2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

ACT: An Autonomous Drone Cinematography System for Action Scenes

Chong Huang[1], Fei Gao[2], Jie Pan[2], Zhenyu Yang[1], Weihao Qiu[1], Peng Chen[3], Xin Yang[4],
Shaojie Shen[2], and Kwang-Ting (Tim) Cheng[2]

Abstract— **Drones are enabling new forms of cinematography. Aerial filming via drones in action scenes is difficult because it requires users to understand the dynamic scenarios and operate the drone and camera simultaneously. Existing systems allow the user to manually specify the shots and guide the drone to capture footage, while none of them employ aesthetic objectives to automate aerial filming in action scenes. Meanwhile, these drone cinematography systems depend on the external motion capture systems to perceive the human action, which is limted to the indoor environment. In this paper, we propose an Autonomous CinemaTography system "ACT" on the drone platform to address the above the challenges. To our knowledge, this is the first drone camera system which can autonomously capture cinematic shots of action scenes based on limb movements in both indoor and outdoor environments. Our system includes the following novelties. First, we propose an efficient method to extract 3D skeleton points via a stereo camera. Second, we design a real-time dynamical camera planning strategy that fulfills the aesthetic objectives for filming and respects the physical limits of a drone. At the system level, we integrate cameras and GPUs into the limited space of a drone and demonstrate the feasibility of running the entire cinematography system onboard in real-time. Experimental results in both simulation and real-world scenarios demonstrate that our cinematography system "ACT" can capture more expressive video footage of human action than that of a state-of-the-art drone camera system.**

I. INTRODUCTION

The emergence of drones has raised the bar for cinematic quality and visual storytelling for filmmakers. Compared with conventional camera carriers (e.g., tripods, trucks and cranes), drones benefit from their high mobility to capture more cinematic shots with continuously varying viewpoints. However, it is very challenging to capture high-quality footage of human actions because it requires several professional techniques: accurate comprehension of dynamic scenarios, practiced skill of flying, artistic skill of composition, and simultaneous realization of all the requirements in real time. Some commercial products (e.g., DJI Mavic) have been developed to provide automatic or semi-automatic functions to simplify aerial filming. Typically, these systems consider a subject as a 2D point on an image plane and adjust the camera to place the subject at the image center. Without analyzing limb motions, these systems cannot adaptively provide a suitable viewpoint for different human actions.

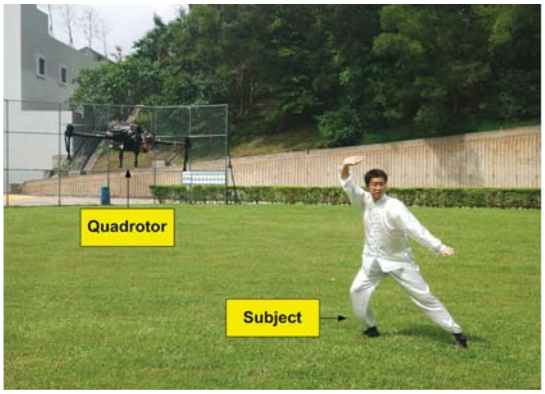


Fig. 1. Autonoumous filming human action of the proposed ACT system.

Some studies [1] [2] [3] introduced cinematography concepts to control a camera drone. They utilized external GPS sensors to help perceive the dynamic environments. For instance, Joubert et al. [1] localized a subject with a portable GPS remote sensor. Given the GPS position of the subject, the user can customize the shooting viewpoints and camera motion. Nageli et al. [2] [3] utilized the Vicon system to obtain accurate 3D poses of subjects based on which users can specify the shots to plan quadrotor motion automatically. These techniques simplified the camera control in specific scenarios but do not consider any visual aesthetic objectives to automate the camera drone control. As a result, the quality of the footage highly relies on the user's input.

Viewpoint selection for human actions has been studied extensively in computer graphics. Metrics have been proposed to evaluate viewpoint quality, such as subject's visibility [4] [5], shape saliency [6] and motion area [7] based on which the system can guide the camera placement in a virtual environment. However, there exist several challenges which prohibit direct usage of existing metrics in a drone cinematography system.

1) Pose Estimation: The 3D pose information of characters in a virtual environment can be easily obtained from graphics software, while human pose in real scenes without any constraints are highly difficult to obtain. Kinect and Vicon

[1]Chong Huang, Zhenyu Yang and Weihao Qiu are with Department of Electrical and Computer Engineering, University of California, Santa Barbara, Santa Barbara, CA 93106 USA. (chonghuang@umail.ucsb.edu, zhenyuyang@umail.ucsb.edu, wqiu@umail.ucsb.edu)

[2]Fei Gao, Jie Pan, Shaojie Shen and Kwang-Ting (Tim) Cheng are with Department of Electrical and Computer Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. (fgaoaa@connect.ust.hk, jpanai@connect.ust.hk, eeshaojie@ust.hk, timcheng@usk.hk)

[3]Peng Chen is with the College of Information and Engineering, Zhejiang University of Technology, Hangzhou 310023 China. (chenpeng@zjut.edu.cn)

[4]Xin Yang is with the School of Electronics Information and Communications, Huazhong University of Science and Technology, Wuhan, Hubei 430074 China. (xinyang2014@hust.edu.cn)

978-1-5386-3081-5/18/$31.00 ©2018 IEEE

systems can capture accurate motion data, but they are restricted to the indoor environment due to the reliance on infrared sensors. GPS sensors work in the outdoor environment but it only provides a 2D position with low accuracy.

