A Multidrone Approach for Autonomous Cinematography Planning

Arturo Torres-González[1], Jesús Capitán[1], Rita Cunha[2], Aníbal Ollero[1], Ioannis Mademlis[3]

[1] Robotics, Vision and Control Group
University of Seville, Seville, Spain
[arturotorres,jcapitan,aollero]@us.es
[2] Institute for Robotics and Systems in Engineering and Science (LARSyS)
Instituto Superior Técnico, Lisbon, Portugal
rita@isr.ist.utl.pt
[3] Department of Informatics
Aristotle University of Thessaloniki, Thessaloniki, Greece
imademlis@aiia.csd.auth.gr

Abstract. This paper proposes a multidrone approach for autonomous cinematography planning. The use of drones for aerial cinematography is becoming a trend. Therefore, the aerial cinematography opens a new field of application for autonomous platforms that need to develop intelligent capabilities. This becomes even more challenging if a team of multiple drones are considered for cooperation. This paper introduces the novel application of planning for cinematography, including the challenges involved and the current state of the art. Then, it proposes a first version of an architecture for cooperative planning in cinematography applications, like filming sport events outdoors. The main features for this architecture are the following. The system should be able to reproduce typical shots from cinematography rules autonomously, shooting static and mobile targets. It should also ensure smooth transitions along the shots, implementing collision avoidance and being aware of no-fly zones, security and emergency situations. Finally, it should take into account the limited resources of the drones (e.g. battery life).

Keywords: Aerial cinematography; multi-robot planning; cooperative robots

1 Introduction

The use of drones for cinematography and audio-visual applications is becoming quite trendy. First, small drones are not expensive and can be equipped with high-quality cameras that are available in the market for amateur and professional users. Second, they are able to produce unique video streams thanks to their maneuverability and their advantageous view points when flying.

The process of getting video footage with aerial vehicles in an autonomous manner is challenging though. It imposes a series of difficulties related with robot positioning and navigation, smooth control of the camera, collision avoidance, etc., mainly when the application is outdoors.

A novel idea is to create a team of autonomous cameras for filming cooperatively outdoor events. When gathering a team of Unmanned Aerial Vehicles (UAVs) with cameras onboard, additional challenges arise. It is true that different targets could be shot at the same time, or even the same one from several points of view, enabling the composition of novel artistic shots. However, planning and scheduling is a relevant issue, given the uncertainties involved in the events to film, the sensing capabilities of the UAVs themselves, the communication infrastructure, etc. The UAVs must be able to plan and follow smooth trajectories without colliding and cooperate among themselves, ideally even taking into account their limited resources (e.g., battery life).

This paper proposes a multidrone approach for autonomous cinematography planning. This work lies within the framework of the MultiDrone European project[4], which is one of the first attempts to produce an intelligent team with multiple UAVs for media production. The system developed will be tested for filming sport events outdoors, such as football games, cycling or boat races.

The MultiDrone project started recently in January 2017 and it is still at a design phase. Therefore, the primary objective of this paper is to present a first version of the architecture for cooperative planning in the project, describing the different modules involved and their interconnections. A review of the state of the art for autonomous aerial cinematography is also included, providing a good overview of the challenges for this novel application.

The main features for our architecture are the following. The system should be able to reproduce typical shots from cinematography rules autonomously, shooting static and mobile targets. It should also ensure smooth transitions along the shots, implementing collision avoidance and being aware of no-fly zones, security and emergency situations. Finally, it should take into account the limited resources of the drones (e.g. battery life).

This paper is organized as follows: Section 2 describes related work for autonomous cinematography; Section 3 introduces the typical shots for aerial cinematography; Section 4 presents the overall architecture for planning; Section 5 describes more in detail the different components; and Section 6 gives conclusions and next steps for future work.

2 Related Work

Cinematography is experiencing a remarkable upgrade due to the appearance of drones. Given their maneuverability and the fact that they can be equipped with high-quality cameras in a not very costly way, they are taking a relevant role as aerial cinematographers. On the one hand, they provide new and interesting features for cinematography directors, since they can access difficult places and perform acrobatic trajectories. On the other hand, they also impose constraints due to their dynamics and must follow cinematography rules.

