16711 Kinematics, Dynamics and Controls - Project Report

Rogerio Bonatti (rbonatti), Jerry Hsiung (shsiung) , Ratnesh Madaan (ratneshm)

Due: 05/10/2018

Abstract

Autonomous aerial cinematography has the potential to enable automatic capture of aesthetically pleasing videos without requiring human intervention, empowering individuals with the capability of high-end film studios. Current approaches either only handle off-line trajectory generation, or offer strategies that reason over short time horizons and simplistic scenarios for obstacles, which result in jerky movement and low generalization. In this work we develop an approach that is able to trade off shot smoothness, occlusion, and cinematography guidelines in a principled manner, even under noisy actor predictions. We present a novel algorithm for real-time covariant gradient descent that we use to efficiently find the desired trajectories by optimizing a functional based on a set of cost functions. We also compare our algorithm with gaussian process motion planning. Our results show that the approach is able to create attractive shots, which we validate experimentally re-planning at 5Hz with a 10s time horizon in more than 1.25 hours of real-life experiments while filming human actors, cars and bicycles among obstacles.

1 Introduction

Aerial vehicles are revolutionizing the way both professional and amateur film makers capture shots of actors and landscapes, increasing the flexibility of narrative elements and allowing the composition of aerial viewpoints which are not feasible using traditional devices such as hand-held cameras and dollies. However, the use of drones for filming today is still extremely difficult due to several motion planning and human-computer interaction challenges. Aerial cinematography incorporates objectives from different areas, such as high-speed flight [2] ,inspection and exploration, formation flight [23], and artistic intent [1, 3].

Previous approaches in aerial filming do not address the complete problem in a sufficiently generalizable manner to be used in real-life scenarios. Off-line trajectory generation cannot be used for most practical situations, and the on-line trajectory generation methods that have been proposed have limitations such as ignoring artistic objectives or only dealing with limited obstacle representations (ignoring obstacles altogether in many cases).

Our key insight in this work is that this problem can be efficiently solved in real-time as a smooth trajectory optimization. Our contributions in this project are: (1) we formalize the aerial filming problem following cinematography guidelines for arbitrary types of shots and arbitrary obstacle shapes, (2) we present an efficient optimization method that exploits covariant gradients of the objective function for fast convergence, (3) we compare the covariant optimization approach with Gaussian process motion planning, and (4) for over 1.25 hours of flight time while re-planning, we experimentally show robustness in real-world conditions with different types of shots and shot transitions, actor motions, and obstacle shapes.

2 Problem definition and related work

Following literature in cinematography [1, 3], we identified a small set of parameters that can define a large span of shot types (Figure 1). We define *static shots* as shots whose parameters remain static over time, independently of the motion of the actor in the environment, and *dynamic shots* as having time-dependant parameters. We focus on scenarios where one quadrotor is filming one actor, with trajectories ξ_q and ξ_a respectively, where $\xi(t)$ maps time $t \in [0, t_f]$ to a configuration.

In a generic framework for aerial filming, the motion planner's objective is to minimize a cost function $J\left(\xi_q\right)$, that results in a smooth, safe, occlusion-free trajectory that follows our artistic guidelines as closely as possible:

$$J\left(\xi_q\right) = J_{\text{shot}}\left(\xi_q, \xi_a\right) + J_{\text{smooth}}\left(\xi_q\right) + J_{\text{occ}}\left(\xi_q, \xi_a\right) + J_{\text{obs}}\left(\xi_q\right)$$

This framework borrows concepts from the disparate fields of virtual cinematography, world representation and smooth trajectory generation. Camera control in virtual cinematography is typically on through-the-lens control [4] but disregards real-world limitations such as robot physics constraints and noisy motion predictions. When dealing with arbitrary real-life environments, voxel occupancy maps and truncated signed distance field (TSDF) [16] are common representations, and can supply distance and gradient of a point to nearest object surface. Aerial trajectory generation methods [13, 22] are typically designed for aggressive flight and rely on easy to evaluate objective or constraint functions. In general domains, we find techniques with more relaxed requirements [17, 21] . We build on CHOMP [18] due to its simple update rule that is amenable to new cost functions.

