

Uncut Aerial Video via a Single Sketch

Hao Yang Ke Xie Shengqiu Huang Hui Huang[†]

Shenzhen University

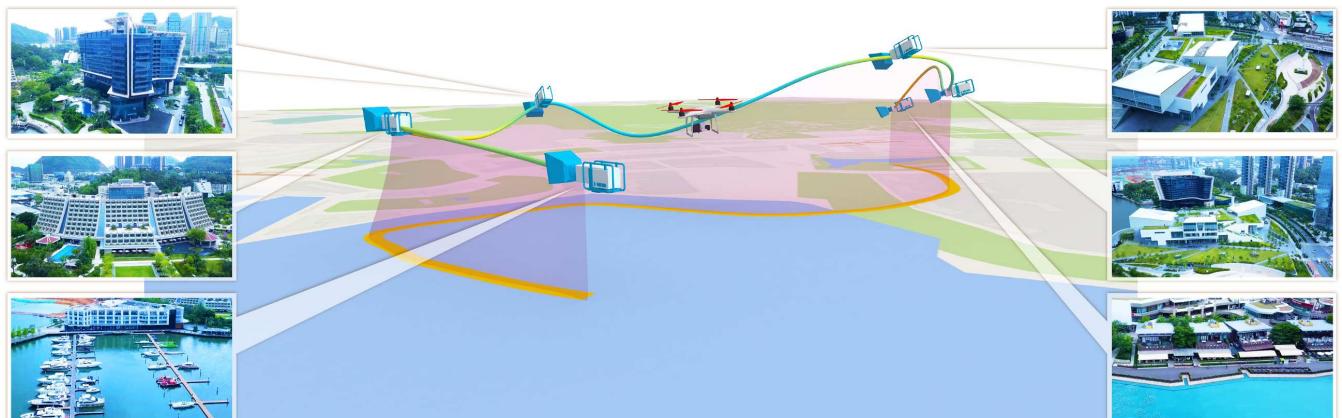


Figure 1: Given a 2D sketch (yellow) that the user casually drew on a map, a safe and suitable camera trajectory can be automatically generated in 3D, along which the drone's flying velocity (color coded) is also optimized to better film the scene in a continuous manner.

Abstract

Nowadays UAV filming is getting popular, more and more stunning aerial videos appearing online. Nonetheless, making a good uncut aerial video with only one-long-shot for the large-scale outdoor scenes is still quite challenging, no many eye-catching pieces available yet. It requires users to have both consummate drone controlling skill and good perception of filming aesthetics. If totally manual, the user has to simultaneously adjust the drone position and the mounted camera orientation during the whole flyby while trying to keep all operation changes executed smoothly. Recent research has proposed a number of planning tools for automatic or semi-automatic aerial videography, however, most requires rather complex user inputs and heavy computations.

In this paper, we propose a user-friendly system designed to simplify the input and automatically generate continuous camera moves to capture compelling aerial videos that users prefer to see without any post cutting or editing. Assume a rough 2.5D scene model that includes all the regions of interest are available, users are only required to casually draw a single sketch on the 2D map. Our system will analyze this rough sketch input, compute the corresponding quality views in 3D safe flying zone, and then create a globally optimal camera trajectory passing through regions of user interest via solving a combinatorial problem. At end, we optimize the drone flying speed locally to make the resulting aerial videos more visually pleasing.

CCS Concepts

- Computing methodologies → Computer graphics; Graphics systems and interfaces; Interest point and saliency detections;

1. Introduction

An unmanned aerial vehicle (UAV), commonly known as a drone, is an aircraft without a human pilot aboard. In recent years, with the rapid expansion of UAV technology in the field of commer-

[†] Corresponding author: Hui Huang (hhzhiyan@gmail.com), College of Computer Science and Software Engineering, Shenzhen University, China.

cial, UAV is becoming a hot spot focusing on the photography and videography. More and more people start to use UAVs shooting their aerial imagery and video footage.

Nonetheless, from the Internet, one can see that most of the aerial videos are made up of short movie segments. In particular, for large-scale outdoor scenarios, almost all available aerial videos are discontinuous and composed of many clips/shots for different architectures or attractive regions. It is thus very hard to really provide the audiences a clear overview or a good geographic understanding of the whole environment. On the other hand, making a good uncut aerial video with only one-long-shot for the large areas is very challenging, as it requires the user to have both consummate drone controlling skill and outstanding perception of filming aesthetics. If totally manual, the user has to simultaneously adjust the drone position and the mounted camera orientation during the whole flight while constantly keeping all operation changes executed smoothly, almost impossible for any novices.

Most recently, the emergence of platforms, such as DJI Ground Station Pro [DJI] and Airnest, enable people to plan automatic flights for the drone. Nonetheless, using these path planning platforms to make a beautiful aerial video is very difficult and complicated, requiring intensive practice and human efforts. With these systems, users have to manually set a waypoint flight path, pre-define the drone configuration at each waypoint, including the altitude, speed, gimbal pitch and aircraft rotation, and may need many times of flying trial and post adjusting. What's more, it is challenging even for professional users to accurately estimate whether at a certain new location, the preset position, altitude, heading, and pitch of the camera can lead to a nice shooting or not.

In this work, we propose a new user-friendly path planning tool, which aims to enable novices easily shoot a continuous aerial video in complex large-scale outdoor scenes. The input to our system is a coarse 2.5D model of the scene, and a 2D sketch casually drew by the user. Our algorithm combines both cinematographic knowledge and aesthetic rules to automatically generate a 6D collision-free trajectory, including drone position, camera orientation and flight speed. The resulting uncut aerial videos and the user evaluations clearly show the superiority of our approach.

The rest of the paper is organized as follows. We review related work in Section 2, and provide an algorithm overview in Section 3. In Section 4 we present our 2D-to-6D trajectory lifting method in detail. Experiments and evaluation are described in Section 5, and we draw conclusion and discuss future work in Section 6.

2. Related work

2.1. Automated Camera Control

The importance of camera control in applications cannot be over-emphasized as a number of implicit rules drive the location and motion of cameras and impact the user's mental representation and understanding of the environment.