[4] https://www.multidrone.eu

An interesting idea is to have aerial systems that can autonomously help the director when shooting, eliminating the need for duplicated operators, controlling the drone and the camera. For this purpose, a director could specify a set of desired shots; and the drone should be able to fly there and place the camera at the required position. Moreover, transitions between shots should fulfill visual composition principles, so that the drone produces camera trajectories that are, not only safe and feasible, but also visually pleasing.

In this application, several communities need to be brought together: (i) cinematographers can provide insight in canonical shot types for aerial cameras; (ii) computer graphics researchers have studied algorithms to capture visually pleasing footage with virtual cameras; and (iii) roboticists have developed multiple algorithms for aerial path planning.

In the literature, there is some recent work fusing the previous ideas to produce semi-autonomous aerial cinematographers [8, 7]. In these works, the user or director specifies high-level commands such as shot types and positions, and the drone is in charge of the navigation control. In [8], an outdoor application to film people is proposed, and different types of shots from the cinematography literature introduced (e.g., close-up, external, over-the-shoulder, etc). Timing for the shots is considered by means of an easing curve that drives the drone along the planned trajectory (i.e., the curve can modify the velocity profile). In [7], an iterative quadratic optimization problem is formulated to get camera and target smooth trajectories. No time constraints nor moving targets are included.

Path planning for aerial vehicles is also a well-known topic in robotics. For instance, many works [14, 11, 1] present algorithms to design safe trajectories for unmanned aerial vehicles. They deal with collision avoidance, dynamic constraints and control issues to execute the planned trajectories.

As mentioned above, designing virtual camera trajectories for pleasing footage is a classic problem in computer graphics. For instance, [6, 10] and the references therein present different works on pleasant interpolation between two camera poses. Furthermore, it has been proven that the trajectories must be $\mathcal{C}^4$ continuous in order to satisfy physical equations from quadrotors [9]. Note that some previous works [14, 11] hold with this constraint and deal with obstacle avoidance at the same time.

The approaches above focus mainly on static shots, i.e., when the drone does not track a moving target. There are also numerous works on systems for visual tracking of moving targets with aerial vehicles, although they are not usually thought to comply with cinematography rules. In particular, some propose tracking controllers based on classic PIDs [17]; while others use alternative techniques for position control such as LQR [13]. An interesting work more related to cinematography is also presented in [4], where a discrete POMDP is used to take frontal shots from a moving target. The POMDP selects between two actions: staying or moving to a new goal location (facing the target). The idea is to estimate target's intentions (changing location/orientation or staying) and minimize camera movements accordingly. In [3], the video input from a robotic camera, which tracks actively a pre-defined region-of-interest (e.g., centroids of players in a sport event), is processed in real time to produce a virtual camera video output following a smooth, aesthetically pleasing trajectory.

In the literature, there is no much work on extending the previous concepts to multi-camera systems. Indeed, the fact of using a team of aerial cameras for cinematography may bring new artistic rules and novel kinds of shots to be explored. The work in [15] is somehow related, since they propose a method to place as few drones as possible to cover all the available targets in the scenario without occlusion. However, this is done in a 2D space and considering that cameras must be always facing the targets. Moreover, smooth transitions are not considered, as they only reason about computing the shooting points. There is also a recent work on several UAVs tracking the same moving target [12]. The approach uses a fixed gimbal camera mounted on the UAV, which is controlled through errors on image coordinates. UAV replacement is considered when its battery is running out. In that case, the first UAV does not land until the second one has the target also within its field of view. In [2], an optimization-based algorithm is presented for the computation of a single, aesthetically pleasing video, conforming to basic cinematographic guidelines (such as the 180-degree rule and jump cut avoidance), from raw feeds coming separately from multiple cameras. Operating also within a multi-camera context, automated editing can be considered as a problem of camera selection over time [5].

3 Canonical Shot Types

In the literature, a lot can also be found about cinematographic rules and canonical types of shots. A complete guide on how to use drones for photography and video shooting can be found in [16]. In order to understand better the problem pre-requisites and the planning architecture proposed, this section provides a summary of the most typical aerial shots that can be found in cinematography.

