FlyCam: Multitouch Gesture Controlled Drone Gimbal Photography

Hao Kang, Haoxiang Li, Jianming Zhang, Xin Lu, and Bedrich Benes

***Abstract*—We introduce FlyCam—a novel framework—for gimbal drone camera photography. Our approach abstracts the camera and the drone into a single flying camera object so that the user does not need to think about the drone movement and camera control as two separate actions. The camera is controlled from a single mobile device with six simple touch gestures such as rotate, move forward, yaw, and pitch. The gestures are implemented as seamless commands that combine the gimbal motion with the drone movement. Moreover, we add a sigmoidal motion response that compensates for abrupt drone swinging when moving horizontally. The smooth and simple camera movement has been evaluated by user study, where we asked 20 human subjects to mimic a photograph taken from a certain location. The users used both the default two joystick control and our new touch commands. Our results show that the new interaction performed better in both intuitiveness and easiness of navigation. The users spent less time on task, and the System Usability Scale index of our FlyCam method was 75.13, which is higher than the traditional dual joystick method that scored at 67.38. Moreover, the NASA task load index also showed that our method had lower workload than the traditional method.**

***Index Terms*—Gesture, posture and facial expressions, telerobotics and teleoperation, virtual reality and interfaces.**

I. Introduction

THE consumer civilian drone technology has became increasingly accessible and affordable. Many advances have been dedicated towards longer flight time, collision avoidance and path customization. Consumer drones are also often equipped with a high-quality camera mounted on a rotatable gimbal that is controlled separately. People most commonly fly the drones to obtain impressive videos or to take pictures. In a typical configuration, the real-time drone camera streaming is viewed with the help of a mobile application running on a smart phone or a tablet. Some drones store the videos on the on-board memory card that can be viewed later.

Most of the technological progress has been dedicated to the drone themselves and the most common way to control them is by using a dual joystick remote controller (RC), where one joystick is used for turning the drone and the other joystick is for propelling. The gimbaled camera needs additional control that complicates the navigation. While commonly used in amateur and professional planes and drones, this kind of navigation is not intuitive for beginners. The dual joystick operation asymmetry leads to a long learning curve for the starters and has caused many failures and destroyed drones. Even skillful users need to take into account additional consideration for drone control that is distracting when a particular objective, such as a photo or a video, is being targeted. This situation is exacerbated with drones with a separate camera control. In order to get a desired view, the user must steer the drone to reach an approximate location, then adjust camera orientation to see the resulting view. If the view is not as expected, the drone needs to be moved further, camera adjusted, etc. The user usually needs to iterate this process to achieve the desired camera view.

Our key observation is that the Human-Drone Interaction could be more intuitive and natural if one would decouple the mechanical control from the desired objective. A goal-oriented design would let the users forget about the drone and only focus on the high level tasks. The low level motor control would be abstracted out from the users and the users should be able to operate the views with their flying camera directly, rather than worrying about the direction where the drone has to go. Moreover, as a derived camera application running on mobile devices, the drone photography applications could naturally integrate common touch gestures for camera and drone controls to replace the RC.

In this letter, we introduce FlyCam, a multi-touch camera view manipulation framework for drones equipped with cameras with gimbal. Our framework substitutes the traditional RC for drone controls by simplifying the low level aircraft controls, together with gimbal operations, to only six simple and intuitive multi-touch gestures. A single finger drag rotates the aircraft and camera; a double finger drag drives the drone up/down or left/right; and a single/double tap hold moves the drone forward or backward along the camera optical axis. The speed of the drone actions are controlled by the dragging distance on the screen or the tapping pressure. The direct manipulation of the camera view instead of drone and gimbal operations significantly reduces the difficulty of photo composition process with

Manuscript received February 23, 2018; accepted June 19, 2018. Date of publication July 16, 2018; date of current version August 8, 2018. This letter was recommended for publication by Associate Editor S. Rossi and Editor D. Lee upon evaluation of the reviewers' comments. The work was supported only by Adobe Research. *(Corresponding author: Hao Kang.)*

H. Kang and B. Benes are with the Department of Computer Graphics, Purdue University, West Lafayette, IN 47901 USA (e-mail: kang235@purdue.edu; bbenes@purdue.edu).

H. Li is with the AIBee, Palo Alto, CA 94306 USA (e-mail: hxli@aibee.com).

J. Zhang, and X. Lu are with the Adobe Research, San Jose, CA 95110 USA (e-mail: jianmzha@adobe.com; xinl@adobe.com).

This letter has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. The Supplemental Materials contain a video demonstrating the user interface of FlyCam multi-touch gesture drone control framework. This material is 5.63 MB in size.