Another framework that we considered is Gaussian Process Motion Plannin (GPMP) [5, 14]. In this framework, a trajectory is formulated using smooth Gaussian prior and optimized using sampled points from the Gaussian process function. Dong et al. [5] has shown that using state-of-the-art factor graph optimizer, the performance of GPMP can be suprior than that of CHOMP. As part of the project we would like to implement our cost function using GPMP for comparison.

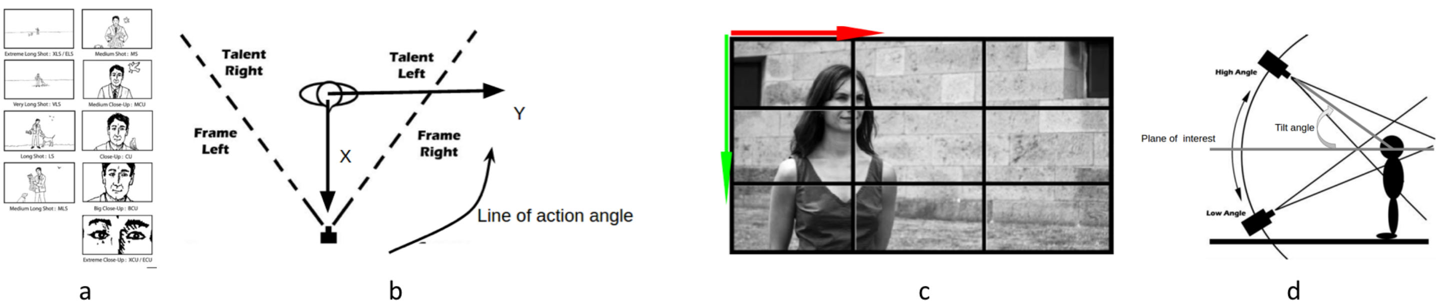

Figure 1: Shot parameters, adapted from Bowen and Thompson [3]: a) shot scale ss corresponds to the size of the projection of the actor on the screen; b) line of action angle $\phi_{rel} \in [0, 2\pi]$; c) screen position of the actor projection $sp_x, sp_y \in [0, 1]$; d) tilt angle $\theta_{rel} \in [-\pi, \pi]$

On aerial filming, Roberts and Hanrahan [20] generate off-line trajectories given infeasible human-defined key-frames and Joubert et al. [10] provide a tool for interactive off-line design of camera trajectories. Lino and Christie [12] analytically interpolate between viewpoints while maintaining shot quality. Galvane et al. [6] used this method to control quadrotors, but only in obstacle-free environments. Galvane et al. [7] considered dynamic targets, however they do not address outdoor settings with occlusion and noise. Similarly, Joubert et al. [11] transition between shots for static actors while not colliding with them, but offer no solution to obstacle and occlusion avoidance. Closest to our work, Nägeli et al. [15] apply MPC considering occlusion and safety. However, they plan for short time horizons, use simplistic elliptical representations for all obstacles, and use a high-accuracy motion-capture system. It is not clear if the black-box MPC solver that is used is amenable to other obstacle representations and noise in localization. In contrast, our method works for long time horizons, has a simple unconstrained update rule, operates on TSDFs, has small runtime onboard and can deal with noise in actor motion predictions.

3 Approach

Unlike previous work that operate either with high-accuracy indoor motion capture systems or precision RTK GPS outdoors, we use only conventional GPS, resulting in high noise for both drone localization and actor motion prediction. Therefore we decided to decouple the motion of the drone and the camera. The camera is mounted on a 3-axis independent gimbal and can place the actor on the correct screen position, solely using the visual input, despite errors in drone position. With camera decoupled, the trajectory to be optimized becomes $\xi_q(t) = [x_q(t)\ y_q(t)\ z_q(t)]^T$, assuming that the drone's orientation ψ_q points towards the actor at all times.