First, shots can be aimed at filming moving targets or static targets/scenes. Besides, different kinds of movement patterns can be applied to the camera while taking the shot.

Static shot: The drone is hovering and the camera fixed or surveying a static scene (top-down or side-to-side).

Still shot: The drone is hovering and the camera tracking a moving target.

Lateral tracking shot: The drone flies sideways/in parallel to the target, matching its speed if possible, while the camera remains always focused on the moving target. The camera axis is approximately perpendicular to the target trajectory and parallel to the ground plane.

Vertical tracking shot: The drone flies exactly above the target, matching its speed if possible. The camera remains always focused, vertically down, on the moving target. The camera axis is perpendicular to the target trajectory.

Pedestal/elevator shot: The drone is slowly flying up or down, along the z-axis, with constant velocity. The camera rotates slowly (mainly along the pitch axis), in order to always keep the linearly moving or static target properly framed. The projections of the camera axis and the target trajectory on the ground plane remain approximately parallel during shooting.

Reveal shot: The drone flies at a steady trajectory with constant velocity and the camera in a fixed position. The target is initially out of frame (e.g., hidden behind an obstacle) until the movement of the UAV reveals its position.

Orbital shot: The drone circles around the target, following the target's trajectory (if it is moving). The camera is slowly rotating in order to always keep the still or moving target properly framed. During shooting, the difference in altitude between the drone and the target remains constant.

Fly-over shot: The drone follows/intercepts the target from behind/from the front, flying parallel to its trajectory and with constant velocity, until passing over the target. The camera is slowly rotating (mainly along the pitch axis), in order to always keep the still or moving target properly framed. Once the target is passed, the drone keeps flying along the same trajectory for some time, with the camera still focusing on the receding target.

Fly-by shot: The drone follows/intercepts the target from behind/from the front and to the left/right, passes it by and keeps on flying at a linear trajectory with steady altitude. The drone and target trajectory projections onto the ground plane remain approximately parallel during shooting. The camera moves to always keep the still or linearly moving target properly framed.

Chase/follow shot: The drone follows the target from behind/from the front, at a steady trajectory, steady distance and matching its speed if possible. The camera remains always focused on the target.

4 Overall Planning Architecture

In this section, the overall approach for planning and re-planning that will be implemented in MultiDrone is described. In this architecture, planning can take place at different phases and with different modules. For instance, computing a safe path to a landing spot or to a specific shooting position can be considered planning, but distributing different shooting tasks among the team members and coordinating them is also planning. Moreover, planning will also be an online functionality. Given the commands from the director, an initial plan can be computed. However, during the execution of the plan, the original circumstances may vary, making that initial plan no longer useful. Imagine for instance that some parts of the plan were not successfully accomplished (uncertainties in UAVs' actions or targets movements), that there were new commands from the director, or that unexpected emergency events happened. All those situations would trigger a new planning phase at the higher level.

In the system there will exist a high-level planner in charge of interpreting high-level commands from the cinematography director, translate them into different tasks (e.g., positions to visit and specific shots to be taken) and distribute them among the UAVs of the team. The whole set of high-level commands specified by the director will be denoted as the *Shooting Mission*, and it will be split into sequential or parallel *Shooting Actions* to be performed by the different UAVs in the team. Thus, each UAV will incorporate the necessary functionalities to perform its assigned Shooting Actions, making use of additional path planners depending on each action. As stated above, the high-level planner will be used for pre-planning, but also for re-planning after unexpected events or new director's commands.

This is illustrated in Figure 1, with the interconnections of the different blocks. The director is able to specify a set of artistic shots or so-called high-level commands through a GUI named ***Director Dashboard***. Those commands will make up the Shooting Mission, which could be saved into a XML-based file with some language to describe missions. The mission together with the mission configuration (i.e., specific parameters, annotated maps, etc.) are fed into the ***High-level Planner***, which splits it into Shooting Actions and assigns them to the different UAVs as high-level plans. Each UAV runs onboard a UAV ***Scheduler*** that is in charge of executing the actions assigned to it in the right order. When the ***High-level Planner*** or any UAV need to compute a path to navigate somewhere, they can make use of the ***Path Planner*** functionality to find different types of safe paths to the destination. Then, different lower-level controllers are activated at each moment depending on the action in hand for the UAV, namely a ***Trajectory Follower*** or a ***Target Tracker***. The former is used to navigate along a specific path (it could be while shooting or not), whereas the latter is used to track a specific target while shooting.