Digital Object Identifier 10.1109/LRA.2018.2856271

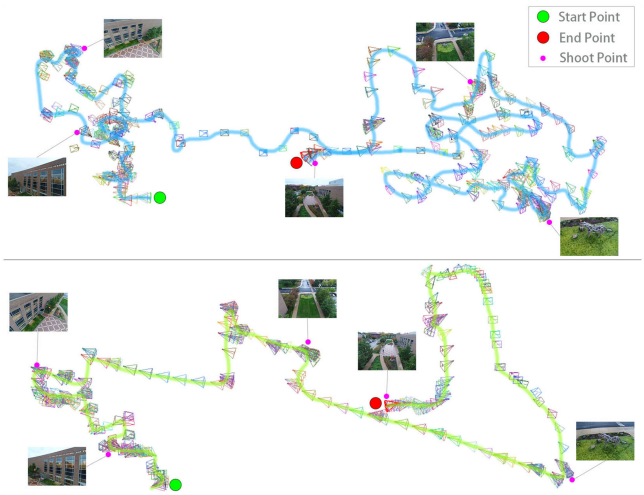

Fig. 1. Comparison of the drone trajectories taken by the traditional dual joystick RC method (top) and our new FlyCam method (bottom). The goal of the experiment was to recover five views given as photographs. The top trajectory is longer and more intricate, indicating that the user had to perform more adjustments and put more efforts during the task. The bottom trajectory is more direct and concise, indicating that the user was able to get to the desired locations quickly and fine-tune the position better by using our method.

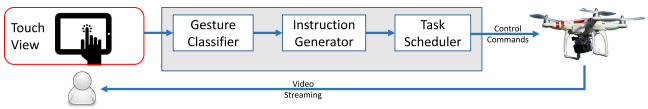

Fig. 2. Overview of the pipeline for FlyCam framework. The user inputs gestures that are classified and instructions for the drone navigation are generated and scheduled. Visual feedback is immediately shown on the screen.

Fig. 3. Holding behavior of the control tablet can vary for different sizes of the screen. Small screens are controlled with thumbs, whereas large screens are controlled by one hand.

drone camera. The difference is reflected clearly in Fig. 1 as trajectories extracted from one user study of five shooting tasks with a drone (VI-C).

We have performed a user study, where we compare the traditional RC control with our new interface. The results show that our framework offers better efficiency in the drone photography tasks, and our new interface provides a better usability and a lower workload to the users. Our main contribution is in providing a unified framework that encapsulates control of movement of drones and camera control.

II. Related Work

We relate our work to the control of gimbaled camera and aircraft and to the touch gestures.

Gimbaled camera and aircraft control The gimbaled camera control of UAVs was discussed before the consumer drones becomes popular in [1], [2]. Two studies of fly-by-cameras [3], [4] analyzed the kinematics models of drones and camera and motivated our work. Drone manufactures have also introduced various First-Person View (FPV) displays [5], [6]. The displays can track the head pose of the user, and reflect action to the drone gimbal. Contrary to the previous work, the novelty of our work is that the user can control both the gimbal and the drone movement by touch gestures from a single mobile device.

The most conventional method of consumer drone controls relies on a dual joystick Remote Controller (RC) and it is commonly used for example in works [5], [7], [8]. For lighter, smaller, and more affordable consumer drones, the RC is replaced by a mobile application that uses on-screen virtual joysticks [9] or device built-in accelerometers to control the drone [10]. However, this requires the users to have an understanding of drone dynamic behavior, which are not designed naturally for efficient and undemanding photo composition. This often causes navigation errors and even damage to the UAVs.

Prior research on human robot interactions proposes a number of novel drone controls. Various hands free control methods, such as eye tracking [11], [12], speech [13], [14], and brain electroencephalogram [15], [16], were applied to control UAVs. Body gestures were also widely studied and some rely on external sensors to capture the gestures, such as Microsoft Kinect in [17]–[19], the Leap Motion controller [20], [21], or wearable devices [22], [23]. Other methods use the on-board cameras or sensors to guide a single UAV or a team of UAVs [24]–[27]. Empirical studies on Human-Drone Interaction (HDI) using body gestures were conducted to explore the natural human behaviors in the interaction scenarios [28]–[31]. Multi-modal UAV controls were also used to gain better control over hybrid modes. The combinations of speech, gesture (hand and body), and visual markers were applied in [32], [33], and they were compared and discussed by Abioye *et al.* [34]. The nontraditional input modalities were analyzed to form a scheme in developing intuitive input vocabulary [35]. However, it is difficult to translate natural vocabulary into drone instructions for precise control.