Bares et al. [BGL98] claimed that virtual camera must "film" the behaviors of characters. They presented a detailed definition of desired frame properties which are modeled as a set of constraints that the camera configure has to satisfy. Since Bares' definition,

more and more researchers formulated the camera control problem by finding an optimal camera solution that satisfied the frame constraints they set. Halper et al. [HO00] proposed a graphical camera planning system to control the camera, which account for the visual effects and solved their constrained problems with optimization approaches. Kennedy et al. [KM02] transformed the description of animator intentions and character action into a series of camera shots. The descriptor they designed has been used in their camera planning system. Their project contributed a large movement toward more intelligent computer animation tools.

Afterwards, the definition of camera frame properties was the same in essentials while differing in minor points. Most of the definitions were described in terms of visibility, object size and positioning [ZTD05, Bar06, Nor07, SL17]. More researchers turn the problem to the solution of constraint satisfaction problems or apply to different practical problems. The works [BDGER08, DGER08, PHaU16] presented particle swarm optimization approach for the virtual camera composition problem, which worked on static scenes and performs significantly better than an exhaustive search in a discretized space.

If there are moving objects in the scene, real-time camera control is necessary. The system developed in [OSTG09] modeled the real-time camera control problem as a global search on a precomputed visibility-aware roadmap together with local runtime refinement. The roadmap data structure permitted the precomputation of a coarse representation of all collision-free paths through an environment, together with an estimate of the pair-wise visibility between all portions of the scene. Burelli et al. [BP14] formulated the automatic controlling camera problem by phrasing it as a dynamic multi-objective optimization problem. They proposed this method for off-line calculation of camera configuration sequences that can be used to visualize a computer animation or to automatically place camera viewpoints in dynamic environment. Huang et al. [HGP*18] designed a drone camera system to autonomously capture cinematic shots of action scenes. An efficient method is proposed to extract 3D skeleton points via a stereo camera together with a real-time dynamical camera planning strategy.

Nägeli et al. [NMD*17] formulated the motion plan generation and tracking problems as a joint real-time receding horizon optimization problem. The real-time nature of the method allows for incorporating of feedback and dynamic constraints, thus enabling the planning of collision-free paths in cluttered environments. Another related work [NAMD*17] proposed a modular cost function based on the reprojection error of targets and formulated the minimization problem as a finite horizon optimal control problem, which fulfills aesthetic objectives and adheres to nonlinear constraints of the filming robot and collision constraints with obstacles. Joubert et al. [JGB*16] presented a system that attempts to follow composition principles when autonomously capturing footage of people with a drone. They calculated feasible, safe and visually pleasing transitions by using a novel real-time transition planner.

In this paper, we compute the image properties using metrics proposed in [XYH*18]. The saliency features that depict the objects in 3D environment and the popular aerial aesthetic rule (the Rule of Thirds) are all counted. Xie et al. [XYH*18] voxelized the flight zone surrounding the region of interest the user designated

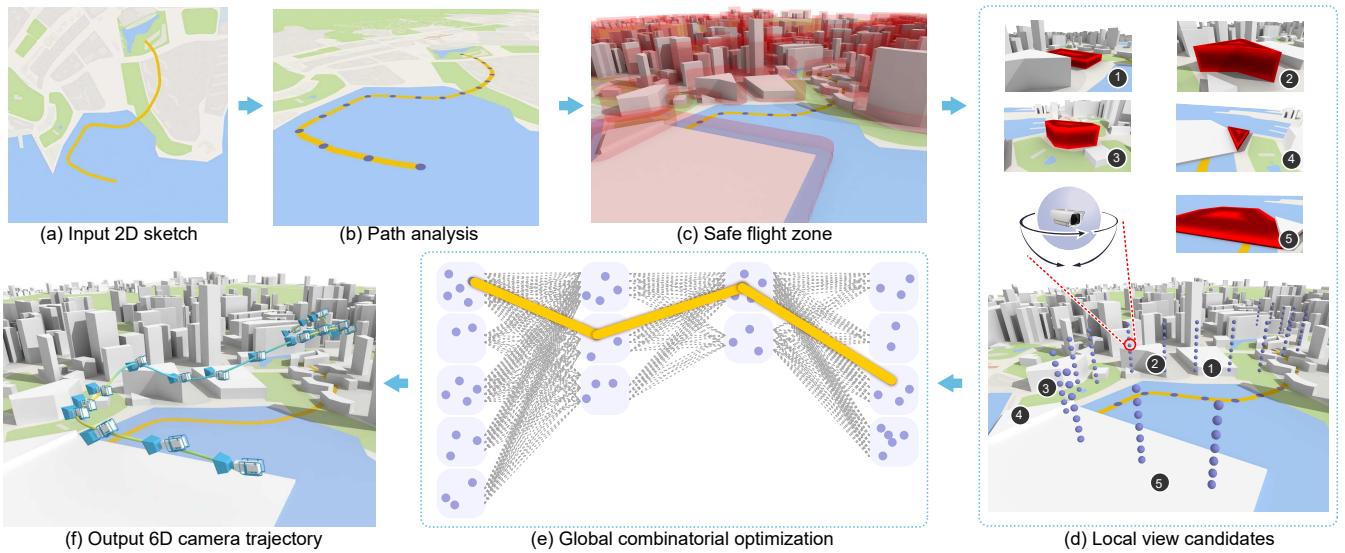


Figure 2: Algorithm overview: starting from a simple 2D user sketch (a), our system can analyze it and sample a set of waypoints (b); then in pre-computed safe flight zone (c), by sampling above the sketch point and positioning virtual cameras on every height sample targeting different regions of interest, we are able to select local view candidates (d); an instance of STSP is constructed next to obtain the path with the lowest total cost (e); after flight speed optimization, the final 6D (position, orientation, speed) camera trajectory is generated (f).

and computed the image quality at each voxel toward the region of interest. Here, to avoid heavy volumetric computation, we sample the view point in the vertical flight zone above each sketch point on the path and compute the image quality at each height sample toward each candidate region of interest.

