Through-the-Lens Drone Filming

Chong Huang[1], Zhenyu Yang[1], Yan Kong[1], Peng Chen[2], Xin Yang[3], and Kwang-Ting (Tim) Cheng[4]

***Abstract*— Aerial filming in action scenes using a drone is difficult for inexperienced flyers because manipulating a remote controller and meeting the desired image composition are two independent, while concurrent, tasks. Existing systems attempt to utilize wearable GPS-based or infrared-based sensors to track the human movement and to assist in capturing footage. However, these sensors work only in either indoor (infrared-based) or outdoor environments (GPS-based), but not both. In this paper, we introduce a novel drone filming system which integrates monocular 3D human pose estimation and localization into a drone platform to remove the constraints imposed by wearable-sensor-based solutions. Meanwhile, given the estimated position, we propose a novel drone control system, called "through-the-lens drone filming", to allow a cameraman to conveniently control the drone by manipulating a 3D model in the preview, which closes the gap between the flight control and the viewpoint design. Our system includes two key enabling techniques: 1) subject localization based on visual-inertial fusion, and 2) through-the-lens camera planning. This is the first drone camera system which allows users to capture human actions by manipulating the camera in a virtual environment. From the drone hardware, we integrate a gimbal camera and two GPUs into the limited space of a drone and demonstrate the feasibility of running the entire system onboard with insignificant delays, which are sufficient for filming in our real-time application. Experimental results, in both simulation and real-world scenarios, demonstrate that our techniques can greatly ease camera control and capture better videos.**

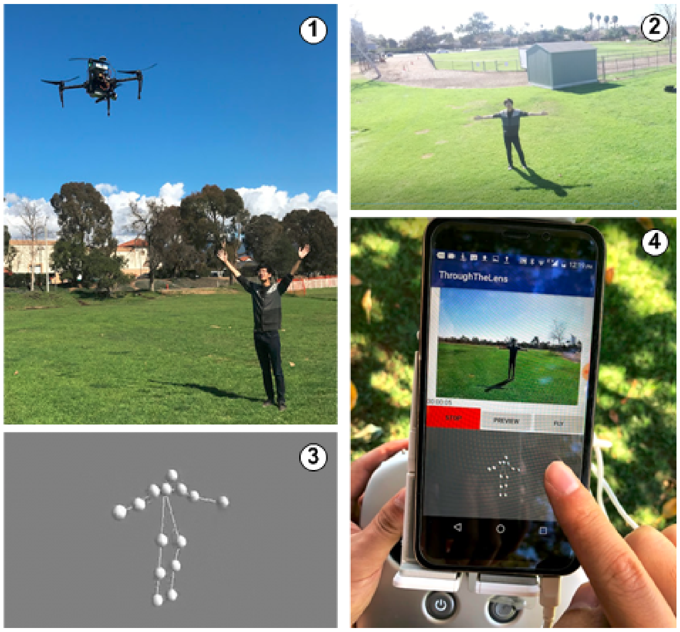


Fig. 1. Overview of the through-the-lens drone filming. (1) The filming scene. (2) Camera View. (3) The preview of 3D human model estimated from camera view. (4) User manipulates the 3D model in the preview to design the viewpoint.

I. INTRODUCTION

The availability of intelligent drones makes it more convenient and accessible to manually capture aerial footage. However, it is still very challenging to manipulate multiple control sticks in a remote controller to capture human movement from a desired viewpoint. While moving control sticks can directly control a drone's motion parameters (i.e. roll, yaw, pitch, and throttle), controlling these parameters do not offer a precise control of the movement of objects in the camera screen. Compared with the direct control of the drone's parameters, through-the-lens camera control [1] parameterizes the camera pose in terms of azimuth, elevation, and radius in a subject-centered spherical coordinate system rather than six degrees of freedom (DOF) in a reference-fixed Cartesian coordinate system. Through-the-lens camera control allows a user to drag or zoom in (out) the subject in the image space to adjust the image composition. This control mode greatly simplifies viewpoint control of a moving subject, so it is widely used in action games and 3D animation. Introducing through-the-lens control operations to drone filming can greatly reduce the difficulty of the manual control and allow a cameraman to focus more on the viewpoint selection.

However, it is difficult to apply this subject-centered control mode in real-world scenarios because the subject's position cannot be directly obtained like computer graphics. Some studies [2] [3] [4] localize the subjects by wearable sensors (e.g. GPS, Vicon) to assist in the drone filming, but these sensors are constrained to specific environments. For example, the GPS-based sensors work only in an outdoor environment. In addition, the wearable-sensors-based solutions are ineffective for unknown targets.

Some researchers [5] [6] [7] [8] use vision-based methods and the prior knowledge of a subject's height to localize the subject. These methods work in both indoor and outdoor environments. Lim et al. [7] localizes the subject based on the position and size of the bounding box estimated from

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

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

[3] 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)

[4] Kwang-Ting (Tim) Cheng is with School of Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. (timcheng@usk.hk)

person detection, but the size of the bounding box is sensitive to the person's pose (e.g. Bending over outputs the smaller bounding box than stretching), which affects the localization accuracy. More sophisticated methods [5] [6] [8] adopt 3D human pose estimation to improve subject localization. Because the output of 3D pose estimation loses the absolute scale and depth, the actual height of the subject is required to recover the scale and depth. However, it is not always feasible to request a user to input the height of each subject in unknown scenes.