Designing differentiable cost functions for cinematography

We want trajectories which are smooth, safe, occlusion-free and that follow our artistic guidelines as closely as possible. Following the derivation seen in Section 3 of Zucker et al. [24], we define a parametrization-invariant smoothness cost that can be expressed as a quadratic function, and an obstacle avoidance cost based on a penalization c of the TSDF, for both the environment and the dynamic actor. In addition, we define two more cost functions specifically for cinematography: *Shot quality*:

$$J_{\text{shot}}\left(\xi_q, \xi_{shot}\right) = \frac{1}{t_f}\frac{1}{2}\int_0^{t_f} ||\xi_q(t) - \xi_{shot}(t)||^2 dt \approx \frac{1}{2(n-1)} Tr(\xi_q^T A_{shot}\xi_q + 2\xi_q^T b_{shot} + c_{shot})$$
$$\nabla J_{\text{shot}}\left(\xi_q\right) = \frac{1}{n-1}(A_{shot}\xi_q + b_{shot})$$

Written in a quadratic form, it measures the average squared distance between ξ_q and an ideal trajectory ξ_{shot} that only considers positioning via cinematography parameters. ξ_{shot} can be computed analytically: for each point $\xi_a(t)$ in the actor motion prediction, the drone position lies on a sphere centered at the actor with radius calculated via the shot scale, and angles given by ϕ_{rel} and θ_{rel}, as in Figure 1.

Occlusion avoidance:

$$J_{\text{occ}}\left(\xi_q, \xi_a\right) = \int_{t=0}^{t_f}\int_{\tau=0}^{1} c(p(\tau))\left|\left|\frac{d}{d\tau}p(\tau)\right|\right| d\tau \left|\left|\frac{d}{dt}\xi_q(t)\right|\right| dt$$

Even though the concept of occlusion is binary, *i.e*, we either have or don't have visibility of the actor, a major contribution of our work is to define a differentiable cost that expresses a viewpoint's occlusion intensity for arbitrary obstacle shapes. Mathematically, occlusion is the integral of the TSDF cost c over a 2D manifold connecting both trajectories ξ_q and ξ_a. We then have the functional gradient:

$$\nabla J_{\text{occ}}\left(\xi_q, \xi_a\right)(t) = \int_{\tau=0}^{1} \nabla c(p(\tau))|L||\dot{q}| \left[I - (\hat{q} + \tau(\frac{\dot{a}}{|\dot{q}|} - \hat{q}))\hat{q}^T\right] - c(p(\tau))|\dot{q}| \left[\hat{L}^T + \frac{\hat{L}^T\dot{L}\hat{q}^T}{|\dot{q}|} + |L|\kappa^T\right] d\tau$$

Where:

$$q = \xi_q(t), \quad a = \xi_a(t), \quad p(\tau) = (1-\tau)q + \tau a, \quad \hat{v} = \frac{v}{|v|}, \quad \kappa = \frac{1}{|\dot{q}|^2}(I - \hat{q}\hat{q}^T)\ddot{q}, \quad L = a - q$$

Intuitively, the term $\nabla c(p(\tau))$ is related to variations of the gradient in space, and the term τ acts as a lever for the gradient. The term $c(p(\tau))$ is linked to changes in path length between camera and actor.

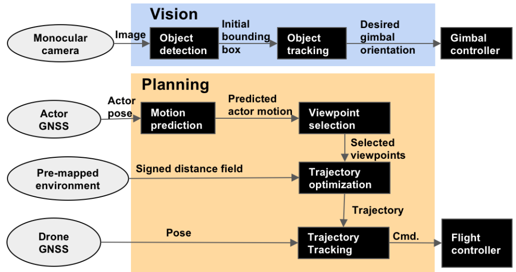

Figure 2: System architecture. The vision subsystem controls the camera orientation using only the monocular image, independently of the planning subsystem. Planning uses the drone's and actor's current location, and the environment to generate trajectories for the flight controller.