2.2. UAV Trajectory Planning

Feasible trajectory planning of UAVs and other robots has been tackled actively. Lai et al. [LYW06] modeled the problem as time-optimal control problem and propose a nonlinear programming method to solve it. By using genetic algorithm to solve the NLP, they generate minimum time point-to-point trajectories in simulation. Bouktir et al. [BHC08] used b-spline to fit a set of control points and generate the trajectory candidates. Trajectory candidate was modeled as a composition of a path function and a monotonically increasing motion function. The optimal solution of these two function is found using a nonlinear optimization technique. Deits et al. [DT15] formulated the trajectory planning problem as a mixed-integer convex program, ensuring that the entire piecewise polynomial trajectory is free of collisions using convex constraints.

Fleureau et al. [FGTG16] designed a global control architecture based on a high-level generic API, which integrates rotary-wing drones with full state feedback strategy. The control architecture is combined with an automatic camera path planning approach that encompasses cinematographic techniques. Gebhardt et al. [GSH18] developed a variable, infinite horizon, contour-following algorithm that optimizes positional and temporal reference fit jointly to generate globally smooth trajectories while retaining user control over reference timings. Roberts et al. [RH16] proposed an algorithm that

guaranteed the feasibility of the output trajectory by re-timing the input trajectory, perturbing its timing as little as possible while remaining within velocity and control force limits.

The work [SP08] modeled the path planning problem as a Traveling Salesmen Problem (TSP) to plan the trajectory that interpolated the viewpoints they set. Ke et al. [XYH*18] generated several local trajectories for each region of interest they designated, and then used a generalized version of TSP to select the optimal local trajectories and chain them with transition path to get a final trajectory. Our method adopts this approach to plan the globally optimal path, but the corresponding combinatorial graph problem we formulate is different from the previous work. We aim to find the best altitude, heading, camera pitch of the associated samples, and the optimal transitions among them.

2.3. Interactive Design System

Researchers tend to design interactive camera systems that can simplify the user operations on UAVs. Lan et al. [LSHZ17] introduced an explore-and-compose approach for photo taking and provided an interactive model for flying cameras that enables the user to interact with photos directly instead of piloting the drone. Huang et al. [HYK*18] designed a novel drone control system, called “through-the-lens drone filming”, where the user can capture the visual-pleasing footage by manipulating the 3D target model.

Note that, due to the difficulty of UAV control, most related research was only implemented in simulation environment, although UAVs are more applied to human real lives. Therefore, some works have extended the design of interactive trajectory planning tools to the practical system performance in the challenging real world.

Joubert et al. [JRT^{*}15] designed an interactive tool that enabled users to specify shots visually using keyframes and preview the shots in a virtual environment. They have derived an algorithm for synthesizing C4 continuous camera trajectories from user-specified keyframes and easing curves. The work [GHN^{*}16] presented a user-friendly design tool that helped users create trajectory without requiring deep knowledge in both UAV control. The user could design an initial trajectory around one or several targets and specify the keyframes. Their system would then generate a smooth and collision free drone trajectory and allow the user to virtually fly and shoot by dragging the virtual quadrotor along the trajectory. Lino et al. [LC15] introduced the Toric space for intuitive and effective virtual camera control. An intuitive viewpoint manipulator was proposed to control the visual result directly in screen space which performed brilliantly in the task of crafting viewpoints and was easier to use. Building on Lino’s Toric space model, Galvane et al. [GLC^{*}18] presented the Drone Toric Space as a dedicated camera parameter space with embedded constraints. They also proposed a sketching tool that made automatic path planning from hand drawn input paths, which accounted for cinematographic properties and physical feasibility.

While our approach shares some ideas with [GLC^{*}18], we propose a higher level and more automated design tool. User just need to draw a sketch path in a 2D map, and our tool will automatically compute the best height and best view direction of each point on the sketch path and generate the smooth 3D trajectory.

3. Overview

In this paper, we focus on the automated computation of UAV flight trajectory. Our system intends to simplify user’s aerial video shooting operations. User just requires roughly drawing a 2D sketch without designating the altitude, heading and pitch of the camera. Assume that rough 2.5D scene models that include all the regions of interest are available, our system will automatically compute a feasible and aesthetic trajectory that satisfies the user expectations.

Figure 2 illustrates the overview of our method, which mainly consists of: i) preprocessing the 2D sketch path, ii) planning the trajectory within safe flight zone, iii) optimizing the speed of the UAV, to produce a 6D camera trajectory at end. Specifically, we start by smoothing and sampling the input sketch path. Then we compute the safe flight zone to avoid drone collisions with the buildings or any other obstacles in the scene, where the view points are sampled along a vertical line above each 2D sketch point. A sample view towards a shooting region of interest is defined as a local candidate. The view quality of each candidate is computed with the consideration on cinematography aesthetics. Next, we formulate the task of selecting the optimal local view candidate associated with each sketch point and best connecting them as an instance of the Set Traveling Salesman Problem (STSP), a generalized version of the traveling salesman problem. In the final, we propose a constrained optimization process to adjust the flight speed of the drone for further enhancing the stability and quality of resulting aerial videos.

4. 2D-to-6D Camera Trajectory Lifting

4.1. Analyzing 2D Input Sketch

User requires drawing a continuous sketch path on the 2D map. The manual path is often coarse and not smooth. We first smooth the sketch, and then extract its feature points where their curvatures are larger than the threshold. We believe that these feature points can represent the characteristics of the input sketch. Next, we conduct an equidistant sampling in-between these feature points, such that the simplified path is close enough to the original sketch while can largely reduce the computation cost in the following steps. The sampling distance is set as 40m by default. We name these samples on the new simplified path as sketch points.

2.5D coarse scene model. The 2.5D coarse scene data, consisting of 2D footprints associated with architecture heights, is the input to our system. In our implementation, such data was obtained via an interactive map tool together with sparse stereo estimations.