2) Camera Planning: Given the entire sequence of character motion, camera planning is typically formulated as an offline optimization problem which generates a timed reference trajectory from the user-specified 3D positions. However, human action in the real scenarios is unknown, and hence it is infeasible to calculate a global optimum from the conventional optimization approaches. In addition, virtual environments are not limited by real-world physics and robot dynamic constraints; thus, arbitrary camera trajectories, velocities and viewpoints can be generated. In contrast, our system must respect the physical limits of a real quadrotor.

To address the above challenges, our autonomous drone cinematography system includes the following techniques:

1. For pose estimation, we combine stereo-based depth estimation and 2D body skeleton detection to estimate the 3D skeleton pose, and we refine the pose information based on the temporal properties of body movement.

2. For camera planning, we propose a real-time dynamically trajectory generator to guide the camera control for unknown body movements. The generated trajectory can balance aesthetic objectives and the physical limits of real robots.

To achieve real-time onboard pose estimation and camera planning, we mount a stereo camera and two GPUs on a DJI Matrix 100 drone. We use a gimbal RGB camera to capture the stabilized footage.

In summary, our contributions are two-fold. First, we propose an efficient 3D skeleton detection method based on a stereo camera and a real-time camera planning algorithm that can balance the aesthetic objective and physical limits. The system can be used in both indoor and outdoor environments. Second, we implement the entire system on the limited computation resource of the drone platform, including skeleton detection, viewpoint estimation, trajectory planning and localization, and demonstrate its feasibility of running the system in real-time (see Fig. 1).

We discuss related work in Sec. II, and introduce our 3D skeleton detection based on stereo camera in Sec. III. The camera planning based on next-best-view is presented in Sec. IV. The system architecture is detailed in Sec. V. In Sec. VI, we present simulation and experimental results in real-world scenarios. Finally, we give the conclusion and future work in Sec. VII.

II. Related Work

Autonomous aerial filming: With the development of vision-guided flight control in aerial filming, researchers have combined cinematography concepts and characters' pose information to capture more professional shots. Joubert et al. [1] utilize the visual composition principle to guide the camera control. Although the system has been successfully used to film a range of activities, such as taking a selfie, the subjects are stationary and the camera control does not respond to the limbs movements. Nageli et al. [2] [3] represent the body motion as a set of 3D markers, and allow users to specify the subject size, viewing angle and position on the screen to generate quadrotor motion plans automatically. These techniques simplify the camera control in specific scenarios and do not consider any visual aesthetic objectives to automate the camera drone control. As a result, the quality of the footage highly relies on the user's input.

3D skeleton detection: Existing 3D skeleton detection methods rely on infrared-based depth sensors. The Kinect sensor is an easy-to-operate device for depth detection. The Kinect can track multiple subjects without requiring users to wear extra sensors. Benefited from its compact size, the Kinect sensor [8] can be mounted on the robot to perceive unknown environments. However, the Kinect sensor calculates depth with a infrared laser projector, so it cannot work in the outdoor environment. Vicon is another widely-used system in the field of motion analysis because of its high accuracy of pose tracking. However, it is also restricted in the indoor environment because of its optical properties. Meanwhile, its immobility only track the subjects within a limited space.

Camera planning in computer graphics: Camera planning for human action has been widely studied in computer graphics. Researchers focus on searching for a set of suitable camera configurations for capturing an expressive video clip, while obeying a set of cinematographic rules [9], as well as other constraints such as occlusion [10], objects visibility [11] [12], layout in the resulting image [13], frame coherency [9] and orientations [14]. There are several metrics to quantify the aesthetic quality of the video clips such as subject's visibility [4] [5], shape saliency [6] and motion area [7]. Using these attributes, the system measures the quality of each frame taken from different viewpoint and outputs the best view. Camera planning is typically formulated as an offline optimization problem which seeks a camera path in space-time 4D space by balancing the viewpoint quality and smoothness of the camera path. In addition, virtual environments are not limited by real-world physics and robot constraints and hence can produce arbitrary camera trajectories, velocities and viewpoints.

III. 3D Skeleton Detection based on Stereo Camera

In this section, we introduce 3D skeleton detection. Our intuition is to recover the depth of the 2D skeleton position from the depth map. We do not consider the active depth sensor (e.g., Kinect) because it cannot work in outdoor environments. Instead, we use stereo cameras to calculate depth. However, the stereo-based depth estimation may generate inaccurate depth for the region with motion blurs. We add some constraints to refine the result. Details are introduced as follows:

A. Raw Depth Acquisition

We utilize a stereo camera to calculate depth based on semi-global block-matching (SGBM) [15] as Fig. 2 (b) shows. The rectified image stream from left camera feeds to a opensource library OpenPose [16] to detect 2D skeleton points. If the full body parts are visible, OpenPose can detect 13 keypoints, including the head, nose, hip, left and right shoulders, elbows, hands, knees and feet (see Fig. 2 (a)), which are used to represent the full human pose. Given a depth map based on left camera, we can convert each 2D body keypoint (x, y) on the image plane to a 3D body keypoint (X, Y, Z) as Eq. 1. The 3D body keypoints are connected as 3D skeleton (see Fig. 2(c)).

$$\begin{aligned} Z &= depth(x,y), \\ X &= (x - c_x) \cdot Z/f_x, \\ Y &= (y - c_y) \cdot Z/f_y, \end{aligned} \tag{1}$$

where c_x, c_y are the center of the image and f_x, f_y are the focal length of the camera on both axes.