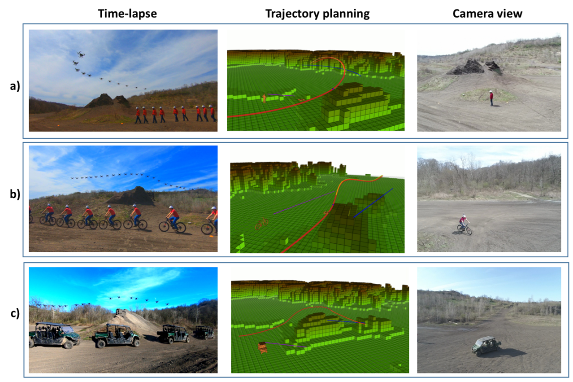

Figure 3: Results: a) Circling shot around person, b) Side shot following biker, c) Side shot following vehicle. The planned trajectory (red) avoids colliding with and being occluded by the mountain, while remaining smooth even under high actor motion prediction noise. The actor's motion forecast is in purple, and the desired artistic shot is in blue.

Covariant gradient and steepest descent method

Our objective is to minimize the functional $J(\xi_q)$. Following a first-order Taylor expansion around the current iteration k using gradient g_k, we have [18]:

$$\xi_{qk+1} = \xi_{qk} - \frac{1}{\lambda} M^{-1} g_k$$

Incorporating costs to GPMP solver

GPMP utilizes factor graph, which solves for Maximum a Posterior (MAP) inferences assuming Gaussian noise [5]. The optimizer solves for the below cost

$$\xi^* = \arg\min_{\xi} \{ \frac{1}{2} \|\xi - \mu\|_{\kappa}^2 + \frac{1}{2} \|h_{\text{obs}}(\xi)\|_{\text{obs}}^2 + \frac{1}{2} \|h_{\text{shot}}(\xi)\|_{\text{shot}}^2 + \frac{1}{2} \|h_{\text{occ}}(\xi)\|_{\text{occ}}^2 \}$$

where the three $h(\xi)$ are the cost function defined above. Since this is a nonlinear function, GPMP solves for such system using a linearized system using above Jacobians and iteratively solve for $\delta\xi$ for updates.

4 Experiments

Our robot is the DJI M210 drone and we use the NVIDIA TX2 processor for high-level perception and planning (Figure 2). In the vision pipeline, we detect the actor using Faster-RCNN [19] with MobileNets [9] for feature extraction and Kernelized Correlation Filter (KCF) [8] for tracking, controlling the camera gimbal to keep the actor within the desired screen position. The actor wears a Pixhawk PX4 module on a hat that sends his pose to the onboard computer via radio communication, and a linear Kalman filter estimates his velocity, which is used to predict the actor's pose for the next 10 s. Using a point cloud map of the test site we compute a TSDF map of the region of interest off-line. Re-planning happens at 5 Hz with a 10 s horizon.

(a) Algorithm robustness: We evaluated our algorithm performing different types of static and dynamic shots, following different types of actors: humans, cars and bicycles at different speeds and motion types. In total, we collected over 1.25 hours of flight time while re-planning and avoided obstacles and/or occlusion 65 times. The maximum velocity achieved during the tests was of 7.5 m/s. Figure 3 summarizes the most representative shots.

(b) Using occlusion cost function: For the same shot type, we planned paths with and without the occlusion cost function (Figure 4).

(c) Generating smooth trajectories: We compare tracking performance under real-world noise between greedily tracking the current artistic viewpoint versus trajectory optimization (Figure 5).

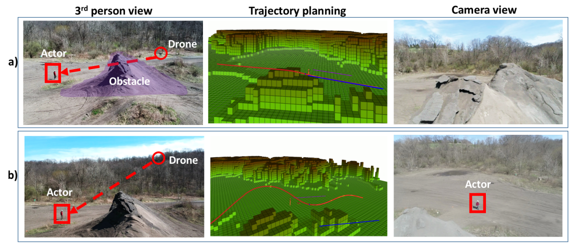

Figure 4: Comparison of planning a) without and b) with occlusion cost function. The occlusion cost function significantly improves the quality of the camera image in comparison with pure obstacle avoidance, for same shot type.