Safe flight zone. For each sketch point on the simplified path, we set the minimal safe height for the drone to 30m by default. To our experience, the height of 30m enables the drone avoid small obstacles such as trees and streetlights. If the sketch point overlaps a region of interest model, the minimal safe height will be the height of model plus 20m. In our experiments, due to the UAV flying control by law, the maximum height of the restricted flight is 120m. All of our computations will be at the safe flight zone between the minimal and the maximum height.

Regions of interest. For each sketch point, we compute the visual interest [HLH^{*}16] of the regions within 150m neighborhood. Then, to reduce the computation, we set by default the top $k = 5$ regions of highest visual interest score as candidate regions.

4.2. Selecting and Connecting Quality Views

Local view candidates. We sample the view point in the vertical zone above each sketch point on the sketch path. These height samples represent the possible locations of the drone. The sampling strategy is sampling the zone every 10m between minimal and maximum safe height which guarantee the drone is collision-free on each sketch point. For each height sample, we compute the view quality toward the center of each candidate region of interest.

One height sample toward one candidate region of interest is represented as one local candidate. For example, after sampling, if a sketch point has n height samples, then this sketch point has no more than $n \times k$ local candidates.

The computation of view quality is based on the most recent work [XYH^{*}18]. For each interest model, we compute its salient features, including the shape silhouettes, sharp edges as well as the skeletal region of the model top surface. For each local candidate x , we render the region model shape using the intrinsic and extrinsic parameters of the mounted camera thus gets a visual interest map, $I(x)$. Then we use a composition weight map similar to [XYH^{*}18], I^w , that assigns higher weights based on proximity to the center and the Rule of Thirds lines. The view quality is computed as the dot product of the visual interest and the weight maps:

$$Q(x) = I(x) \cdot I^w. \quad (1)$$

Given a local candidate x , we compute the view quality score as:

$$E_{\text{local}}(x) = 1 - Q(x). \quad (2)$$

We define the transition path as the line that connects two local candidates. A good transition trajectory should satisfy several requirements. It shall be very smooth and collision-free. We would like the camera motion to be smooth, avoiding fast camera rotation and unnecessary turning. Note that safety is the most important. Below we describe how we attempt to satisfy these requirements.

Transition in-between views. Given a transition path named as $T_{((i,j),(i+1,k))}$ from the i th sketch point's j th local candidate to the $(i+1)$ th sketch point's k th local candidate, we define its associated cost $E_{\text{trans}}(T_{((i,j),(i+1,k))})$ as:

$$E_{\text{trans}}(T_{((i,j),(i+1,k))}) = w_h * E_h + w_p * E_p. \quad (3)$$

The first term E_h measures the heading changing ratio between the local candidate $x_{(i,j)}$ and $x_{(i+1,k)}$:

$$E_h = \frac{|h(x_{(i,j)}) - h(x_{(i+1,k)})|}{d(x_{(i,j)}, x_{(i+1,k)})}, \quad (4)$$

Where $h(x)$ is the heading of the local candidate x , and $d(x_{(i,j)}, x_{(i+1,k)})$ is the distance in-between these two candidates.

The next term H_p penalizes large changes in drone pose:

$$H_p = \exp\left(\frac{|h(x_{(i,j)}) - h(x_{(i+1,k)})| - 120}{360}\right). \quad (5)$$

Empirically, we do not want the drone to change heading over 120° on a transition path, which easily leads to a vertigo in video.

The last term E_p measures the camera pitch changing ratio between the local candidate $x_{(i,j)}$ and $x_{(i+1,k)}$:

$$E_p = \frac{|p(x_{(i,j)}) - p(x_{(i+1,k)})|}{d(x_{(i,j)}, x_{(i+1,k)})}, \quad (6)$$

Where $p(x)$ is the camera pitch of the local candidate x .

Heading and pitch changing ratios are two important factors that affect the drone to shoot a nice aerial video. We set $w_h = 0.8$ and $w_p = 0.2$ by default, biasing the heading influence as the visual stability of aerial videos depends more on heading changing ratio.

Global combinatorial optimization. The trajectory planning problem is hence the combinatorial optimization problem that takes both the cost of the local candidate and the transition among them into the consideration. For each sketch point, we have several local candidates in different altitude towards different candidate regions of interest. From one sketch point to the next, we have several transition paths connecting different pairs of local candidates; see an illustration in Figure 2. We model this combinatorial optimization as a generalized version of the traveling salesman problem, known as the Set Traveling Salesman Problem (STSP).

Given a graph with several disjoint subsets(clusters) of nodes, as well as a set of edges for each pair of adjacent node clusters, the goal of STSP is to find a shortest tour, which visits each cluster

exactly once with the minimal cost on both nodes and edges. The standard TSP may be viewed as a special case of STSP, with only one node per cluster.

In our case, all local candidates of one sketch point form a node cluster in the graph. The edges of the graph correspond to the transition paths. The cost of a node is given by E_{local} in Eq. (2) and the cost of an edge is defined by E_{trans} in Eq. (3). The STSP problem is solved using the Lin-Kernighan-Helsgaun (LKH) algorithm with the software package developed in [Hel15].

While preprocessing the sketch path in the first step, we compute the safe flight zone of each sketch point, in this step, if the transition path has a collision with a region of interest, the cost of the transition path will be set infinite so as to guarantee the drone fly on the safe transition path.

To solve the global optimization, we have also tested other methods, such as Viterbi algorithm, simulated annealing algorithm and genetic algorithm. Both the LKH algorithm and Viterbi algorithm can output the results in real-time. The simulated annealing and genetic algorithms instead need much more iterations to converge, and often fall into local optima. In term of memory space occupancy, the LKH algorithm and Viterbi algorithm consume similarly as well, but in term of the efficiency and scalability, the average computation time of LKH is $0.003ms$ while that of Viterbi is $25.4ms$. Thus, we choose to formulate the problem as STSP and solve it using LKH algorithm in the system.

4.3. Optimizing the Flight Speed

Because most path planning studies are still in simulation environment, and the goals or missions of path planning are different, to our best knowledge, current researches mainly focus on trajectory generation and optimization, the actual flying speed of the drone is generally maintained as constant.