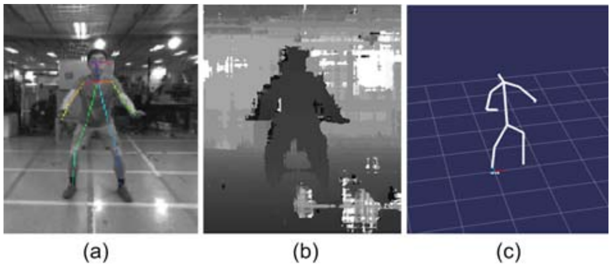


Fig. 2. (a) 2D skeleton (b) depth map, and (c) 3D skeleton

B. Skeleton Refinement

As mentioned before, the noise of the depth map may affect the recovered 3D pose, especially moving subjects. We will refine the 3D pose based on the temporal consistency of human action.

Assuming that the movement of the body joints is smooth, we can use polynomial regression to parameterize the trajectory of each keypoint in terms of time. The predictor resulting from polynomial expansion can determine whether the current pose estimated from the depth map is trustworthy. Given a set of trustworthy poses [s_0, s_1,...,s_N], we set the following optimization function to solve the polynomial coefficients:

$$\begin{aligned} &\min_{\{a_n\, a_{n-1} \cdots a_0\}} \sum_{i=0}^{N} (\bar{s}_i - s_i)^2 \\ &\text{subject to} \quad \bar{s}_i = a_n t_i^n + a_n t_i^n + \cdots + a_0, \end{aligned} \tag{2}$$

where $\bar{s}_i$ is the pose to be modeled at the ith frame and t_i is the timestamp of the ith frame. N is the number of training frames and n is the order of the polynomial function. Considering that the acceleration of the body movement respects the physical limits of limb, we add a penalty term to control the acceleration along the trajectory as follows:

$$\min_{\{a_n\, a_{n-1} \cdots a_0\}} \sum_{i=0}^{N} (\bar{s}_i - s_i)^2 + w \int_0^T (\ddot{s})^2 dt, \tag{3}$$

where w is the penalty weight and is set as 200, and T is the time taken during N frames.

However, the actual limb movement is complex, and the accuracy of polynomial fitting is related to the pace and speed of human action. We discuss the selection of the parameter n and N in four scenarios from CMU Motion Capture Dataset: 1) TaiChi, 2) Walk, 3) Ballet dance, and 4) Run. Each clip of data takes 10 seconds, and the 3D skeleton of the entire sequence is known. We use the pose history in the past N frames to predict the current pose. We evaluate the performance by the average limb distance between the predicted and actual current pose.

Fig. 3 shows the distribution of the prediction error in different human actions. TaiChi and Walk have smooth limb motion and our polynomial function can achieve highly accurate prediction. In contrast, Run and Ballet Dance have fast limb motion with various paces. The limb movements are more difficult to predict, because they require a shorter time window to model the rapidly changing limb movement. The higher order polynomial function cannot improve the prediction accuracy. Therefore, we set the N and n as 15 and 5, respectively in our polynomial fitting.

In our system, we apply a simple voter to refine the 3D pose. If the distance between the pose estimated from the depth map and the modeled pose is larger than 0.5m, we set the modeled pose as refined pose. Otherwise, pose estimated from the depth map is set as refined pose. The refined pose is trustworthy and will be used to predict the future pose.

IV. Camera Planning based on Next-Best-View

In this section, we introduce camera planning. The goal is to design a trajectory that fulfills the aesthetic objectives and respects the physical limits of the real drone. First, we predict the human pose in the next frame by using predictor in Sec. III.A, and then calculate the best viewpoint of this pose, and then we generate the physically feasible trajectory that points to this viewpoint. Details on viewpoint selection and trajectory planning follow.

A. Viewpoint Selection

The viewing space is a subject-center sphere for each human pose. We estimate the best viewpoint in terms of the radius and orientation angle.

The radius, the camera-to-subject distance, determines the size of the subject on the image. On the one hand, we need to ensure a smooth displacement of the subject on the image plane. Vzquez et al. [17] proposes to increase the camera-to-subject distance if the subject moves fast, and vice versa. On the other hand, we must keep a safe distance between the camera and the subject to avoid collision. The radius is estimated as follows:

$$r = r_0(1 + kv), \tag{4}$$

where r_0 is the minimum camera distance and k is a constant parameter to adjust the camera distance. v is the subjects current speed, which is represented by the average speed