To address the above challenges, we propose an efficient drone filming mode, called "Through-the-Lens drone filming" (see Fig. 1), which is enabled by the following techniques:

1. An automatic subject localization method without the prior knowledge of a subject's height. We utilize the drone's motion information and the normalized 3D human pose to estimate the subject's height. With the estimated height, we can localize the subject accurately during filming.

2. An effective interaction that allows the user to control the drone by manipulating the virtual camera in the preview of a 3D model. In addition, our system can convert the desired viewpoints in the virtual environments to a physically-feasible trajectory in the true metric space.

To facilitate users' real-time operation, we mount two GPUs (NVIDIA Jetson TK1 and TX2) on a DJI Matrix 100 drone. In addition, we develop an Android app to provide through-the-lens drone control.

The contributions of this paper are three-fold. First, the localization does not require the prior knowledge of the subject's height, which broadens the application of the system to unknown scenes. Second, the proposed through-the-lens drone filming simplifies the manual control for capturing the subject-focused shot and enables the user to customize the viewpoint for moving subjects in real-time. Third, we optimize the implementation of the entire system based on the limited computation resource of a drone platform, including 2D skeleton detection, 3D pose estimation and localization, and camera trajectory planning, and demonstrate the feasibility of running the system in approximately real-time.

We discuss related work in Sec. II, and introduce subject localization in Sec. III, followed by camera trajectory planning in Sec. IV. The system architecture is in Sec. V. In Sec. VI, we present the simulation and experimental results based on real-world scenarios. We conclude the paper in Sec. VII.

II. Related Work

Camera Control: Through-the-lens camera control [1] has been widely used in virtual cinematography [9] [10] [11] [12] and action games [13] [14]. However, these techniques are not feasible in real-world scenarios because a subject's position cannot be directly obtained like in virtual environments. Some researchers [2] [3] [4] use wearable sensors to localize a subject and automate filming for some predefined shots. In addition to the constraints imposed by sensors, their systems do not provide an efficient interaction for users to design desirable viewpoints.

Subject Localization: The GPS-based [2] and the infrared-based [3] [4] wearable sensors are widely used for subject localization. However, the GPS does not work in the indoor environments, and the infrared-based sensors are restricted to the indoor environment because of their optical properties. Furthermore, it is not convenient to require every subject to wear sensors for filming. The vision-based localization frees the subject from wearable sensors, but the related work [5] [6] [7] requires the user to provide the subject's height, based on which to compute the global translation under perspective projection. However, these methods become invalid for users with unknown heights. Besides, Huang et al. [15] utilizes a stereo camera mounted on the drone to localize the subject, but its field of view is subjected to the drone body and cannot efficiently track the subject when the drone or the subject is moving.

III. Subject Localization based on Visual-Inertial Fusion

In this section, we introduce subject localization based on visual-inertial fusion. Because skeleton-based localization [5] [6] is robust for the varying pose of the subject, and a full 3D model can facilitate users to operate the camera, we adopt the monocular 3D human skeleton estimation (Sec.III.A) as baseline. As mentioned above, skeleton-based methods require a known height to recover the scale and depth of the normalized 3D pose. Fig. 2(a) shows that incorrect height inputs render biased localization. Under an undistorted perspective projection, the localization error is proportional to the error between the actual and the assumed height. Figs. 2(b)(c) show that a subject's positioning information from different camera viewpoints may differ when the assumption is inconsistent with the true height, so our intuition is to find an optimal height which can minimize the bias caused by the camera's movement. Compared with the conventional multi-view 3D reconstruction, the scale range of a human subject is limited (we set the height range for an adult between $1.4m$ and $2.2m$ in the proposed system to reduce the search space). In addition, the normalized 3D poses contain an inherent structure among 3D points cloud and thus, in turn, make it feasible to localize the subject from a moving camera, even when the subject is moving. The proposed subject localization includes three steps: 1) monocular camera 3D human pose estimation, 2) scale and depth initialization, and 3) global subject localization.

A. Monocular Camera 3D Human Pose Estimation

We extract 3D human pose from the images captured by a gimbal camera. This task consisted of two steps: First, we use OpenPose [16] to detect 14 2D joints, including the head, nose, left and right hip, shoulders, elbows, hands, knees, and feet. Second, we use a sequence-to-sequence network proposed in Hossain et al [8] to estimate 3D pose from a sequence of 2D joints. To address incomplete 2D joint

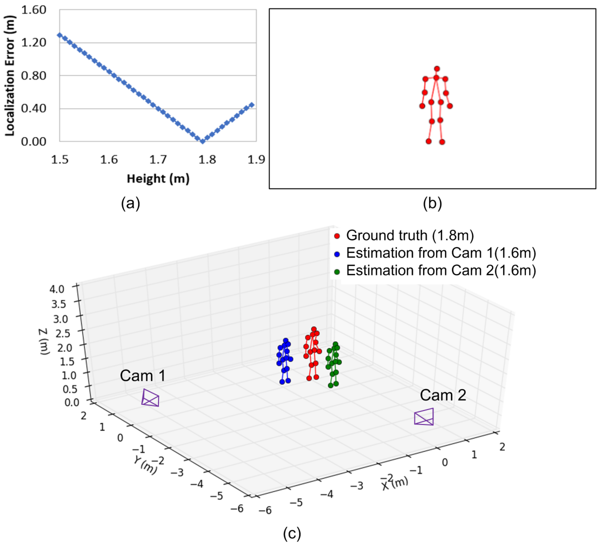


Fig. 2. (a) The localization error for a person with 1.8m height standing in front of a camera (x- and y-focal length is 380 pixel without distortion) by 5m. The x-axis represents the height guess and y-axis represents the error of localization in depth. (b) The camera view captured from Cam 2. (c) A static subject with 1.8m height stands in the groundtruth position (red skeleton). The blue and green skeletons are estimated based on 1.6m assumption from Cam 1 and Cam 2 placed 5 meters away from the subject in different directions. The estimated position from different viewpoints differs from each other.