With intensive flyby experiments, we find that to shoot a more visually pleasing aerial video, it is necessary to optimize the drone flight speed. After the previous global planning, the lifted trajectory is feasible and aesthetic. Our goal in this flight speed optimization is then focused on the visual stability and interest. That is, if the local heading or pitch changing ratio is quite large in-between two adjacent transition paths, it is better to let the drone fly slower; and vice versa to let drone fly faster to increase visual content changes while reduce the battery consumption.

The optimization process is then formulated as:

$$\begin{aligned} & \max \sum_i w(i) * v_i^2 - \lambda(v_i - v_{i+1})^2, \\ & w(i) = \log_{1/2} \left(w_h * \frac{|h_i - h_{i-1}|}{d(i,i+1)} + w_p * \frac{|p_i - p_{i+1}|}{d(i,i+1)} \right), \\ & \text{s.t., } 0 \leq v_i \leq 15. \end{aligned} \quad (7)$$

The weighting function $w(i)$ controls the speed to increase or decrease. When the heading or pitch changing ratio is relatively large, the result of the $w(i)$ becomes negative that motivates the objective function to have a smaller v value. Moreover, the log function is applied to penalize more the weights. The last term is a smooth term. On one hand, we hope that the drone can fly at a suitable speed under different heading changing and pitch changing ratio. On the

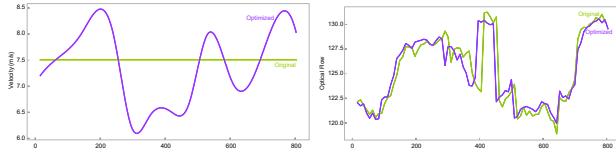


Figure 3: How drone fly: (a) the constant speed input (green) and the output (purple) of speed optimization; (b) comparison of the optical flow before and after speed optimization.

other hand, we hope that the changing of the drone speed is smooth and reasonable when the drone transit between different transition paths. We constrain the speed from 0 to 15m/sec as in practical the maximum speed of the drone we can use is 15m/sec.

Within this formalization, the optimization problem can now be solved using an existing non-linear programming library NLOpt [MKD]. NLOpt is a free/open-source library for non-linear optimization, providing a common interface for a number of different free optimization routines available online as well as original implementations of various other algorithms. Due to the no large scale of our problem, the speed optimization result can be obtained in real-time. The quantitative analysis of speed optimization is shown in Figure 3. Note that since the trajectory result of global combinatorial optimization is already quite good, the optimized optical flow does not deviate much from the original, but still enhance the video stability in a visible way, in particular around local peaks.

5. Results and Evaluation

5.1. Quadrotor Control Platform

The output of our automated flight planning system is a trajectory made up of a series of 6D waypoints with 3D positions, 2D camera orientations and 1D velocities. We developed a software platform based on DJIWaypointMission SDK to control drone to fly and shoot video. With this SDK it is possible to specify a sequence of waypoints. The desired physical locations to which the quadrotor will fly and the camera heading and tilt as well as the velocity of the quadrotor could be specified for each waypoint. The quadrotor will travel between waypoints, execute actions at waypoints, and adjust heading, altitude, pitch and velocity between waypoints. In short, the actual flight trajectory of the quadrotor is basically the same as that generated by our system. The small error between the actual flight trajectory and the planned trajectory might be caused by the influence of GPS signal and wind speed.

The quadrotor we use in this paper is a Phantom 4 Pro V2.0, equipped with a 1-inch 20-megapixel camera capable of shooting 4K/60 fps video. The controllable heading range of the quadrotor is $[-180^\circ, 180^\circ]$ and the tilt range of the camera is $[-90^\circ, 0^\circ]$. We used the platform developed based on DJI to load the trajectory data to the quadrotor. Then the quadrotor will follow the preplanned trajectory and execute preset actions to shoot video.

5.2. Uncut Aerial Videos

We tested our path planning system on four large-scale outdoor scenes. These scenes (Figures 1, 4, 5, and 6) have different archi-

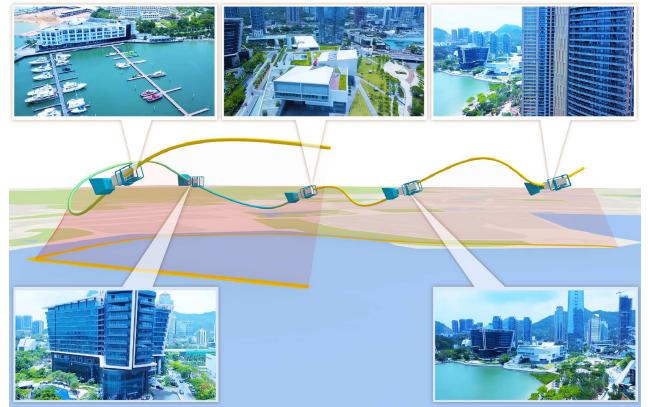


Figure 4: A 6D camera trajectory is automatically generated from a simple 2D sketch. Please see the supplementary video for our beautiful uncut shots by the sea.

tectural layout, and the building styles are also various. Users were asked to observe the scene with 2.5D rough models and draw the coarse sketch path which expresses how they expect the aircraft to tour and catch video in the scene. Users have different ideas of aerial route in different scenes, even in the same scene.

The resulting captured aerial video sequences are included in the supplementary materials, sped up by a factor of x4 or x8 due to the limited size of the uploaded video. Even with the accelerated playback, the aircraft movement and camera motion were quite smooth. In particular cases, the effect of the original and not accelerated video is better than that after acceleration.

Compared to the previous work [XYH^{*}18], we do not have to compute the view quality for every voxel around the landmark, we do it only for every local candidate above the sparse sketch points, which were the output of our preprocessing. This helps us to achieve real-time response (Table 2).

5.3. User Study

To evaluate the effectiveness and the efficiency of our trajectory planning system, we conducted a user study by comparing the quality of aerial videos resulted from our sketch-based tool, the commercial DJI GS Pro [DJI] design tool and the interactive Joubert's tool proposed in [JRT^{*}15]. We chose the university campus as the scene for user study, as the layout of the campus is relatively scattered so that it is easy to design different paths.