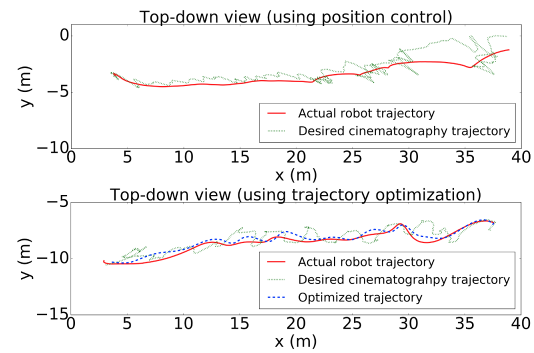

Figure 5: Top-down view of drone motion. For naive tracking of artistic viewpoint the error is high, in the order of meters with respect to the target position, because it ignores dynamic feasibility. In our method the tracking error with respect to the optimized trajectory is small even under real-world noise, because we consider smoothness.

(d) Simulation for GPMP solver with designed costs We implemented the cost function using Matlab and have designed an example where an obstacle is blocking the view of the drone, so the drone will plan to move in front of the obstacle even though the shortest path is behind the obstacle. Figure 6 shows the result plot.

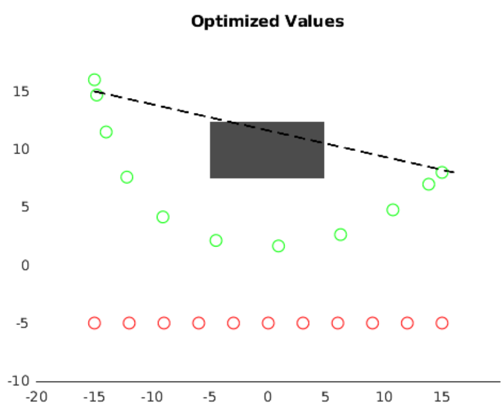

Figure 6: The red dots represent the actor trajectory; the black dotted line is the direct path; the green dots represent the optimized path. As the result shows that GPMP converges to the Green path, which avoids the obstacle and still provides a clear shot of the actor.

5 Discussion and conclusion

We validated our original hypothesis with the experiments: (a) Our algorithm is robust and fast enough to work on a real life scenario running on an onboard computer, generating safe, smooth, occlusion-free trajectories following artistic guidelines. (b) The occlusion cost function we introduced significantly improves the quality of the resulting image. (c) Planning trajectories renders much less error than naively reasoning about the current best viewpoint.

In this project we have implemented the trajectory optimizer using both GPMP and CHOMP. We have noticed that GPMP is sensitive to initialization and parameter tuning, since the hyper-parameters of the Gaussian Process prior really affects the convergence. We have noticed that the solver converges to local minimum in various scenarios, and therefore we have decided to perform drone test using CHOMP rather than GPMP.

As future work, we want to validate the performance of our algorithm from a statistical perspective, plotting performance curves while varying the complexity of the environment, the actor trajectory, shot type and initialization. In addition, we will study failure cases for the algorithm, and how to avoid them. To improve GPMP implementation we will have to derive our own system dynamics, which will improve the Gaussian Process prior parameters and makes it more robust to various initialization parameter tuning.

References

[1] Daniel Arijon. Grammar of the film language. 1976.

[2] Andrew J Barry, Peter R Florence, and Russ Tedrake. High-speed autonomous obstacle avoidance with pushbroom stereo. *Journal of Field Robotics*, 35(1):52–68, 2018.

[3] Christopher J Bowen and Roy Thompson. *Grammar of the Shot*. Taylor & Francis, 2013.

[4] Marc Christie, Patrick Olivier, and Jean-Marie Normand. Camera control in computer graphics. In *Computer Graphics Forum*, volume 27, pages 2197–2218. Wiley Online Library, 2008.