estimation caused by occlusion, we use the value in the previous frame to compensate for the missing space of the current frame. Because the input of the network [8] has been normalized to zero mean and a standard deviation of 1, the estimated 3D pose loses the absolute scale and depth information.

B. Scale and Depth Initialization

This subsection introduces how to recover the scale and depth of a normalized 3D pose by a moving camera. We start with notation definitions. We denote $(\cdot)^w$ as the world frame, which is initialized by the drone's navigation system. $(\cdot)^c$ is the camera frame and $(\cdot)^v$ is the image frame, where the origin is the center of the screen. We assumed that the camera model is weak perspective projection and the subject's movement is smooth during initialization, we define the following optimization function to minimize two terms 1) F: the image projection error from 3D joint locations, and 2) G: the temporal smoothness of the subject's displacement.

$$\min_{\{\alpha,T_0^c,\cdots T_\tau^c\}} F(\alpha,T_0^c,\cdots T_\tau^c)+\lambda G(T_0^c,\cdots,T_\tau^c)$$
$$F=\sum_{t=0}^{\tau}\sum_{n=0}^{N}\left\|p_{t,n}^v-K(\alpha\hat{P}_{t,n}^c+T_t^c)\right\|^2 \qquad (1)$$
$$G=\sum_{t=1}^{\tau}\left\|(R_t^wT_t^c+T_t^w)-(R_{t-1}^wT_{t-1}^c+T_{t-1}^w)\right\|^2$$

where τ and N are the size of the temporal window and the number of joints. Based on the assumption of smooth movements, the scale between the true height and the normalized 3D pose during the time interval [0, τ] shares the same α. Because 3D joints spread in the depth direction is negligible compared to its distance to the camera, we only use one T_t^c to represent the relative position between each joint and camera coordinates at time t. $p_{t,n}^v$ and $\hat{P}_{t,n}^c$ are the n-th 2D joint locations and the normalized 3D pose at time t respectively. K represents the camera projection matrix. λ is the parameter to balance the penalty between projection error and smoothness constraints.

We propose a simple yet highly efficient method to initialize the scale. First, to quantify the relation between scale α and the camera-subject relative position T^c in camera projection Eq. 2, we use the method in [5] to describe $T^c=(T_x^c,T_y^c,T_z^c)$ as expressed in Eq. 3.

$$\min_{T^c}\sum_{n=0}^{N}\left\|p_n^v-K(\alpha\hat{P}_n^c+T^c)\right\|^2 \qquad (2)$$

$$\begin{aligned}
T_x^c&=\alpha(\frac{\gamma}{f_x}\bar{p}_x^v+\bar{\hat{P}}_x^c)\\
T_y^c&=\alpha(\frac{\gamma}{f_y}\bar{p}_y^v+\bar{\hat{P}}_y^c)\\
T_z^c&=\alpha\gamma\\
\gamma&=\frac{\sum_{n=0}^{N}\left\|H(\hat{P}_n^c-\bar{\hat{P}}_n^c)\right\|^2}{\sum_{n=0}^{N}\left\|(p_n^v-\bar{p}_n^v)H(\hat{P}_n^c-\bar{\hat{P}}_n^c)\right\|}\\
H&=\begin{bmatrix}f_x & 0 & 0\\ 0 & f_y & 0\end{bmatrix}
\end{aligned} \qquad (3)$$

where $\bar{p}_x$ and $\bar{p}_y$ are the average values of x and y of 2D joints, $\bar{\hat{P}}_x$ and $\bar{\hat{P}}_y$ are the average values of x and y of the normalized 3D joints. H is a matrix consisting of the focal length f_x and f_y.

Second, because T_t^c is proportional to α in Eq. 3, we rewrite T_t^c as $\alpha\hat{T}_t^c$, where $\hat{T}_t^c$ can be considered as the relative position of the normalized 3D pose in camera coordinates. We can solve α by substituting $\alpha\hat{T}_t^c$ into Eq. 4. The α in Eq. 5 is set as the estimated scale. Note that if the estimated height is beyond a reasonable range (1.4m-2.2m in our experiments), or the variance of $\left\|\frac{\tilde{T}_{t-1}^w-\tilde{T}_t^w}{R_t^w\hat{T}_t^c-R_{t-1}^w\hat{T}_{t-1}^c}\right\|$ exceeds a threshold (0.4 in our experiments), we move the temporal sliding window to restart the initialization.