Path customization was explored as a task level UAV control and some results have been successfully applied to the consumer drone industry. The path customization is mainly set up for drone photography or video recording trajectory planning by using pre-programmed command sets [36], key-frame positioning [37]–[39], viewpoint optimization [40], [41], way-point setting, and following the user motion [7], [8], [42]. Existing systems enable designing cinematography shots ahead of time in a virtual environment. In contrast, our system makes it easier to perform artistic exploration while the drone is in mid-air, which could be useful, e.g., to explore how the scene looks in real-world lighting conditions.

Touch gestures Researchers explored and defined natural multi-touch gestures with 3D objects on large screens in [43], [44], as well as single-touch techniques for virtual camera manipulation on small devices [45], [46]. Navigation in virtual 3D environment using multi-touch gestures were also investigated [47], [48] and these studies are instructive for multi-touch gesture design for drone navigation, but they were focusing on virtual 3D environment.

Multi-touch gestures have been applied in UAV Ground Control Station (GCS) [49]–[51], and experimented in Human-Robot Interaction (HRI) in the context of teleopera-

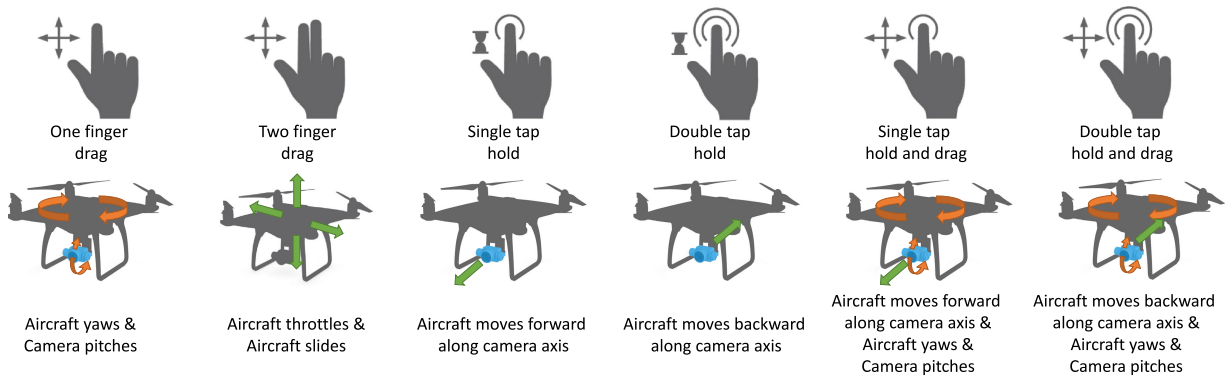

Fig. 4. Our six gestures and the corresponding drone actions.

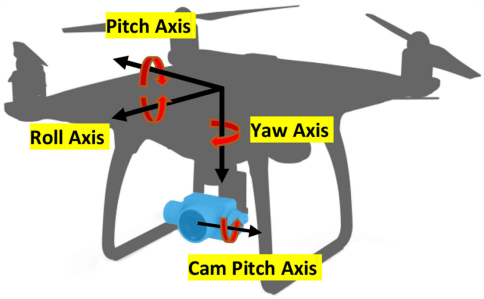

Fig. 5. Drone coordinate system.

tions [52], bipedal walk [53], and general control [54]. Close to our work is the research of Chen *et al.* [55] and Gross [56] who introduced methods to operate a drone through camera view manipulation with multi-touch gestures. A more recent research XPose [57] also provides an intuitive touch-based interface for semi-autonomous photo shooting via points of view. The main difference to our work is that the other studies were not considering the gimbal operations, while gimbal plays an important role in drone photography nowadays. Contrary to our work, the gestures are used to navigate the drone movement and not to unify movement with the control of the camera.

III. System Overview

FlyCam framework consists of four modules shown in Fig. 2. The application runs on a mobile device, takes as input multi-touch gestures, and visualizes the drone camera streaming as the output (see also Fig. 8 for the graphical user interface and the accompanying video for real-time demo).