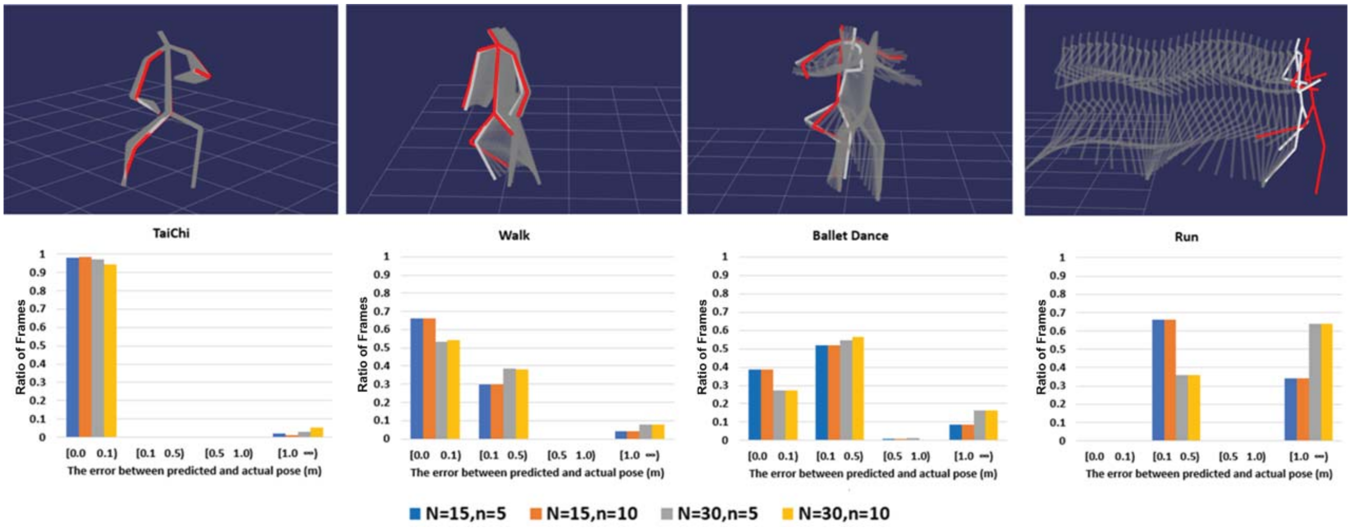


Fig. 3. First row: actual current pose (white), previous pose (gray) and predicted current pose (red). The red pose is predicted from the previous 30 poses within one second. The gap between actual and predicted pose becomes larger as the faster movement pace. Second row: The distribution of prediction error in different human actions. The vertical axis refers to the ratio of frames with the specific prediction error in the whole video sequence, and the horizonal axis is the reconstruciton error between predicted and actual poses. N and n are the size of temporal window and the order of polynomial function, respectively.

of the neck and hip keypoints. To gurantee a view of the subject's whole body, we set minimum distance r_0 as 3m. We set k as 0.4 to keep smooth and stable camera movement.

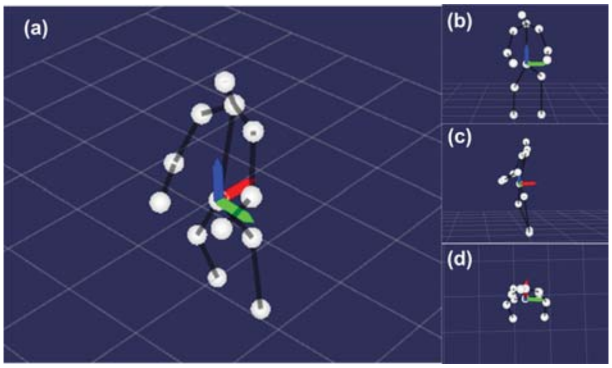


Fig. 4. (a) The 3D skeleton points and its three eigenvectors corresponding to the eigenvalues in the descending order (BGR). The right images illustrate the camera view from (b) the third eigenvector, (c) the second eigenvector and (d) the first eigenvector. It is obvious that the camea view from the third eigenvector displays the maximum projection of point cloud.

The orientation angle defines the pitch and yaw of the camera relative to the subject, which determines the visible part of the subject on the image. There are several ways [4] [5] [6] [7] to measure the quality of a view of a subject in computer graphics. The objective function that measures this is called a view descriptor, and the best view is that which maximizes this function. The view descriptor in Assa et al. [4] [5] measures the visiblity of the joint points of a character to quantify viewpoint quality. Because we represent human pose as 13 3D skeleton keypoints, we evaluate each frame with this metric. First, we calculate PCA for 3D skeleton keypoints to get three eigenvalues (λ_1, λ_2, λ_3) in descending order. The eigenvector corresponding to the minimum eigenvalue is set as the best view angle, because it is perpendicular to the plane with the largest projection of the point cloud as Fig. 4 shows.

$$\theta = \{\theta | P\theta = \lambda_3 \theta\}, \tag{5}$$

where P is a matrix consisted of 3D skeleton points and λ_3 is the minimum eigenvalue. The best viewing direction is along with the eigenvector θ pointing to the subject's center.

It is noted that not all the viewpoints with maximum projection are feasible. First, if the projections of the subject from different viewpoints are similar, subtle motion in the consecutive frames also cause "viewpoint jumping" (see Fig. 5). "Viewpoint jump" will cause a sudden change of acceleration during trajectory planning. This not only increases the instability of the flight control but also makes the footage unpleasing, so we do not move the camera in this case. Considering that eigenvalues is inversely proportional to the variation of the projection to which corresponding eigenvector is perpendicular, we can compare eigenvalues to evaluate the distinctiveness of different viewpoints. Therefore, we define Eq. 6 to estimate the probability of "viewpint jumping". If it is smaller than a threshold, it is likely to be "viewpoint jumping" and we skip this viewpoint. The threshold is set as 1.5.

$$\epsilon = \lambda_2 / \lambda_3. \tag{6}$$

Second, because the stereo camera is fixed on the front of the drone, the observatoin range is determined by the drone's pose. If the viewing direction calculated by our algorithm can generate too high- or too low-angle shots, the stereo camera fails to track the subject within the field of view. To prevent this case, we set the maximum and minimum pitch angle of viewpoint as 15 and -45 degrees. In addition, we set the minimum flight height as 0.3 m to avoid colliding ground.

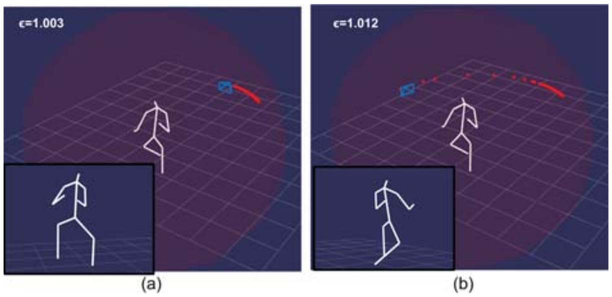


Fig. 5. Viewpoint jumping. The world view and camera view (subfigure on the left-bottom) are shown at (a) T_0 and (b) $T_0 + 0.3s$. The red points are the best viewpoint for each frame and the blue camera is the current camera pose. Because the viewpoint quality of both camera views is very similar (ϵ is close to 1), subtle limb movement causes switch between second and third eigenvectors. The dramatic change of best viewpoint makes it infeasible to move the camera within short time in the real scenarios.