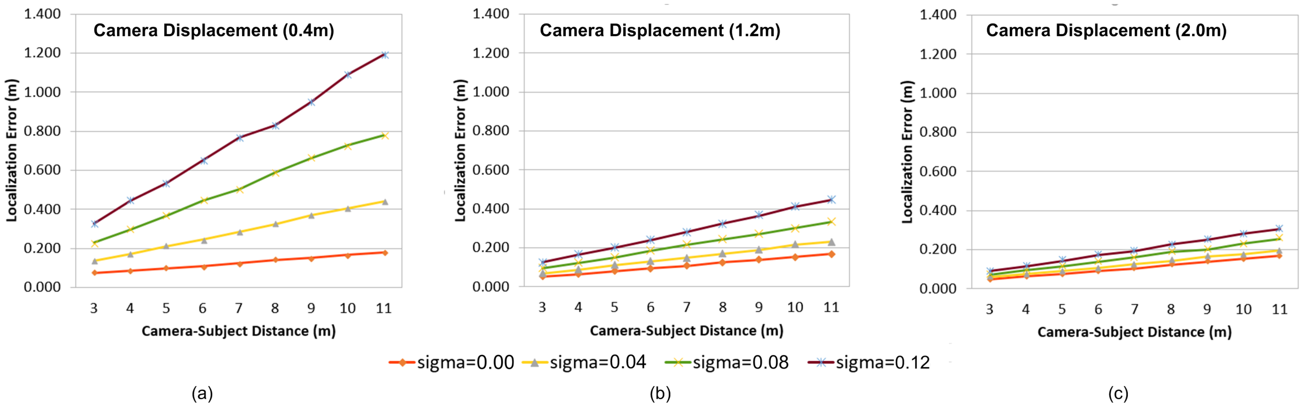
$$\min_{\alpha}\sum_{t=1}^{\tau}\left\|\alpha(R_t^w\hat{T}_t^c-R_{t-1}^w\hat{T}_{t-1}^c)+T_t^w-T_{t-1}^w\right\|^2 \qquad (4)$$

$$\alpha=\frac{1}{\tau-1}\sum_{t=1}^{\tau}\left\|\frac{\tilde{T}_{t-1}^w-\tilde{T}_t^w}{R_t^w\hat{T}_t^c-R_{t-1}^w\hat{T}_{t-1}^c}\right\| \qquad (5)$$

C. Global Subject Localization

Once we finish scale initialization, we can estimate the subject's global position based on Eq. 6 for the following frames.

$$P_t^w=\alpha R_t^w\hat{T}_t^c+T_t^w \qquad (6)$$



(a)

(b)

(c)

Fig. 3. Localization error in terms of different initialization states (camera-subject distance, noise levels and camera displacement)

D. Discussion

In this subsection, we discuss how to move the camera to optimize the localization performance. Considering that the uncertainty of the depth is a function of the length of the baseline between different views, the drone automatically moves sideways to collect 15 images of a subject within a period of 2 seconds to estimate the height. Our strategy is partially motivated by DJI Spark's "ShallowFocus" mode in which a drone creates the effect of shallow depth of the field from 15 images captured during its automatic rising within 20cm. We do not adopt the strategy of elevating the drone to collect images because the change of elevation is likely to degrade the performance of pose estimation.

Scale estimation can possibly be affected by noisy measurements from the navigation system. We neglect the noise of the rotation because the camera gimbal stabilization system can achieve accurate and consistent rotation measurements. To evaluate the localization performance with respect to different initialization states, we design simulation experiments to evaluate the localization error with respect to different camera displacements, camera-subject distance and different levels of noise. For these experiments, we select 42 downsampled motion capture data (average 140 frames, 8 fps, including standing, jumping, sitting, climbing and walking) from Carnegie Mellon University Motion Capture Dataset. We set the height of the 3D model as $1.8m$.

First, we tested the localization error when the subject has no displacement in the space during initialization, where the subject's center is fixed to the origin of the world coordinate system. Based on the physical property of the drone's navigation system, we evaluated the localization error when the camera's translational displacements are $0.4m$, $1.2m$ and $2.0m$ respectively and the standard deviation of the positioning noise is $0.00m$, $0.04m$, $0.08m$ and $0.12m$ respectively. Considering the subject's safety and the maximum distance of 3D pose estimation, we set the range of the camera-subject distance as $[3\text{-}11]m$. Fig. 3 shows that the larger displacement can improve the localization accuracy. In addition, it is harder to localize the subject if the subject is far away from the camera during initialization. This can be explained that the resolution of the limb decreases when the subject moves away from the camera, increasing the image projection error of the subject. In particular, when the displacement is $0.4m$, the increasing noise impacts the performance more obviously as the camera-subject distance increases.

Second, we tested the performance of localization when the subject is free to move within a region during initialization, where the radius of the moving region is set as $[0.3, 0.5, 1.0, 2.0]m$. We also set zero displacement (radius = $0m$) as a reference. This comparison focuses on the case when the camera's displacement is $1.2m$ and the standard deviation is $0.12m$. For cases of other displacement and standard deviation values, the trends are similar. Fig. 4 indicates that the localization becomes worse as the moving region is widened.

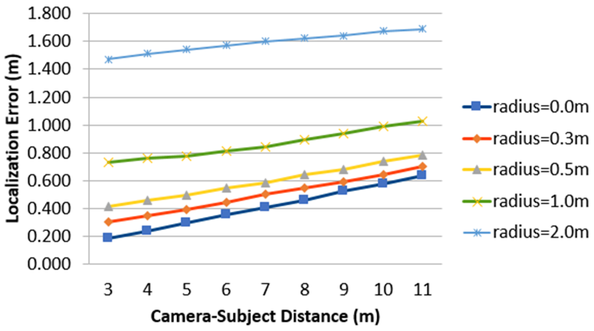


Fig. 4. Localization error with respect to different initialization states (subject's moving regions and camera-subject distance).