Six users were asked to draw a path using our sketch tool (ST). Among the six users, two users are relatively experienced drone pilots, and the other novice users were trained to use the tools. The user sketches and the corresponding lifted trajectories are shown in Figure 6. Then, they were asked to design a similar path using DJI GS Pro (DGS) tool, including to designate the waypoints and set up the altitude, heading, pitch of the camera at each waypoint. Next, they were instructed to design another similar path using Joubert's tool (JT) by designating the key-frames in a virtual 3D environment that we built. At last, we shoot the aerial videos along the trajectories generated by these three tools.



Figure 5: Two very different camera trajectories automatically generated from two different sketches drew at the same region around City Bay. Please see the supplementary video.

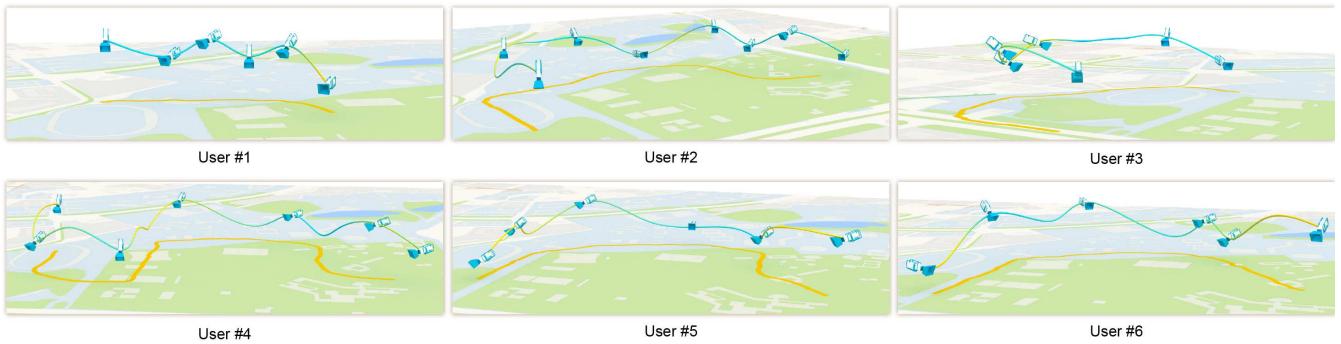


Figure 6: Various camera moves generated from six sketches drew by six different users for the same university scene.

Our hypotheses are: the videos created by our tool are more pleasing to watch (H1), provide a clearer overview of the scene (H2), better satisfy the expectations of the user (H3), than others. We have 6 video groups from 6 users, each group consisting of three video versions generated by our ST, DGS and JT. To assess these hypotheses, 30 users (including the previous six users) were recruited to watch 12 side-by-side video comparisons: ST v.s. DGS and ST v.s. JT, respectively. We also showed the users the corresponding input information (i.e., the sketches of ST, the waypoints of DGS and the key-frames of JT) to help them make the assessments. The videos can be watched as many times as they want, and then they were required to answer 3 questions on a 5-point Likert scale. Each video is about 1-2mins, and the total average experiment time for a user was around 30 mins.

The questions were: (Q1) the left video was more pleasing to watch than the right one; (Q2) the left video provided a clearer overview of the scene we want to explore than the right one; (Q3) the left video showed a more reasonable route and better satisfied your expectation than the right one. These three questions were on a 5 point Likert scale ranging from "-2 (completely disagree)" to "2 (completely agree)", displaying "0 (neutral)" in the middle.

Figure 7 and Table 1 illustrate the results of user assessments for

hypotheses H1 to H3, in comparing our ST v.s. DGS, and ST v.s. JT. Stacked bar charts and statistical values demonstrate that users favor more the videos generated by our tool than those by using DGS and JT. From the user feedback, the videos generated by DGS often failed to be watched pleasingly. It seems because the users are hard to estimate the preview when they designate the waypoints through 2D maps. The videos generated by JT are better as the users can check the shoot preview in a 3D virtual environment before actual drone fly. Nonetheless, the transits between key-frames were sometimes boring due to the simple spline interpolation. The performance of our tool is more comprehensive due to the global trajectory planning and the flight speed optimization.

In order to better illustrate the priority of our approach, we recorded additional quantitative statistics when using our and other tools. The results are shown in Tables 2 and 3. It can be seen that the sketching and trajectory lifting time are very short. The time consumed by using DGS is mainly due to the need for users to set up drone configuration at each waypoint. The time consumed by using JT is mainly due to the difficulty of specifying key frames in a 3D virtual environment. From the comparison of battery consumption, we can tell that both ST and JT save the energy significantly, comparing with using DGS, where most of users have to modify the path design and then re-fly the drone couple of times until satisfied.

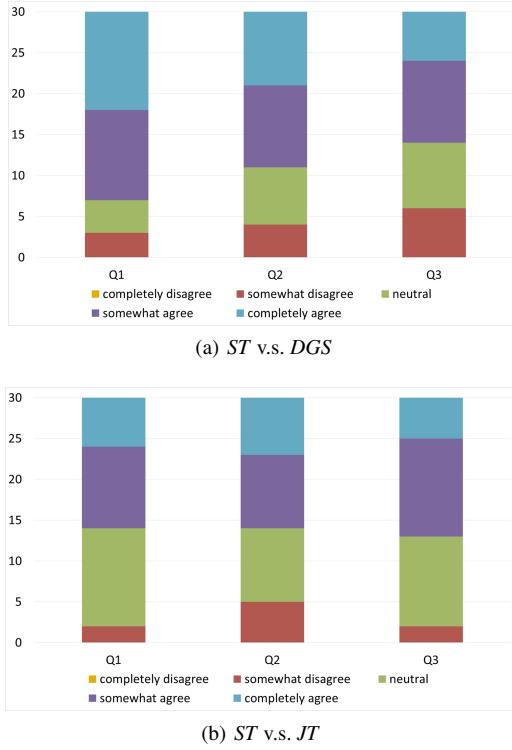


Figure 7: Stacked bar visualizations of the user evaluation comparing: (a) our ST v.s. DGS, and (b) our ST v.s. JT. The charts report for each question the aggregated number of answers. These results demonstrate the superiority of our tool.