B. Trajectory Planning

Although our system shares the same view descriptor as [4] [5], there are several distinct differences in the constraints of the camera control. First, we cannot formulate the camera planning as an offline optimization for unknown human movement. Second, the flight control must respect its physical limists. Third, the drone must keep a safety distance with the subject. In this section, we present our optimization-based trajectory generation method under the above constraints.

In our system, we re-plan a trajectory for each frame in real-time. Each trajectory is calculated based on current camera pose and next waypoint. We model the trajectory as one-piece polynomial, which is parameterized to the time variable t in each dimension x, y, z, yaw. The trajectory of one dimension can be written as follows:

$$f_\mu(t) = \sum_{j=0}^{n} p_j t^j \quad t \in [0, T], \tag{7}$$

where p_j is the jth order polynomial coefficient of the trajectory, and T is total time of the trajectory, which is calculated by the segement length, maximum velocity and acceleration based on trapezoidal acceleration profile [18]. The polynomial coefficients are computed by minimizing the integral of the square of the k^{th} derivative along the trajectory. In this paper, we minimize the snap along the trajectory, so k is 4. Instead of formulating the cost function for each dimension as in [19], in this paper, the coefficients in all x, y, z, yaw dimensions are coupled into one single equation:

$$J = \sum_{\mu \in \{x,y,z,yaw\}} \int_0^T \left(\frac{d^k f_\mu(t)}{dt^k} \right)^2 dt. \tag{8}$$

The objective function can be written in a quadratic formulation $p^T Q p$, where p is a vector containing all polynomial coefficients in all four dimensions of x, y, z, yaw and Q is the Hessian matrix of the objective function.

We must define the following constraints to ensure the feasibility of the trajectory:

1) *Waypoint Constraints*: If there exists a waypoint at the time of T, we have

$$f_\mu(T) = d_T. \tag{9}$$

2) *Continuity Constraints*: The trajecotry must be continuous at all the k^{th} derivatives at each waypoint between two polynomial segments:

$$\lim_{x \to T^-} f_\mu^{(k)}(T) = \lim_{x \to T^+} f_\mu^{(k)}(T). \tag{10}$$

The both constraints can be compiled into a set of linear equality constraints ($Ap = d$) in [20]. Thus, the trajectory generation problem can be reformulated as a quadratic programming problem:

$$\begin{aligned} &\min \quad p^T Q p \\ &\text{subject to} \quad Ap = d. \end{aligned} \tag{11}$$

In practice, we need to check maximum velocity and acceleration of the trajectory to ensure dynamical feasibility. If the acceleration or velocity of trajectory exceeds the maximum value, we extend the flight time T and recalculate Eq. 11 to get a new trajectory. Then check the feasibility of the trajectory until that it meets the requirement. For simplification, we only check the trajectory at most five iterations and extend the time T by 1.2 times each iteration. The maximum acceleration and velocity is set as $2.5 m/s^2$ and $1.5 m/s$. If the trajectory is still infeasible after five iterations, we do not move the camera. In most cases, we can solve a feasible trajectory at most two iterations.

In addition, although we have limited the minimum distance of each waypoint, the distance between subject and generated trajectory is likely to be less than safety distance. For safety, we define a sphere centered at the subject with radius r_s as a safety region. If the generated trajectory intersects with safety region, we skip this waypoint and do not move the camera. The radius r_s is set as 2m.

V. System Architecture

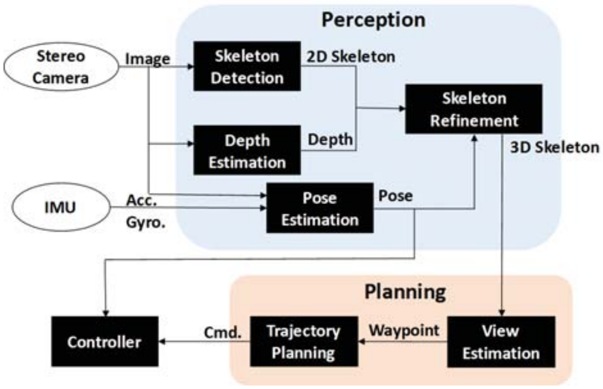


Fig. 6. The Architecture of the System

The system architecture is shown in Fig. 6. In the perception module, we estimate 3D skeleton points by fusing stereo depth and monocular 2D Skeleton detection. Meanwhile, we adopt the state-of-the-art visual inertial system (VINS) [21] to get the 6 degrees of freedom (DoF) state estimation using image stream of left camera and IMU data stream. It is noted that we use image stream of left camera for 2D skeleton detection and state estimation. Given the pose of the camera, we can obtain the subject's pose in the world coordinates. In the planning module, we predict the next best viewpoint based on the subjects 3D movements and publish a sequence of the waypoints. Then the trajectory planning converts the waypoints to a feasible trajectory in real-time. The drone is commanded to fly through the trajectory and capture the footage with gimbal camera. Tab. I shows the runtime of different modules for each frame. Because the 2D skeleton detection and other modules are running parallelly, the runtime of a frame is less than 200ms, which is sufficient in the aerial filming.