From simulation results, we can draw the conclusion that the localization accuracy is determined by a set of initialization states including the subject's movement, the camera-subject distance, the positioning noise and the length of camera's displacement. The localization error, greater than, say, $1.0m$, will affect the subject's safety and thus cannot be allowed. Therefore, to achieve accurate localization within an allowable range, we better choose the moment when the subject is fairly static and set a closer viewpoint to launch initialization.

IV. Through-the-lens Camera Planning

In this section, we introduce through-the-lens camera control and trajectory planning. First, we introduce how a user manipulates the 3D preview to design the viewpoint. Second, we describe a novel automatic filming mode to track the moving subject. Third, we present our trajectory planning strategy to handle these tasks.

A. Through-the-lens Viewpoint Control

This section starts with a short description of the User Interface (illustrated in Fig. 5(A)). The 3D model is rendered by OpenGL based on the normalized 3D pose. Our system allows the user to touch the screen to move the camera view while keeping the position of the 3D object fixed. A user can adjust the viewpoint by rotating and zooming the 3D model and command the drone to capture the desired viewpoint. Through-the-lens control includes:

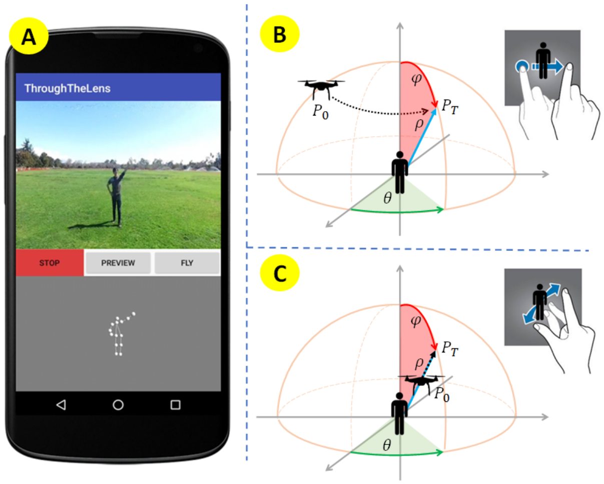


Fig. 5. (A) User Interface of the through-the-lens viewpoint control is consisted of the camera, including a camera view (the upper window) and a 3D model preview (the lower window). The user moves the virtual camera by (B) rotating the 3D model or (C) zooming the 3D model in the 3D model preview window.

Rotate: Swipe the screen to orbit the virtual camera in the horizontal and vertical direction (illustrated in Fig. 5(B)).

Zoom: Spread or pinch the screen to change the field of view of the virtual camera (illustrated in Fig. 5(C)).

We denote $(\cdot)^w_c$ and $(\cdot)^w_s$ as the position of the drone (camera) and the subject in the world coordinates respectively. In addition, we use $(\cdot)^o_c$ to describe the pose of the virtual camera in the virtual 3D environment. Once the user publishes the desired viewpoint to the drone, our system will map the state of the camera $(P(x, y, z), yaw)$ from the virtual 3D environment to the true world space as follows:

$$\begin{aligned} P^w_c &= R^w(\beta P^o_c + T^c) + T^w \\ yaw^w_c &= R^w yaw^o_c \end{aligned} \tag{7}$$

where β is a constant to amplify camera-subject distance for safety.

B. Subject-Oriented Tracking

The above interface allows the user to manipulate 3D model to set the viewpoint. To facilitate users to track the moving person from the desired viewpoint, we further extend it with an automatic filming mode: subject-oriented tracking. In this mode, once the user sets the virtual viewpoint of the subject, the camera will track the subject from a fixed relative position between the subject and the camera. To this end, our system automatically analyzes the subject's skeleton to estimate its orientation. Considering that the direction of two shoulders are normally parallel to the ground, we denote $(\cdot)^s$ as the subject-oriented coordinates, where three axes can be defined as follows:

$$\begin{aligned} z^s &= z^c \\ x^s &= \frac{p^c_{rs} - p^c_{ls}}{norm(p^c_{rs} - p^c_{ls})} \times z^s \\ y^s &= z^s \times x^s \end{aligned} \tag{8}$$

where p^c_{ls} and p^c_{rs} denote the 3D positions of the left shoulder and the right shoulder in the camera coordinates. z^c denotes the z-axis of the camera coordinates. The rotation matrix R^{cs} from camera to subject coordinates is described as $(x^s,\ y^s,\ z^s)$. Once the user sets the tracking viewpoint, our system records the pose of the virtual camera P^s_c and yaw^s_c in the subject coordinates. The corresponding tracking viewpoint in the world space can be expressed as follows:

$$\begin{aligned} P^w_c &= R^w(\beta R^{cs} P^s_c + T^c) + T^w \\ yaw^w_c &= R^w R^{cs} yaw^s_c \end{aligned} \tag{9}$$