5 Modules Description

In this section, the different blocks from the planning architecture are described in detail together with their interfaces.

5.1 High-level Planner

This module receives high-level commands specified by the cinematography director throughout the *Director Dashboard*. In particular, the idea is that the director determines which shots are of interest from the artistic and cinematographic point of view, and at which specific times they should be taken. Thus, the director's input could be summarized as a list of desired Shooting Actions `List(ACTION)`, the so-called Shooting Mission. Table 5.1 describes the data type

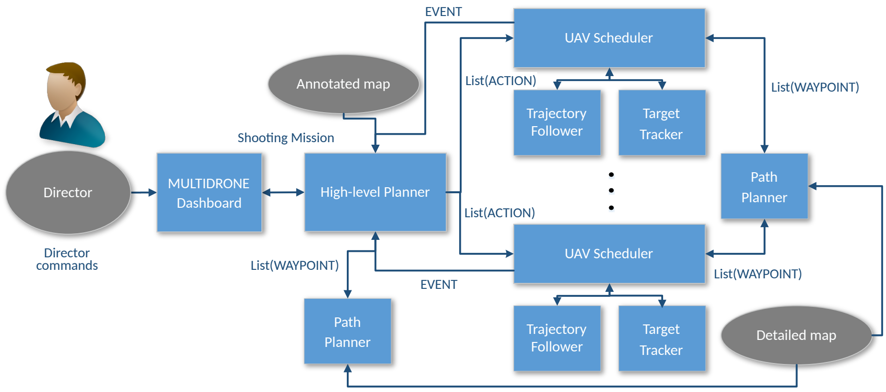

Fig. 1. Block diagram for the planning architecture.

`ACTION`, containing the information required to perform a Shooting Action. That information may be partially specified by the director, and partially computed by the planner (e.g., paths to reach a certain point).

In an `ACTION` data type, the director can specify different information depending on the type of action/shot, i.e., not all fields make sense for any shot. First, starting positions for the camera (look-from) and for the target (look-at), are specified. Depending on the shot, the UAV could place the camera at the corresponding position and point at the target and finish there, or could transition towards the ending look-from/look-at positions while taking a visually pleasant shot. However, the ending position could not make sense for some shots or could be computed automatically in some cases. For instance, if the director wants an *Orbital* or *Vertical* shot of a static point, the ending shot coordinates will be the same as the starting ones; if the shot consists of tracking a moving target for a certain time, it may not be possible to specify the ending position beforehand.

Additionally, a path to the start position or between the start and the end position could be provided. This could be specified by the director or pre-computed by the *High-level Planner* making use of the *Path Planner* module. When not specified, each UAV will be in charge of computing those paths individually accessing the same functionality. Note that during a transition, the trajectories for the look-from and look-at points must be safe (no obstacle collision) and smooth (visually pleasant), and they will also be influenced by the type of shot (*Orbital*, *Lateral*, etc.); whereas only safety is relevant to reach the start position. This will be considered by setting different parameters to the *Path Planner*, which could compute different types of paths depending on the case.

The target size can be used (together with the distance to the target at each moment) to determine the focal length for the camera, and it will depend on the type of framing shot (i.e., long shot, medium shot, close-up, etc.). The director can also specify a starting time for the shooting and a duration, which

ACTION		
Field name	**Data type**	**Comment**
Shot type	Discrete value	See Section 3
Look-from start position	3D global coordinates	Initial position for camera
Look-at start position	3D global coordinates	Initial position for target
Look-from end position	3D global coordinates	Final position for camera
Look-at end position	3D global coordinates	Final position for target
Path to start	`List(WAYPOINT)`	Path to go to start position
Path to shoot	`List(WAYPOINT)`	Path from start to end position
Target size	Pixels or image %	Size of target in image
Starting time	Time in seconds	Time to start shot
Duration	Time in seconds	Duration of the shot
Priority	Discrete level	Priority of this action
Cooperating UAVs	List of natural numbers	*IDs* of other UAVs for this shot

Table 1. Structure for the data type `ACTION`, specifying parameters for a Shooting Action.

will imply a velocity profile for the UAV. Since the timing of the action may be uncertain, different alternatives will be studied. An acceptable time interval could be specified, instead of a concrete time; or a triggering signal from the director could be added to start each shot.