The *Touch View* that takes multi-touch gestures as the input. The *Gesture Classifier* detects and categorizes the user input into meaningful gestures and parameterizes them. For example, moving one finger to the left is interpreted as rotating left. The distance of the stroke is calculated as the parameter of the corresponding angle. The *Instruction Generator* converts the gestures and their parameters into drone control instructions that are sent to the drone as a commands. This block also unifies the heterogeneous operations between the gimbal and drone. Finally the *Task Scheduler* communicates directly with the drone.

IV. Gesture Control

Because of the landscape orientation of the streaming video from the drone, the mobile device is held horizontally. Users prefer to use two thumbs to perform touch gestures on small screen devices and they hold the device with one hand, and perform touch gestures with the other hand alone on large screens as shown in Fig. 3.

We employed four atomic *gestures* in FlyCam framework that are combined into six gestures that serve well for both holding behaviors from Fig. 3. The atomic gestures are one or two finger drag, single tap hold, and double tap hold. These four gestures can be performed easily with two thumbs, as well as one single hand.

Fig. 4 and the accompanying video show the six gestures used in FlyCam framework and the corresponding mapping to the drone actions. These six touch gestures constitute the user input that is captured, parsed, and abstracted by the four modules of the framework. The framework allows to fly the camera freely without the user needing to concentrate on the low level drone controls. It also seamlessly links the aircraft movement with the gimbaled camera operation, which provides a more user-friendly fly-by-camera mode (see Section V-C).

V. System

The six gestures from Section IV are implemented in our system that can recognize them from the touch screen, interpret, and send as control commands to the actual drone (see Fig. 2).

A. Touch View

Touch View is the interaction layer of the framework. It provides the Graphical User Interface (GUI) (shown in Fig. 8) and takes the multi-touch operations as user input. Touch View also receives, decodes, and presents drone camera streaming in real time that allows the user to see immediate visual feedback of their operations.

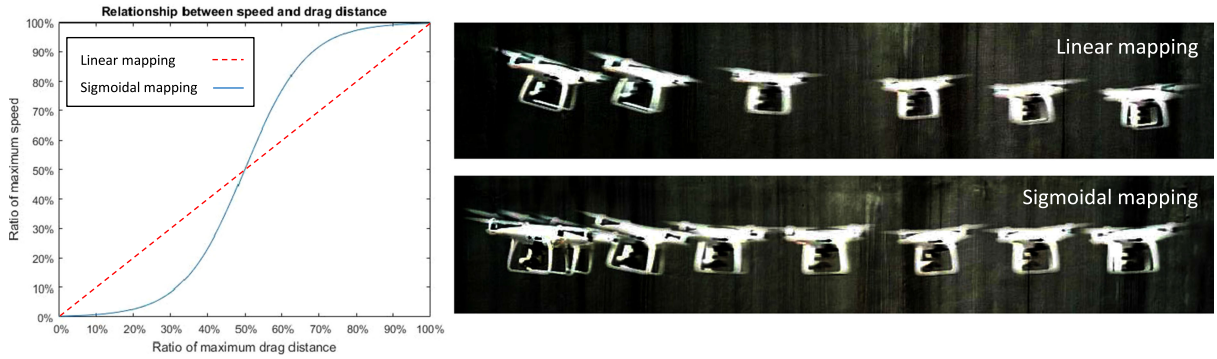

Fig. 6. Linear mapping of the velocity and the drag distance cause the drone to move abruptly and sway at the beginning and at the end of each gesture (top right trajectory). We compensate for this behavior by using a sigmoidal function, and the corresponding trajectory is shown on the bottom right.

B. Gesture Classifier

The *Gesture Classifier* identifies touch gestures and categorizes them into the types of drone action by converting them into parameterized actions for generating drone instructions. The conversion parameterizes the drag distance in the screen $x - y$ coordinate and the touch pressure for tap hold action.

C. Instruction Generator

We assume a drone with five Degrees of Freedom (DoF) and the associated coordinate is shown in Fig. 5. The five DoF are: 1) translation on roll axis, 2) translation on pitch axis, 3) translation on yaw axis, 4) rotation around yaw axis, and 5) rotation around camera pitch axis.