Table 1: Statistical results of the user evaluation comparing: (a) our ST v.s. DGS, and (b) our ST v.s. JT. The mean and variance values of each comparative question illustrate that in general users have relatively higher assessments on our sketch tool.

Question	ST v.s. DGS		ST v.s. JT	
	Mean	Variance	Mean	Variance
Q1	1.06	0.96	0.70	0.83
Q2	0.80	1.06	0.60	1.07
Q3	0.60	1.14	0.66	0.71

6. Conclusion and Future Work

We present a system that attempts to follow current popular composition principles and automatically generates a feasible and smooth trajectory for shooting a no-clipping aerial video. At the same time, the input based on the 2D rough sketch path enables users to express their ideas and creativity in the path planning task. Different from the most recent related work [XYH*18], with the help of user's sketch, we reduce the searching space to zone above the sketch points. This improvement of time and memory efficiency makes the system respond to user input in real-time.

We model the trajectory planning problem as a combinatorial optimization problem of the selection of the local candidate that takes the cost of both local candidate and transition path into con-

Table 2: Processing time comparison among our ST, DGS and JT. The time cost of ST is the sum required for user sketching and 6D trajectory computation. The time cost of DGS is what the user takes to set up the entire path. The time cost of JT is the sum required for user designating the key-frames and trajectory interpolation.

User	Time Cost		
	ST	DGS	JT
#1	2s+4s	20m	10m+1s
#2	3s+4s	40m	11m+1s
#3	1s+2s	15m	8m+1s
#4	1s+2s	10m	5m+1s
#5	2s+4s	30m	10m+1s
#6	2s+4s	32m	12m+1s

Table 3: Drone battery consumption among our ST, DGS and JT. The number in parentheses means how many times the user tried to rectify the path and re-shoot the video, while 100% of the battery means that one fully charged drone battery was used up.

User	Battery Consumption		
	ST	DGS	JT
#1	32% (1)	120% (2)	32% (1)
#2	38% (1)	300% (5)	37% (1)
#3	18% (1)	144% (3)	20% (1)
#4	12% (1)	130% (3)	13% (1)
#5	40% (1)	160% (3)	39% (1)
#6	35% (1)	175% (4)	30% (1)

sideration. We solve this problem by solving the STSP based on a generalized TSP solver. A non-linear problem is then formalized to optimize the speed of the drone on each waypoint which aims to make the final aerial video more stable. We have demonstrated our approach not only in virtual city scenes, but also in real world.

While simplifying the user's input, our system allows user to express their ideas and creativity with a rough sketch path they painted. But in experiments, we find that some users can not grasp the distance from the region of interest they want to shoot in 2D map. For example, some users may want to take a far distant view from a region of interest. However, because the volume of region is huge, the size of the region in screen is too large in actual flight. They need to draw a bit farther to get a far distant view of the region. In future, we plan to enable users to interactively grasp the sense of distance or automatically estimate the distance between the path and region of interests according to the user request.

In speed optimization, we control the speed of the drone based on the heading changing ratio and pitch changing ratio. However, due to the limited opening of the DJI API, we did not involve the study of drone power system. In actual flight, the speed of the drone may appear slight errors on some waypoints. In the future work, we are interested in extending our knowledge on the field of drone power system. And we plan to use a lower level drone with more comprehensive control program that would enable the drone follow a planned trajectory in a more precise manner.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. This work was supported in parts by NSFC (61522213, 61761146002, 61861130365), China Postdoc Foundation (2017M622780), 973 Program (2015CB352501), Guangdong Science and Technology Program (2015A030312015) and Shenzhen Innovation Program (KQJSCX20170727101233642).