Besides, a priority level and the identifiers (*IDs*) of other UAVs performing the action can be specified. Each action is assigned to and performed by a specific UAV, but in multi-camera shots, several UAVs will be assigned actions correlated (e.g., several UAVs tracking the same target from different perspectives). This selection may be done by the director or automatically by the *High-level Planner*. In those cases, information about the other UAVs involved could be useful for coordination during the execution of the action.

The *High-level Planner* receives the annotated map with high-level information about the environment, i.e., positions of landing spots, re-charging areas, points of interest for opportunistic shooting, no-fly zones, flight corridors, etc. This data could be contained in an KML-based file and would be updated along the mission, for instance, because new points of interest are detected or introduced by the director.

Once the *High-level Planner* gets the complete Shooting Mission, it must convert it into a list of Shooting Actions `List(ACTION)` and solve an assignment problem. There is a team of UAVs with certain resources (e.g., battery levels) and a list of shots to be assigned to them. The planner must take into account all the constraints in terms of time (meeting starting shot times and durations) and resources (flight time is bounded for UAVs, there is a limited number of UAVs), and solve the assignment trying to minimize for instance the travelled distance for the UAVs (other criteria could be used here). In general, the *High-level Planner* will require to use the *Path Planner* at different levels. For instance, safe paths to the start positions for each UAVs could be computed solving a multi-UAV navigation problem.

The *High-level Planner* outputs a particular list of actions for each UAV, or an error whether there is no feasible plan. If there were priorities for the actions, the planner could decide to assign those with highest priority and leave others free. This plan can be delivered to each UAV so that they execute them in parallel. Besides, anytime new inputs from the director are received or any unexpected event happens (e.g., an emergency, a UAV failing to meet the plan, an UAV idle after having finished its shot, etc.), the *High-level Planner* needs to compute a new plan in order to account for those new circumstances, i.e., it must re-plan. Thus, another type of event to report to the planner is when a UAV finishes its action and it is idle, because re-planning could be a way of assigning it previous actions that were waiting because they had less priority.

5.2 Scheduler

This module will be in charge of executing the plan assigned to the UAV. For that, different phases can be considered. First, the UAV needs to travel to its starting position, which takes place before the shooting itself. If no path is specified, it can use the *Path Planner* module to get a path toward its destination. Note that path should be safe but not necessarily fulfill with cinematography constraints, since there is no shooting involved. Second, the UAV needs to transition from the starting shot position to the ending position while shooting. Again, the *Path Planner* can be used to compute the transitioning path if not already specified. However, this time the path must be smooth in terms of look-from and look-at trajectories, since cinematographic constraints should hold. Finally, the UAV could need to fly to a landing spot or toward a new shot, making use again of the *Path Planner*.

Once the *Scheduler* has a path to follow, it forwards it to another module in terms of the low-level control, the *Trajectory Follower*. Of course, the *Scheduler* can report on the *High-level Planner* about the successful execution of the plan or about any problems encountered in the process by means of the data type `EVENT` (see Table 5.2). When the task of the UAV is tracking a moving target, another module can be used, which is the *Target Tracker*. In this case, the *Scheduler* specifies the starting position of the target and any other information available and the *Tracker* controls the UAV to track it. Note that these two different modules are needed in general because the behavior to follow a predefined path and track a target are different. In the particular case that the movement of the target could be predicted accurately, a path could be computed to track it and the *Trajectory Follower* would be used. Moreover, the *Scheduler* could have information about the other cooperating UAVs for a given shot, which could be used for synchronization or formation control with other team-members before starting tracking or taking the shot.