[5] Jing Dong, Mustafa Mukadam, Frank Dellaert, and Byron Boots. Motion Planning as Probabilistic Inference using Gaussian Processes and Factor Graphs. *Robotics: Science and Systems XII*. doi: 10.15607/RSS.2016.XII.001. URL http://www.roboticsproceedings.org/rss12/p01.pdf.

[6] Quentin Galvane, Julien Fleureau, Francois-Louis Tariolle, and Philippe Guillotel. Automated cinematography with unmanned aerial vehicles. *arXiv preprint arXiv:1712.04353*, 2017.

[7] Quentin Galvane, Christophe Lino, Marc Christie, Julien Fleureau, Fabien Servant, and Philippe Guillotel. Directing cinematographic drones. *arXiv preprint arXiv:1712.04216*, 2017.

[8] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015.

[9] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[10] Niels Joubert, Mike Roberts, Anh Truong, Floraine Berthouzoz, and Pat Hanrahan. An interactive tool for designing quadrotor camera shots. *ACM Transactions on Graphics (TOG)*, 34(6):238, 2015.

[11] Niels Joubert, Dan B Goldman, Floraine Berthouzoz, Mike Roberts, James A Landay, Pat Hanrahan, et al. Towards a drone cinematographer: Guiding quadrotor cameras using visual composition principles. *arXiv preprint arXiv:1610.01691*, 2016.

[12] Christophe Lino and Marc Christie. Intuitive and efficient camera control with the toric space. *ACM Transactions on Graphics (TOG)*, 34(4):82, 2015.

[13] Daniel Mellinger and Vijay Kumar. Minimum snap trajectory generation and control for quadrotors. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2520–2525. IEEE, 2011.

[14] Mustafa Mukadam, Xinyan Yan, and Byron Boots. Gaussian Process Motion planning. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9–15, 2016. ISSN 10504729. doi: 10.1109/ICRA.2016.7487091. URL http://ieeexplore.ieee.org/document/7487091/.

[15] Tobias Nägeli, Javier Alonso-Mora, Alexander Domahidi, Daniela Rus, and Otmar Hilliges. Real-time motion planning for aerial videography with dynamic obstacle avoidance and viewpoint optimization. *IEEE Robotics and Automation Letters*, 2(3):1696–1703, 2017.

[16] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[17] Nathan Ratliff, Matthew Zucker, J Andrew Bagnell, and Siddhartha Srinivasa. CHOMP: Gradient optimization techniques for efficient motion planning. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 489–494. IEEE, 2009.

[18] Nathan D Ratliff, David Silver, and J Andrew Bagnell. Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots*, 27(1):25–53, 2009.

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[20] Mike Roberts and Pat Hanrahan. Generating dynamically feasible trajectories for quadrotor cameras. *ACM Transactions on Graphics (TOG)*, 35(4):61, 2016.

[21] John Schulman, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, and Pieter Abbeel. Finding locally optimal, collision-free trajectories with sequential convex optimization. In *Robotics: Science and Systems*, volume 9, pages 1–10. Citeseer, 2013.

[22] David H Shim, H Jin Kim, and Shankar Sastry. Decentralized nonlinear model predictive control of multiple flying robots. In *Decision and control, 2003. Proceedings. 42nd IEEE conference on*, volume 4, pages 3621–3626. IEEE, 2003.

[23] Matthew Turpin, Nathan Michael, and Vijay Kumar. Trajectory design and control for aggressive formation flight with quadrotors. *Autonomous Robots*, 33(1-2):143–156, 2012.

[24] Matt Zucker, Nathan Ratliff, Anca D Dragan, Mihail Pivtoraiko, Matthew Klingensmith, Christopher M Dellin, J Andrew Bagnell, and Siddhartha S Srinivasa. CHOMP: Covariant hamiltonian optimization for motion planning. *The International Journal of Robotics Research*, 32(9-10):1164–1193, 2013.