References

- [Bar06] BARES W.: A photographic composition assistant for intelligent virtual 3D camera systems. In *Proc. of Int. Symp. on Smart Graphics* (2006), pp. 172–183. [2](#)
- [BDGER08] BURELLI P., DI GASPERO L., ERMETICI A., RANON R.: Virtual camera composition with particle swarm optimization. In *Proc. of Int. Symp. on Smart Graphics* (2008), pp. 130–141. [2](#)
- [BGL98] BARES W. H., GRÉGOIRE J. P., LESTER J. C.: Realtime constraint-based cinematography for complex interactive 3D worlds. In *Proc. of AAAI/IAAI* (1998), pp. 1101–1106. [2](#)
- [BHC08] BOUKTIR Y., HADDAD M., CHETTIBI T.: Trajectory planning for a quadrotor helicopter. In *Proc. of Mediterranean Conf. on Control and Automation* (2008), pp. 1258–1263. [3](#)
- [BP14] BURELLI P., PREUSS M.: Automatic camera control: a dynamic multi-objective perspective. In *Proc. of European Conf. on the Applications of Evolutionary Computation* (2014), pp. 361–373. [2](#)
- [DG08] DI GASPERO L., ERMETICI A., RANON R.: Swarming in a virtual world: A pso approach to virtual camera composition. In *Proc. of Int. Conf. on Ant Colony Optimization and Swarm Intelligence* (2008), pp. 155–166. [2](#)
- [DJI] DJI: Dji ground station pro. <https://www.dji.com/ground-station-pro>. [2, 6](#)
- [DT15] DEITS R., TEDRAKE R.: Efficient mixed-integer planning for UAVs in cluttered environments. In *Proc. of IEEE Int. Conf. on Robotics & Automation* (2015), pp. 42–49. [3](#)
- [FGTG16] FLEUREAU J., GALVANE Q., TARIOLLE F.-L., GUILLOTEL P.: Generic drone control platform for autonomous capture of cinema scenes. In *Proc. of Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use* (2016), pp. 35–40. [3](#)
- [GHN*16] GEBHARDT C., HEPP B., NÄGELI T., STEVŠIĆ S., HILLIGES O.: Airways: Optimization-based planning of quadrotor trajectories according to high-level user goals. In *Proc. of CHI Conf. on Human Factors in Computing Systems* (2016), pp. 2508–2519. [4](#)
- [GLC*18] GALVANE Q., LINO C., CHRISTIE M., FLEUREAU J., SERVANT F., TARIOLLE F., GUILLOTEL P., ET AL.: Directing cinematicographic drones. *ACM Trans. on Graphics* 37, 3 (2018), 34:1–34:18. [4](#)
- [GSH18] GEBHARDT C., STEVŠIĆ S., HILLIGES O.: Optimizing for aesthetically pleasing quadrotor camera motion. *ACM Trans. on Graphics* 37, 4 (2018), 90:1–90:11. [3](#)
- [Hel15] HELSGAUN K.: Solving the equality generalized traveling salesman problem using the Lin–Kernighan–Helsgaun algorithm. *Mathematical Programming Computation* 7, 3 (2015), 269–287. [5](#)
- [HGP*18] HUANG C., GAO F., PAN J., YANG Z., QIU W., CHEN P., YANG X., SHEN S., CHENG K.-T. T.: Act: An autonomous drone cinematography system for action scenes. In *Proc. of IEEE Int. Conf. on Robotics & Automation* (2018). [2](#)
- [HLH*16] HUANG H., LISCHINSKI D., HAO Z., GONG M., CHRISTIE M., COHEN-OR D.: Trip synopsis: 60km in 60sec. *Computer Graphics Forum (Proc. of Pacific Conf. on Computer Graphics & Applications)* 35, 7 (2016), 107–116. [4](#)
- [HO00] HALPER N., OLIVIER P.: Camplan: A camera planning agent. In *Proc. of Int. Symp. on Smart Graphics* (2000), pp. 92–100. [2](#)
- [HYK*18] HUANG C., YANG Z., KONG Y., CHEN P., YANG X., CHENG K.-T. T.: Through-the-lens drone filming. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (2018). [3](#)
- [JGB*16] JOUBERT N., GOLDMAN D. B., BERTHOUZOZ F., ROBERTS M., LANDAY J. A., HANRAHAN P., ET AL.: Towards a drone cinematographer: Guiding quadrotor cameras using visual composition principles. *arXiv preprint arXiv:1610.01691* (2016). [2](#)
- [JRT*15] JOUBERT N., ROBERTS M., TRUONG A., BERTHOUZOZ F., HANRAHAN P.: An interactive tool for designing quadrotor camera shots. *ACM Trans. on Graphics* 34, 6 (2015), 238:1–238:11. [4, 6](#)
- [KM02] KENNEDY K., MERCER R. E.: Planning animation cinematography and shot structure to communicate theme and mood. In *Proc. of Int. Symp. on Smart Graphics* (2002), pp. 1–8. [2](#)
- [LC15] LINO C., CHRISTIE M.: Intuitive and efficient camera control with the toric space. *ACM Trans. on Graphics* 34, 4 (2015), 82:1–82:12. [4](#)
- [LSHZ17] LAN Z., SHRIDHAR M., HSU D., ZHAO S.: Xpose: Reinventing user interaction with flying cameras. In *Proc. of Robotics: Science and Systems* (2017), vol. 13. [3](#)
- [LYW06] LAI L.-C., YANG C.-C., WU C.-J.: Time-optimal control of a hovering quad-rotor helicopter. *Journal of Intelligent and Robotic Systems* 45, 2 (2006), 115–135. [3](#)
- [MkD] MkDOCS: Nlopt:free/open-source library for nonlinear optimization. <https://nlopt.readthedocs.io/en/latest/>. [6](#)
- [NAMD*17] NÄGELI T., ALONSO-MORA J., DOMAHIDI A., RUS D., HILLIGES O.: Real-time motion planning for aerial videography with dynamic obstacle avoidance and viewpoint optimization. *IEEE Robotics and Automation Letters* 2, 3 (2017), 1696–1703. [2](#)
- [NMD*17] NÄGELI T., MEIER L., DOMAHIDI A., ALONSO-MORA J., HILLIGES O.: Real-time planning for automated multi-view drone cinematography. *ACM Trans. on Graphics* 36, 4 (2017), 132:1–132:10. [2](#)
- [Nor07] NORMAND J.-M.: Computer graphics and constraint solving: An application to virtual camera control. *Trends in Constraint Programming* (2007), 395–404. [2](#)
- [OSTG09] OSKAM T., SUMNER R. W., THUERÉY N., GROSS M.: Visibility transition planning for dynamic camera control. In *Proc. of ACM SIGGRAPH/Eurographics Symp. on Computer Animation* (2009), pp. 55–65. [2](#)
- [PHaU16] PRIMA D. A., HARIADI M., AND TSUYOSHI USAGAWA K. E. P.: Virtual camera movement with particle swarm optimization and local regression. *International Review on Computers and Software* 11, 9 (2016), 773–793. [2](#)
- [RH16] ROBERTS M., HANRAHAN P.: Generating dynamically feasible trajectories for quadrotor cameras. *ACM Trans. on Graphics* 35, 4 (2016), 61:1–61:11. [3](#)
- [SL17] SHEN H., LIU R.: Study on virtual camera with preset shot types based on composition aesthetic computing. In *Proc. of Transactions on Edutainment XIII* (2017), pp. 29–37. [2](#)
- [SP08] SOKOLOV D., PLEMENOS D.: Virtual world explorations by using topological and semantic knowledge. *The Visual Computer* 24, 3 (2008), 173–185. [3](#)
- [XYH*18] XIE K., YANG H., HUANG S., LISCHINSKI D., CHRISTIE M., XU K., GONG M., COHEN-OR D., HUANG H.: Creating and chaining camera moves for quadrotor videography. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 37, 4 (2018), 88:1–88:13. [2, 3, 4, 6, 8](#)
- [ZTD05] ZHANG Y.-B., TONG R.-F., DONG J.-X.: Octree-based camera planning in virtual environments. In *Proc. of Int. Conf. on Computer Supported Cooperative Work in Design* (2005), vol. 2, pp. 767–771. [2](#)