5.3 Path Planner

This module provides paths, i.e. a list of waypoints `List(WAYPOINT)`, to a given destination. See Table 5.3 for the structure of the data type `WAYPOINT`. Depending

EVENT		
Field name	**Data type**	**Comment**
Type	Discrete value	Emergency, idle, error, etc.
UAV position	3D global coordinates	Position of UAV

Table 2. Structure for the data type `EVENT`, used to give feedback to the planner.

WAYPOINT		
Field name	**Data type**	**Comment**
Look-from position	3D global coordinates	Position for camera
Look-at position	3D global coordinates	Position for target
UAV velocity	3D velocity	Velocity when passing by the waypoint

Table 3. Structure for the data type `WAYPOINT`, used to specify paths.

on the case, there may be different versions of the paths computed. Sometimes the UAV only needs to travel somewhere without shooting. For that, standard path planning algorithms for safe navigation could be applied. There would be no need to specify positions for the look-at points, but only for the UAV (look-from). Velocities at each waypoint could also be specified as an option.

If the UAV needs to perform a transition between two frames while taking a shot, we are interested in computing paths that make sense with cinematography. Thus, we may need to specify positions for the look-from and the look-at points, which will translate into camera positions and gimbal angles. This path generation problem has been considered in the literature in the past and could be solved as a coupled optimization problem (UAV and gimbal). For instance, the yaw of the drone could be fixed to point at the target direction all time. In any case, the path planned for the UAV should satisfy some constraints in terms of field of view, since the gimbal cannot usually move freely in any angular range.

In summary, path planning is considered here to compute a path (list of waypoints) to be followed by the UAV. All paths must be safe ensuring there is no obstacle collision, but depending on the case, we may want to specify also positions for the camera to point or velocities for each waypoint. Of course, the module receives as input a detailed map of the environment for collision-free path generation. Moreover, the option of computing collision-free paths for multiple UAVs jointly may also be available within this module.

5.4 Trajectory Follower and Target Tracker

These modules are in charge of the low-level control of the UAV and the gimbal, so-called *Control Actions*. The first one is used when a path can be specified and must be followed. The trajectory of the look-from point would be followed by controlling the movement of the UAV, whereas the trajectory of the look-at point would require to control the gimbal. These two control schemes could work independently depending on the method used. Moreover, the focal length of the camera could also be controlled during the execution of the trajectory. As the UAV gets closer or farther to the target, for instance, for avoiding obstacles, the focal length should adapt automatically to keep a constant target size. Moreover, certain cinematography rules will be imposed, such as the *rule of thirds*, which indicates that the target should be placed covering a third of the image.

In the case of tracking a moving target, it is plausible that there is no specific trajectory for the UAV to follow. A particular control scheme for tracking would be used. One option is to control the UAV to follow the target and point at it (by fixing its heading), and at the same time control the gimbal to keep the target centered on the image plane (or following the rule of thirds).

6 Conclusions

This paper presented a first version of a planning architecture for multidrone filming applications. The application of planning for cinematography with a team of UAVs has been introduced, together with the current state of the art and the involved challenges. Then, a planning architecture has been proposed, describing the main modules and their interfaces in detail. This architecture is being implemented in the framework of the EU-project MultiDrone, where multi-UAV planning algorithms will be developed and tested for outdoor media production (e.g., filming football games, cycling or boat races).

As a next step, a simulated environment has been prepared to deal with multiple UAVs and cameras (gimbals) on board. This simulator is based on Gazebo [5] and uses an abstraction layer developed by the authors to send high-level commands to the UAVs [6]. The simulation will be useful to test the whole architecture and the planning algorithms before integrating them in real platforms. The objective is to implement and test the whole system in a real application throughout the MultiDrone project.

Acknowledgements. 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 only the authors' views. The European Commission is not responsible for any use that may be made of the information it contains.