C. Trajectory Planning

This subsection discusses generation of a feasible path given the customized viewpoints. First, we require the camera to move along the spherical surface centered around the subject P^w_s to achieve visual-pleasing footage and avoid collision with the subject. Therefore, we adopt Spherical Linear Interpolation (Slerp) [17] to uniformly interpolate a set of intermediate waypoints between the current position P^w_{now} and the desired position P^w_{des} along an arc. The interpolated points are described as follows:

$$\begin{aligned} P^w_{c,i} = &\frac{\sin((1-\frac{i}{N})*\theta)}{sin\theta} * (P^w_{c,now} - P^w_s) \\ &+ \frac{\sin(\frac{i}{N}*\theta)}{sin\theta} * (P^w_{c,des} - P^w_s) + P^w_s \quad i = 1, ..., N \end{aligned} \tag{10}$$

where θ is the angle between $P^w_{c,now} - P^w_s$ and $P^w_{c,des} - P^w_s$, and N is the number of interpolated points. In particular, if P^w_s is in the middle of a line between $P^w_{c,now}$ and $P^w_{c,des}$, Eq. 10 will be reduced to a linear interpolation, rendering that the camera moves across the subject. In order to keep a safe camera-subject distance, we add the midpoint $P^w_{c,m}$ of the semicircular arc as the interpolated point.

Second, we use a simple and efficient polynomial optimization algorithm to perform trajectory planning under the physical constraints of an aerial robot. We model the trajectory as a piecewise polynomial, which is parameterized to the time variable t in each dimension x, y, z and yaw. The trajectory of each dimension can be written as follows:

$$f_\mu(t) = \sum_{j=0}^{n} p_j t^j \quad t \in [0, T], \tag{11}$$

where p_j is the jth order polynomial coefficient of the trajectory, and T is the total time of the trajectory, which is calculated by the segment 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. Instead of solving the optimization problem in [19], we minimize the snap (i.e. k = 4) along the trajectory and integrate the coefficients in all x, y, z, yaw dimensions into one single equation:

$$J = \sum_{\mu \in \{x,y,z,yaw\}} \int_0^T \left(\frac{d^k f_\mu(t)}{dt^k} \right)^2 dt. \tag{12}$$

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 and yaw and Q is the Hessian matrix of the objective function.

To ensure the feasibility of the trajectory, we also define the following constraints:

1) *Waypoint Constraint*: If there exists a waypoint at a temporal point T, we have

$$f_\mu(T) = d_T. \tag{13}$$

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

$$\lim_{x \to T^-} f_\mu^{(k)}(T) = \lim_{x \to T^+} f_\mu^{(k)}(T). \tag{14}$$

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

$$\begin{aligned} \min \quad & p^T Q p \\ \text{subject to} \quad & Ap = d. \end{aligned} \tag{15}$$

In practice, we need to check whether maximum velocity and acceleration of the trajectory exceed the physical limits of an aerial robot. If the trajectory is infeasible, we increase the flight time T and recalculate Eq. 15 to get a new trajectory. We then re-check the feasibility of the trajectory until it meets the requirement. In our implementation, we only check the trajectory at most five iterations and increase the time T by 1.2 times in each iteration. The maximum acceleration and velocity is set to 2.5m/s^2 and 1.5m/s. If the trajectory is still infeasible after five iterations, we do not move the camera. In most cases, we can obtain a feasible trajectory within two iterations. In addition, we limit the minimum height of the drone to 1m to avoid moving virtual camera below the subject and colliding with the ground. Last but not least, the gimbal camera can automatically adjust its orientation to place the center of the subject's 2D skeleton in the center of an image.

V. System Architecture

The system architecture is shown in Fig. 6. In the perception module, we extract the normalized 3D skeleton from the result of 2D skeleton detection. We estimate the scale by fusing the normalized skeleton and motion data from the drone's navigation system. After the scale is estimated, we can localize the subject in the world space. In the planning module, the system receives the virtual camera pose from the mobile device and estimates the user's desired viewpoint. Then the trajectory planning converts the waypoints to a feasible trajectory. The drone is commanded to fly along the trajectory and capture the footage.

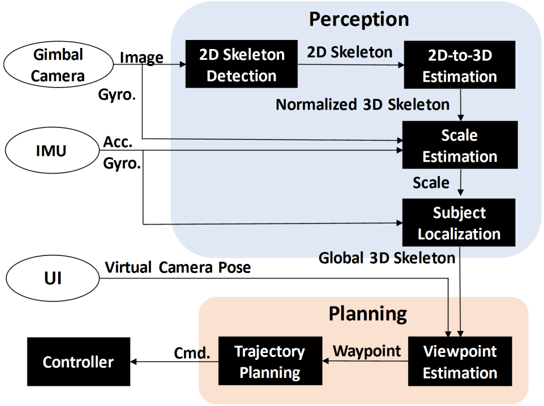


Fig. 6. The architecture of the system

TABLE I
Runtime of Different Modules

GPU	Module	Runtime(ms)
TX2	2D Skeleton Detection	218.47
Manifold	2D-to-3D Estimation	37.44
	Scale Estimation	28.16
	Subject Localization	9.40
	Viewpoint Estimation	12.09
	Trajectory Planning	33.87

We integrate processors and gimbal camera into a DJI Matrix 100 as Fig. 7 shows. We use the DJI Guidance System to provide positioning information. We choose a powerful GPU the NVIDIA Jetson TX2 to run GPU-based 2D skeleton detection. Meanwhile, we use the DJI Manifold (customized Jetson TK1) to decode the video of the onboard gimbal camera and to communicate with the DJI Guidance System. As a result, we use a combination of one Jetson TX2 and one Manifold to run the whole system simultaneously. The 256 GPU cores in the TX2 make it particularly suitable for parallel computing of body keypoints detection. Compared with the Jetson TX2, the Manifold is less powerful and it is equipped with 192 GPU cores. We choose the Zenmuse X3 Gimbal Camera for capturing stabilized footage and record