TABLE I
RUNTIME OF DIFFERENT MODULES

TX2	Module	Runtime(ms)
GPU1	Skeleton Detection	117.64
GPU2	Depth Estimation	35.21
	Skeleton Refinement	39.68
	Viewpoint Estimation	20.51
	Trajectory Planning	12.67
	State Estimation	52.22

We integrate processors, stereo cameras, and gimbal camera on DJI Matrix 100 as Fig. 7 shows. Because 2D skeleton detection and the stereo-based depth map take up most of the computation resources, they require GPU for real-time computation. Considering the power efficiency and limited load of the drone, we use two NVIDIA TX2 to run the whole system simultaneously. The TX2 is equipped with a quad-core ARM Cortex-A57 processor, a dual-core Denver2 processor and 8 GB memory, and consumes approximately 7.5 watts of power. The 256 GPU cores on the TX2 make it particularly suitable for parallel computing of depth images and body keypoints detection. The stereo camera module is constructed of two horizontal forward-looking MatrixVision mvBlueFOX-MLC200w5 global shutter cameras (740x480, 25 fps). We choose Zenmuse X3 Gimbal Camera for capturing stabilized footage and record the footage with resolution 1280x720.

We deploy different modules on two GPUs based on their computation complexity. More precisely, one GPU is only used for 2D skeleton detection, and the other GPU covers the rest of the computations. Both TX2 are powered by the battery of the DJI Matrix 100. The two TX2 are connected using an Ethernet cable. Communication between two computers is done by utilizing the ROS infrastructure.

VI. EXPERIMENTS

In this section, we will evaluate our system on CMU Motion Capture Dataset and real-time action scenes. We compare our system with a state-of-the-art autonomous filming technique "Active Track". "Active Track" is an intelligent flight mode on the DJI Mavic Pro, in which the camera can autonomously keep the distance to follow the target and adjust the camera to place subject on the center of the camera screen. We develop the autonomous cinematography system based on DJI Matrix 100. In the following sections, we offer a detailed discussion on 3D skeleton detection (Sec. 3.1), camera planning (Sec. 3.2) and real-time aerial filming (Sec.3.3).

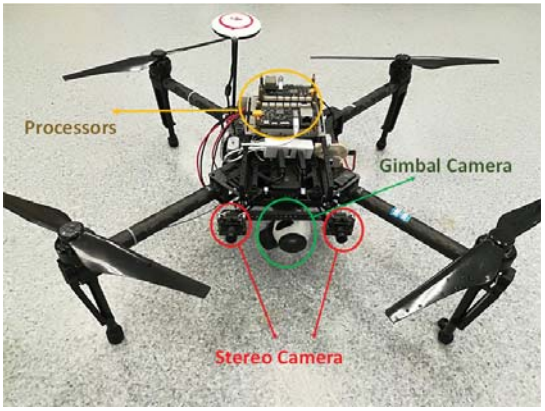


Fig. 7. The prototype drone of the proposed ACT system

A. 3D Skeleton Detection

This section we compare our skeleton detection algorithm with Kinect sensors in the indoor environment. The Kinect can achieve accurate skeleton detection with skeletal tracking SDK, so we set the skeleton keypoints from Kinect as the groundtruth. We evaluate the performance of skeleton refinement (Sec.III.B) in terms of two metrics: 1) Reconstruction Error: The average distance of the 3D point cloud. 2) Viewpoint Estimation Error: Angle difference of the best viewpoint of each frame. We use error distribution in the entire sequence to measure the performance of our system. The larger proportion in the low error case means better performance. We conduct this experiment for slow-paced TaiChi and fast-paced "Gangnam Style" dance .

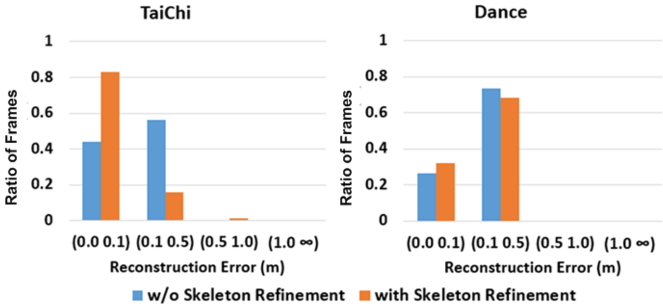


Fig. 8. Comparison of reconstruction between our methods (w and w/o skeleton refinement).

Fig. 8 and Fig. 9 show that skeleton refinement can improve the performance of reconstruction and viewpoint estimation. Meanwhile, the improvement of the reconstruction

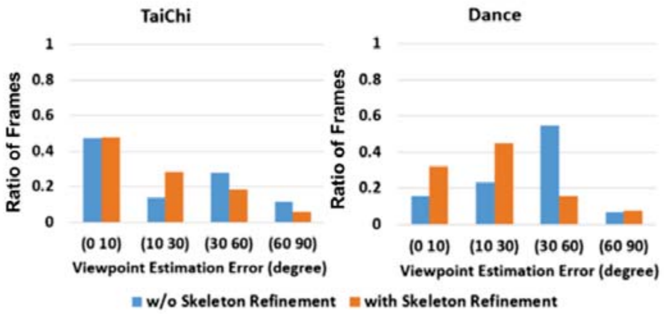
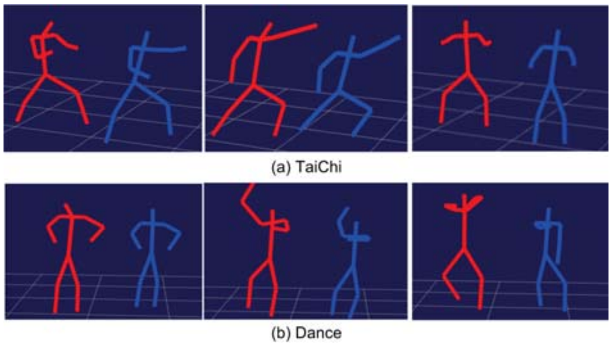


Fig. 9. Comparison of viewpoint estimation between our methods (w and w/o skeleton refinement).

in fast motion is not obvious as in slow motion, because it is more difficult to achieve motion prediction in faster limbs movement. Even so, we can still achieve the accurate best viewpoint estimation after skeleton refinement. Fig. 9 shows that angle error of best viewpoint estimated from our method (with skeleotn refinement) is less than 30 degree in the most cases (more than 70%). In our application, the angle difference within 30 degrees is acceptable, which meets the requirement of view estimation. Moreover, our 3D skeleton detection can work in both indoor and outdoor environments. Fig. 10 demonstrate the skeleton detected from our method with skeleton refinement is similar to that from Kinect. In addition, the reconstruction accuracy decreases as the movement becomes fast.