References

1. Allen, R., Pavone, M.: A Real-Time Framework for Kinodynamic Planning with Application to Quadrotor Obstacle Avoidance. In: AIAA Guidance, Navigation, and Control Conference. p. 1374. American Institute of Aeronautics and Astronautics, Reston, Virginia (jan 2016), http://arc.aiaa.org/doi/10.2514/6.2016-1374

[5] http://gazebosim.org

[6] https://github.com/grvcTeam/grvc-ual

2. Arev, I., Park, H.S., Sheikh, Y., Hodgins, J., Shamir, A.: Automatic editing of footage from multiple social cameras. ACM Transactions on Graphics (TOG) 33(4), 81 (2014)
3. Carr, P., Mistry, M., Matthews, I.: Hybrid robotic/virtual pan-tilt-zom cameras for autonomous event recording. In: Proceedings of the 21st ACM international conference on Multimedia. pp. 193–202. ACM (2013)
4. Coaguila, R., Sukthankar, G.R., Sukthankar, R.: Selecting Vantage Points for an Autonomous Quadcopter Videographer. In: The Twenty-Ninth International FLAIRS Conference. pp. 386–391. AAAI (2016)
5. Daniyal, F., Cavallaro, A.: Multi-camera scheduling for video production. In: Visual Media Production (CVMP), 2011 Conference for. pp. 11–20. IEEE (2011)
6. Galvane, Q., Fleureau, J., Tariolle, F.L., Guillotel, P.: Automated Cinematography with Unmanned Aerial Vehicles. Eurographics Workshop on Intelligent Cinematography and Editing (2016)
7. Gebhardt, C., Hepp, B., Nägeli, T., Stevšić, S., Hilliges, O.: Airways: Optimization-Based Planning of Quadrotor Trajectories according to High-Level User Goals. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16. pp. 2508–2519. ACM, ACM Press, New York, New York, USA (2016), http://dl.acm.org/citation.cfm?doid=2858036.2858353
8. Joubert, N., E, J.L., Goldman, D.B., Berthouzoz, F., Roberts, M., Landay, J.A., Hanrahan, P.: Towards a Drone Cinematographer: Guiding Quadrotor Cameras using Visual Composition Principles. ArXiv e-prints (2016)
9. Joubert, N., Roberts, M., Truong, A., Berthouzoz, F., Hanrahan, P.: An interactive tool for designing quadrotor camera shots. ACM Transactions on Graphics 34(6), 1–11 (oct 2015), http://dl.acm.org/citation.cfm?doid=2816795.2818106
10. Lino, C., Christie, M.: Intuitive and efficient camera control with the toric space. ACM Transactions on Graphics 34(4), 82:1–82:12 (jul 2015), http://dl.acm.org/citation.cfm?doid=2809654.2766965
11. Mellinger, D., Kumar, V.: Minimum snap trajectory generation and control for quadrotors. In: 2011 IEEE International Conference on Robotics and Automation. pp. 2520–2525. IEEE, IEEE (may 2011), http://ieeexplore.ieee.org/document/5980409/
12. Mueller, M., Sharma, G., Smith, N., Ghanem, B.: Persistent Aerial Tracking system for UAVs. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1562–1569. IEEE (oct 2016), http://ieeexplore.ieee.org/document/7759253/
13. Naseer, T., Sturm, J., Cremers, D.: FollowMe: Person following and gesture recognition with a quadrocopter. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 624–630. IEEE (nov 2013), http://ieeexplore.ieee.org/document/6696416/
14. Richter, C., Bry, A., Roy, N.: Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor Environments. In: Inaba, M., , Corke, P. (eds.) Robotics Research: The 16th International Symposium ISRR, pp. 649–666. Springer (2016), http://link.springer.com/10.1007/978-3-319-28872-7_37
15. Saeed, A., Abdelkader, A., Khan, M., Neishaboori, A., Harras, K.A., Mohamed, A.: On Realistic Target Coverage by Autonomous Drones. arXiv preprint arXiv:1702.03456 (feb 2017), http://arxiv.org/abs/1702.03456
16. Smith, C.: The Photographer's Guide to Drones. Rocky Nook, Inc. (2016)
17. Teuliere, C., Eck, L., Marchand, E.: Chasing a moving target from a flying UAV. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 4929–4934. IEEE (sep 2011), http://ieeexplore.ieee.org/document/6094404/