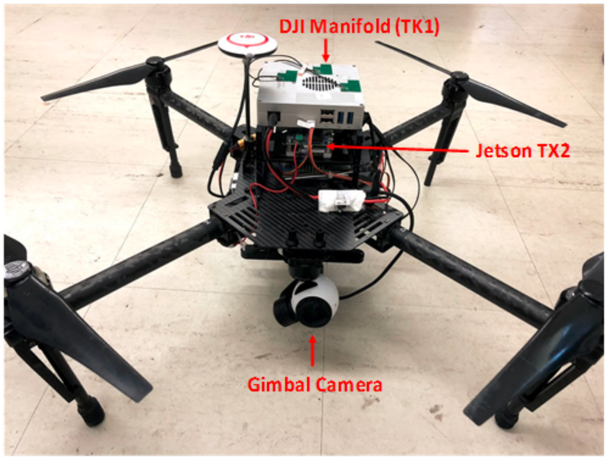


Fig. 7. The prototype drone based on through-the-lens control

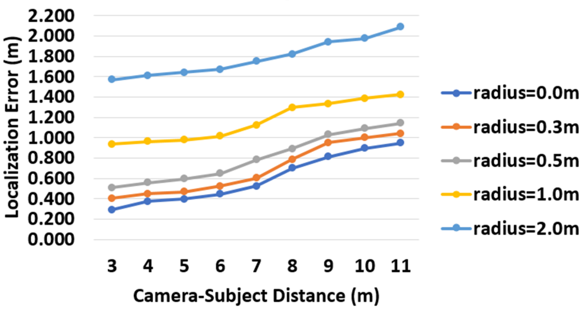


Fig. 8. Localization error in terms of different moving regions (during initialization) and camera-subject distance (after initialization).

the footage with a resolution of 1280x720. To reduce the computation delay, each frame is downsized to 304x176 before further processing.

Table I shows the runtime of different modules for each frame. We deploy different modules to the two GPUs based on their computation complexity. More precisely, one GPU is dedicated for 2D skeleton detection, and the other GPU covers the rest of the computations. Both GPUs are powered by the battery of the DJI Matrix 100 and are connected using an Ethernet cable. Communication between two computers is done by utilizing the ROS infrastructure. The system takes about 300ms to respond to the user's input, which is sufficiently fast for our filming application.

VI. Experiments

A. Subject Localization

In this section, we test the localization accuracy of our system on 8 persons ($1.6m$-$1.9m$) in real filming. We set the camera-subject distance during initialization as $5m$ and allow the subjects to move. Fig. 8 illustrates that our system can achieve sufficient location accuracy (error is less than $1.0m$) in real scenes within $7m$ camera-subject distance. The localization bias becomes more obvious when the camera-subject distance is farther than $7m$. This trend is quite intuitive as decreasing the subject size in the image increases the difficulty of 2D skeleton detection, where the incorrect 3D skeleton further degrades the localization accuracy.

B. Camera Planning

In this section, we evaluate the footage captured from two modes: through-the-lens viewpoint control and subject-oriented tracking. We start with subject-oriented tracking in the simulation. We use the distance between the current and the desired camera position to measure the tracking error. We tested 3 motion capture data (walking, dancing and Tai Chi, average 1300 frames, 30 fps) from the CMU Motion Capture Dataset. We set the height of the 3D model as $1.8m$ and the maximum speed and acceleration as $1.5m{\cdot}s^{-1}$ and $1.0m{\cdot}s^{-2}$, respectively. Meanwhile, we set the viewpoint to focus on the frontal direction of the subject by $3.5m$. We use the average speed of the desired viewpoint (ASDV) to describe the intensity of the human movements. Table II shows that our system can reach the desired viewpoint, with a tracking error less than $1.0m$, for different human movements.

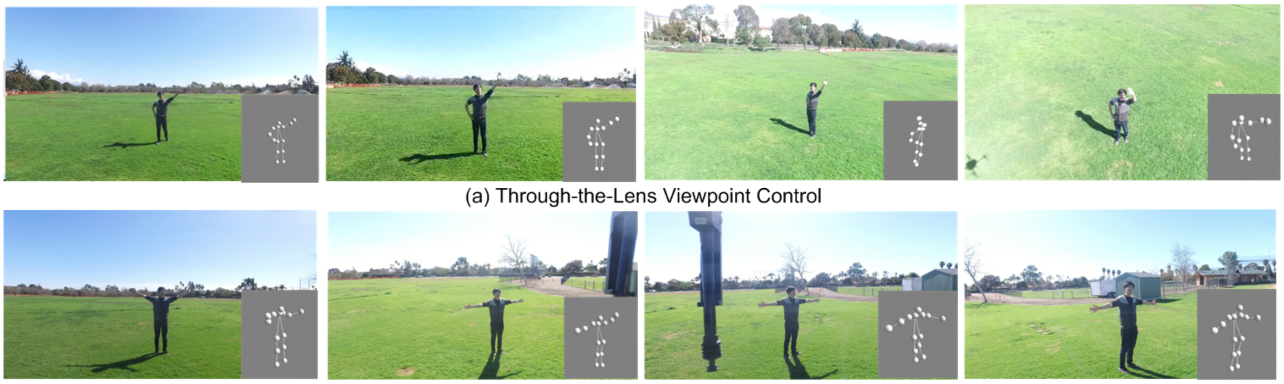
TABLE II
Subect-Oriented Tracking on Different Movements