(a) TaiChi

(b) Dance

Fig. 10. Comparison of 3D skeleton between Kinect (blue) and our method (red) with skeleton refinement

B. Camera Planning

We test our camera planning algorithms on the CMU Motion Capture Dataset (MOCAP), including a set of the motion data and reference video. There is only one subject in each clip of motion data. The 3D motion data is extracted from 41 Vicon markers taped on the subject's body. The motion data is recorded for 6-12 seconds in 120 Hz. In our application, we only consider 13 markers to represent the human action. We select 10 clips of motion capture data from the MOCAP dataset and classify them as 5 clips of slow-paced motion (e.g. TaiChi, Walk) and 5 clips of fast-paced motion (e.g., Dance, Run).

We define the viewpoint that corresponds to maximum projection of point cloud as best viewpoint. We evaluate the viewpoint quality of each frame by the angle difference between the actual viewpoint and the best viewpoint. The aesthetic effect of each piece of footage is evaluated by the distribution of the viewpoint quality. Fig. 11 shows that our drone system can capture the footage from a good viewpoint with higher frequency. This can be explained as the "Active Track" just follows the subject and ignores the pose. Meanwhile, we can see that fast-paced motion (Fig. 11(right)) makes it more difficult for the drone to capture the subject from the best viewpoint because of the physical limits of the drone.

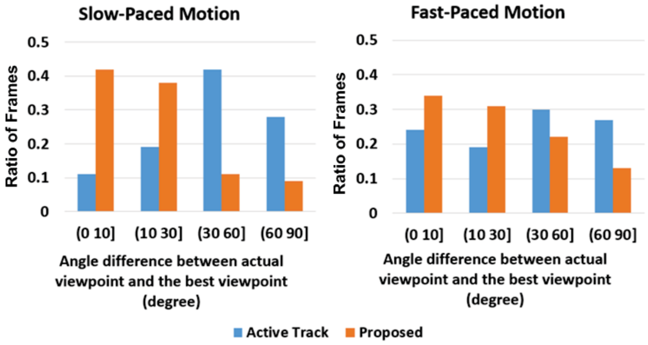
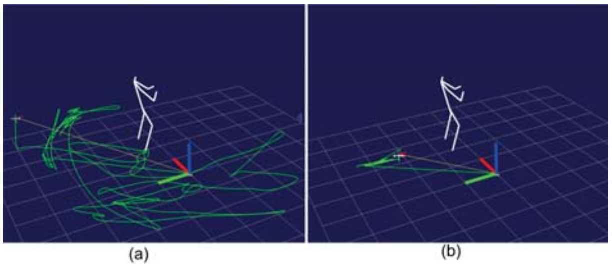


Fig. 11. The distribution of viewpoint quality.



(a)

(b)

Fig. 12. The camera trajectory of (a) the proposed system (b) Active Track

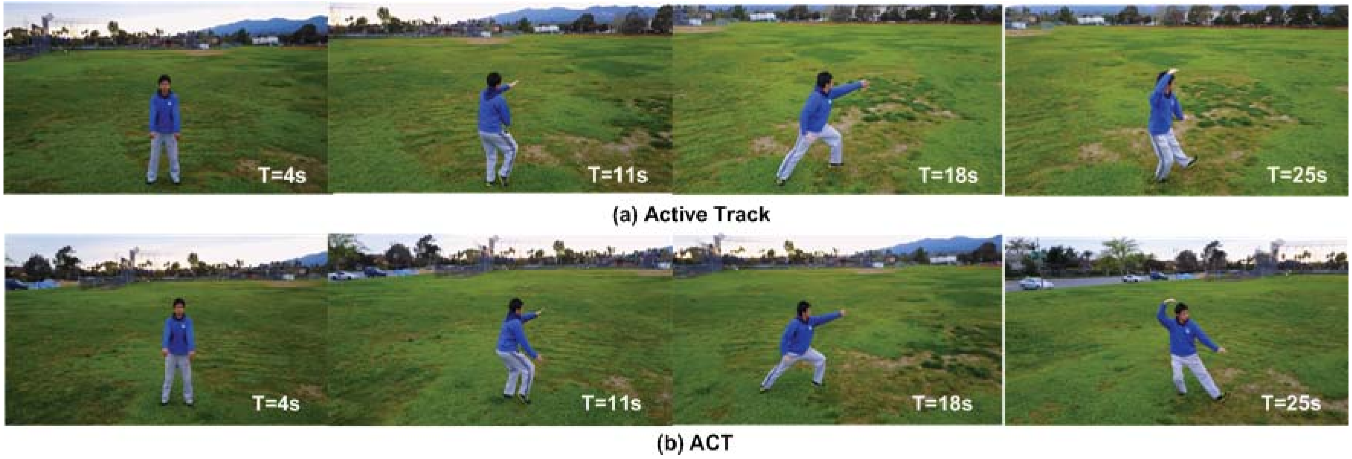
Fig. 12 compares the camera trajectory from our systems and "Active Track" for Tai Chi. Compared with Active Track, the camera trajectory from our system covers more viewpoints and captures more creative footage.

