supposed to be executed by the onboard *Schedulers* in a distributed manner. Each Scheduler listens to events coming from the Mission Controller and starts or stops the execution of the current shooting action when the corresponding event happens. This is done by calling another module called *Executer*, which is the one actually implementing the flight and gimbal controllers to carry out the shots. Synchronization for multi-camera shots is achieved because all involved drones will be waiting for the same event to start. Moreover, the Executer can also be called to execute navigation actions.

The Schedulers report back to the Mission Controller the status of the mission execution, i.e., which action is the drone executing or waiting for. In case of an emergency in the drone, e.g. low battery or loss of GPS, the Scheduler commands the drone a safe landing and reports back an emergency status. Then, the Mission Controller would trigger a re-planning procedure calling again the Planner with the available drones and remaining shots to execute. Each drone will finish with the ongoing action, and will append behind the new list of actions to execute.

Fig. 2. Drone cinematographer during our outdoor trials.

III. EXPERIMENTAL RESULTS

We tested our system with example shooting missions in simulation. For that, we use the GAZEBO [9] simulator and the PX4 [10] SITL (Software In The Loop) functionality to simulate the autopilot. The simulator also uses UAL (UAV Abstraction Layer) [11], an open-source library to interact with autonomous drones, abstracting the user from the specifics of the platform used. With this simulator, it is easy to run and test different examples without flying with the real drones but using the very same software. In particular, we simulated different shooting missions to demonstrate the integration of the whole system and evaluate the types of views obtained by the drone cameras. A video with an example simulation can be seen at: `https://youtu.be/qRPXTid9dFI`. In the example, there is a car moving along a straight road that has to be filmed. A drone performs a lateral shot followed by a static shot, whereas another two drones perform an orbital shot together. The mission starts taking off all drones after a `GET_READY` event. Then, the go to their corresponding starting waypoints and wait there for a `START_RACE` event that triggers the shooting actions. After 20 seconds, the drones come back to the home position and land.

Currently, we are running tests in outdoor, mock-up scenarios in order to test functionalities of the system, but results are still preliminary. Figure 2 shows one of the prototypes that we built filming a moving target with a mounted GPS.

ACKNOWLEDGMENT

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731667 (MULTIDRONE). This publication reflects the authors' views only. The European Commission is not responsible for any use that may be made of the information it contains.

References

[1] N. Joubert, M. Roberts, A. Truong, F. Berthouzoz, and P. Hanrahan, "An interactive tool for designing quadrotor camera shots," *ACM Transactions on Graphics*, vol. 34, no. 6, pp. 1–11, oct 2015.

[2] Q. Galvane, J. Fleureau, F.-L. Tariolle, and P. Guillotel, "Automated Cinematography with Unmanned Aerial Vehicles," *Eurographics Workshop on Intelligent Cinematography and Editing*, 2016.

[3] N. Joubert, J. L. E, D. B. Goldman, F. Berthouzoz, M. Roberts, J. A. Landay, and P. Hanrahan, "Towards a Drone Cinematographer: Guiding Quadrotor Cameras using Visual Composition Principles," *ArXiv e-prints*, 2016.

[4] C. Gebhardt, B. Hepp, T. Nägeli, S. Stevšić, and O. Hilliges, "Airways: Optimization-Based Planning of Quadrotor Trajectories according to High-Level User Goals," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, 2016, pp. 2508–2519.

[5] C. Huang, F. Gao, J. Pan, Z. Yang, W. Qiu, P. Chen, X. Yang, S. Shen, and K. T. Cheng, "Act: An autonomous drone cinematography system for action scenes," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 7039–7046.

[6] R. Bonatti, C. Ho, W. Wang, S. Choudhury, and S. Scherer, "Towards a Robust Aerial Cinematography Platform: Localizing and Tracking Moving Targets in Unstructured Environments," *arXiv e-prints*, p. arXiv:1904.02319, Apr 2019.

[7] T. Nägeli, L. Meier, A. Domahidi, J. Alonso-Mora, and O. Hilliges, "Real-time planning for automated multi-view drone cinematography," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–10, jul 2017.

[8] M. Saska, V. Krátký, V. Spurný, and T. Báča, "Documentation of dark areas of large historical buildings by a formation of unmanned aerial vehicles using model predictive control," in *22nd IEEE Intl. Conf. on Emerging Technologies and Factory Automation (ETFA)*, Sep 2017, pp. 1–8.

[9] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004, pp. 2149–2154.

[10] L. Meier, T. Gubler, J. Oes, D. Sidrane, D. Agar, B. Kng, A. Babushkin, px4dev, M. Charlebois, R. Bapst, and et al., "Px4/firmware: v1.7.3 stable release," Jan 2018.

[11] F. Real, A. Torres-González, P. R. Soria, J. Capitán, and A. Ollero, "Ual: an abstraction layer for unmanned vehicles," in *2nd International Symposium on Aerial Robotics (ISAR)*, 2018.