Motion Description	ASDV (m/s)	Tracking Error (m)
Walk	0.11	0.12
Tai Chi	0.47	0.39
Dance	1.04	0.53

For the real-world scenes, we compare the actual and desired viewpoints in both modes. Fig. 9(a) shows that when the user customizes the viewpoint by zooming in, rotating horizontally and vertically, the viewpoints of the captured footage match the desired viewpoints of 3D model in the through-the-lens viewpoint control mode. Fig. 9(b) shows that the proposed subject-oriented tracking enables the drone to capture the subject from a consistent viewpoint, even when the subject is moving and rotating. The attached demo video confirms high accuracy and impressive performance.

C. Discussion

The current system works well when the subject's limbs are clearly visible, but it becomes difficult for users to manipulate the drone when the human pose cannot be accurately recognized. Fig. 10(a) shows that the limb of the subject is vague due to a long camera-subject distance. In addition, the camera viewpoint also affects the 3D model visualization. Fig. 10(b) shows that a sharp angle decreases the body's visibility and makes it difficult to recognize the limbs. These problems are partially due to the fact that our system processes resized images (304x176) to reduce the computation delay. We plan to compress the current



(a) Through-the-Lens Viewpoint Control

(b) Subject-Oriented Tracking

Fig. 9. The actual viewpoint in real-world filming and the desired viewpoint of the 3D model (the subfigure on the right-bottom). (Top) The snapshot of through-the-lens viewpoint control. The user controls the drone by manipulating the 3D model. (Bottom) The snapshot of the subject-oriented tracking. The user sets the desired viewpoint as the front-right direction in the 3D preview window (first from the left), and then the drone camera keeps tracking the subject from the customized viewpoint.

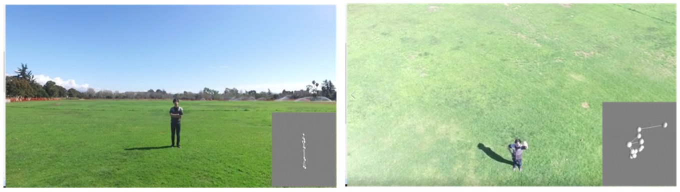


Fig. 10. The viewpoint and distance affects the 2D skeleton detection, making it difficult for user to visualize the 3D model.

2D skeleton network to process a larger-size image and to perceive a greater range.

VII. Conclusion

We propose a novel and efficient drone filming mode: through-the-lens drone filming, where the user can capture the visual-pleasing footage by manipulating the 3D model of the target. This mode closes the gap between the controller manipulation and viewpoint design and greatly reduces the difficulty of drone control for inexperienced flyers. The proposed system comprises two modules: 1) subject localization based on visual inertial fusion, and 2) through-the-lens camera planning. Compared with the state-of-the-art techniques, our localization method does not require the wearable sensors and the prior knowledge of a subject's height, making it applicable to unknown scenes. The through-the-lens control mode enables the user to design the viewpoint for a moving subject in real-time.

References

[1] M. Gleicher and A. P. Witkin, "Through-the-lens camera control," in *SIGGRAPH*, 1992.

[2] 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.

[3] 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," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1696–1703, 2017.

[4] 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.

[5] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3d human pose estimation in the wild using improved cnn supervision," in *2017 Fifth International Conference on 3D Vision (3DV)*, 2017.

[6] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt, "Vnect: Real-time 3d human pose estimation with a single rgb camera," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 44, 2017.

[7] H. Lim and S. N. Sinha, "Monocular localization of a moving person onboard a quadrotor mav," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2182–2189.

[8] M. R. I. Hossain and J. J. Little, "Exploiting temporal information for 3d pose estimation," *arXiv preprint arXiv:1711.08585*, 2017.

[9] L.-W. He, M. F. Cohen, and D. H. Salesin, "The virtual cinematographer: a paradigm for automatic real-time camera control and directing," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 217–224.

[10] J. Assa, L. Wolf, and D. Cohen-Or, "The virtual director: a correlation-based online viewing of human motion," in *Computer Graphics Forum*, vol. 29, no. 2. Wiley Online Library, 2010, pp. 595–604.

[11] A. Litteneker and D. Terzopoulos, "Virtual cinematography using optimization and temporal smoothing," in *Proceedings of the Tenth International Conference on Motion in Games*. ACM, 2017, p. 17.

[12] C. Lino and M. Christie, "Intuitive and efficient camera control with the toric space," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 82, 2015.

[13] T. Wischgoll, "Display systems for visualization and simulation in virtual environments," *Electronic Imaging*, vol. 2017, no. 1, pp. 78–88, 2017.

[14] T. Kawagoe, Y. Yamada, H. Umemiya, and M. Ogawa, "Video game apparatus and method with enhanced virtual camera control," Sept. 2 2003, uS Patent 6,612,930.

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

[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] K. Shoemake, "Animating rotation with quaternion curves," in *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '85. ACM, 1985.

[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 *IEEE International Conference on Robotics and Automation (ICRA)*, 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.