C. Real-Time Aerial Filming

We compare the proposed ACT system with "Active Track" mode of DJI Mavic in the outdoor environment. We initialize the same postion of camera and subject, and then subject performs TaiChi and dance in front of the drone camera. Fig. 14 shows several snapshots of footage captured from both systems. As the human motion goes on, the difference of the footages from both systems becomes more obvious. We can see that the subject in the footage from our system looks more pleasing because of more visible motion and fewer limbs occlusions. The attached videos will provide a more convincing comparison.

VII. Conclusion and Future Work

In this paper, we propose a drone cinematography system ACT that can autonomously capture cinematic shots of action scenes based on limbs movements in both indoor and



(a) Active Track

(b) ACT

Fig. 13. Comparison of the footage snapshots between (a) "Active Track" mode of DJI Mavic and (b) the proposed ACT system. We set the same initial relative position between subject and drone. The snapshots of both methods at T=4 s looks similar, but our system can capture more pleasing shots as the TaiChi performance goes on.

outdoor environments. Our system comprises two modules: 3D skeleton detection and camera planning. First, we propose an efficient method to extract 3D skeleton based on fusion between stereo depth detection and 2D skeleton detection. Second, we design a real-time dynamical camera planning strategy that fulfills the aesthetic objectives for filming and respects the physical limits of a drone. We implement the entire system onboard and demonstrate its feasibility in real scenarios.

The current 3D skeleton detection only utilizes a simple voter to refine the pose, our next step is to use the extended Kalman filter (EKF) to fuse the steoreo measurement and prediction. Besdies this, we aim to extend our system to large-scale action scenes that target multiple unknown subjects. This scenario requires more sophisticated perceptual method and viewpoint quality descriptors. The viewpoint quality descriptor can be learned from a training dataset rather than heuristic definition. Additionally, the current system assumes that there is no obstacle except for one subject. We plan to introduce obstacles avoidance to capture the videos in more general environments.

References

[1] N. Joubert, D. B. Goldman, F. Berthouzoz, M. Roberts, J. A. Landay, P. Hanrahan, *et al.*, "Towards a drone cinematographer: Guiding quadrotor cameras using visual composition principles," *arXiv preprint arXiv:1610.01691*, 2016.

[2] T. Nägeli, J. Alonso-Mora, A. Domahidi, D. Rus, and O. Hilliges, "Real-time motion planning for aerial videography with dynamic obstacle avoidance and viewpoint optimization," vol. 2, no. 3, pp. 1696–1703, 2017.

[3] T. Nägeli, L. Meier, A. Domahidi, J. Alonso-Mora, and O. Hilliges, "Real-time planning for automated multi-view drone cinematography," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 132, 2017.

[4] J. Assa, D. Cohen-Or, I.-C. Yeh, T.-Y. Lee, *et al.*, "Motion overview of human actions," in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 5. ACM, 2008, p. 115.

[5] I. Yeh, C.-H. Lin, H.-J. Chien, T.-Y. Lee, *et al.*, "Efficient camera path planning algorithm for human motion overview," *Computer Animation and Virtual Worlds*, vol. 22, no. 2-3, pp. 239–250, 2011.

[6] D. Rudoy and L. Zelnik-Manor, "Viewpoint selection for human actions," *International journal of computer vision*, vol. 97, no. 3, pp. 243–254, 2012.

[7] J.-Y. Kwon and I.-K. Lee, "Determination of camera parameters for character motions using motion area," *The Visual Computer*, vol. 24, no. 7, pp. 475–483, 2008.

[8] A. Sanna, F. Lamberti, G. Paravati, and F. Manuri, "A kinect-based natural interface for quadrotor control," *Entertainment Computing*, vol. 4, no. 3, pp. 179–186, 2013.

[9] N. Halper, R. Helbing, and T. Strothotte, "A camera engine for computer games: Managing the trade-off between constraint satisfaction and frame coherence," in *Computer Graphics Forum*, vol. 20, no. 3. Wiley Online Library, 2001, pp. 174–183.

[10] W. H. Bares, S. Thainimit, S. McDermott, and C. Boudreaux, "A model for constraint-based camera planning," in *Proceedings of AAAI spring symposium on smart graphics*, 2000, pp. 84–91.

[11] B. Tomlinson, B. Blumberg, and D. Nain, "Expressive autonomous cinematography for interactive virtual environments," in *Proceedings of the fourth international conference on Autonomous agents*. ACM, 2000, pp. 317–324.

[12] E. Dichter, "What's in an image," *Journal of consumer marketing*, vol. 2, no. 1, pp. 75–81, 1985.

[13] M. Gleicher and A. Witkin, "Through-the-lens camera control," in *ACM SIGGRAPH Computer Graphics*, vol. 26, no. 2. ACM, 1992, pp. 331–340.

[14] M. Christie, P. Olivier, and J.-M. Normand, "Camera control in computer graphics," in *Computer Graphics Forum*, vol. 27, no. 8. Wiley Online Library, 2008, pp. 2197–2218.

[15] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 807–814.

[16] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," *arXiv preprint arXiv:1611.08050*, 2016.

[17] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich, "Viewpoint selection using viewpoint entropy." in *VMV*, vol. 1, 2001, pp. 273–280.

[18] T. Kröger and F. M. Wahl, "Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 94–111, 2010.

[19] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2520–2525.

[20] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research*. Springer, 2016, pp. 649–666.

[21] Y. Lin, F. Gao, T. Qin, W. Gao, T. Liu, W. Wu, Z. Yang, and S. Shen, "Autonomous aerial navigation using monocular visual-inertial fusion," *Journal of Field Robotics*, 2017.