The movements of the aircraft and the gimbaled camera are controlled by a combination of the velocities on the five DoF - three line velocities and two angular velocities. The parameters received from the *Gesture Classifier* contains drag distance or touch pressure, together with drone action type - translation on camera optical axis, slide, throttle, yaw, and gimbaled camera pitch. The drag distance and touch pressure are used for determining the speeds. For the action of translation on camera optical axis, the stronger the pressure is, the higher the speed is. The pressure is retrieved from the device as a float pointing number in range $[0.0, 1.0]$. For the slide, throttle, yaw and gimbaled camera pitch actions, the larger the drag distance is, the higher the speed is. The relationship between the speed and drag distance is shown in Fig. 6 and we use two kinds of mapping:

$$r_v = c||p_1 - p_0|| \tag{1}$$

$$r_v = 1/\left(\mathrm{e}^{-12||p_1 - p_0||+6} + 1\right), \tag{2}$$

where r_v is the ratio of drone maximum velocity, and p_0 and p_1 are gesture start and end points, and c (Eqn (1)) is a scalar constant depending on device resolution. The mappings are shown and compared in Fig. 6. The simple linear mapping in Eqn (1) causes the drone to accelerate fast and overshoot at the end. We experimentally observed that the logistic function mapping (Eqn 2), which is used in our implementation, compensates for the weight of the drone and provides smoother and more coherent drone trajectory which leads to stable images and better user experience. The second row of Fig. 6 shows that the trajectory is nearly as horizontal as directed.

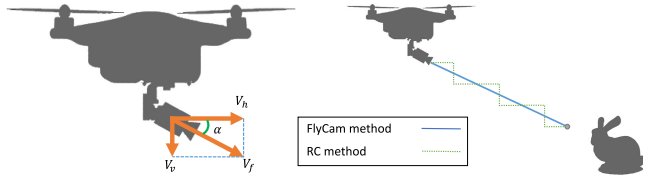

Fig. 7. Our unified control of the drone motion and camera motion allows for a smooth transition between the motion and camera aiming.

Our main contribution is the union of the heterogeneous operations between gimbal and aircraft that is achieved by redefining the forwards and backwards actions. Rather than being relative to the drone heading direction, these two actions are changed to be relative to the camera optical axis. The horizontal speed (on roll axis) and vertical speed (on yaw axis) of the drone can be calculated with orthogonal decomposition on forwards or backwards speed:

$$\begin{bmatrix} v_h \\ v_v \end{bmatrix} = \begin{bmatrix} 0 & \cos\alpha \\ \sin\alpha & 0 \end{bmatrix} \begin{bmatrix} v_f \\ v_f \end{bmatrix} \tag{3}$$

$$v_f = r_v v_{\max} \tag{4}$$

where v_h is the horizontal and v_v the vertical component of the forward velocity v_f, and α is the camera pitch angle relative to the horizontal plane. The forward velocity v_f is a portion of the drone maximum velocity $v_{\max}$ determined by the ratio r_v from Eqn (2).

Fig. 7 shows the velocity decomposition and the trajectory comparison of the two methods on a diagonal motion towards a target. The trajectories reflect the operation simplification brought by FlyCam method.

The drone command set is constructed with the velocity information that is based on the ratio of max speed for each DoF and the velocity decomposition.

D. Task Scheduler

The communication from the framework to the drone is executed by the *Task Scheduler* module. This module maintains a thread that periodically reads the afore-described velocities

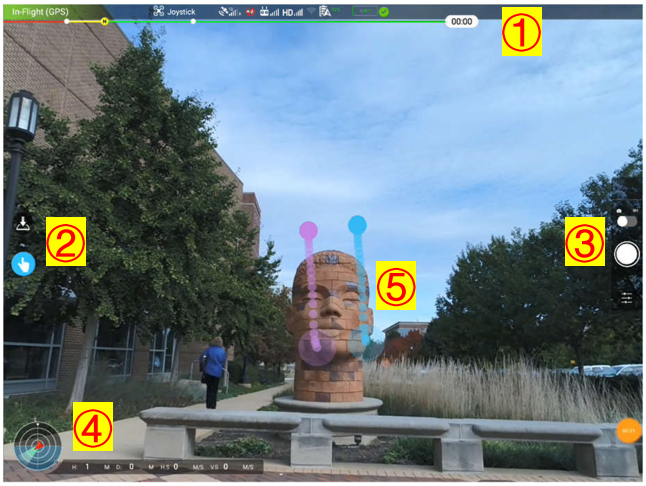

Fig. 8. The Graphical User Interface of FlyCam framework.

TABLE I
MODULE TIMING IN [MS]

Gesture Classifier	0.18
Instruction Generator	3.79
Task Scheduler	20.32
Framework & Drone Communication [a]	33.60
Sum	57.89

[a] The latency was measured with a distance of 30 meters in the open space with a strong signal.

inside the command set [v_{pitch_axis}, v_{roll_axis}, v_{yaw_axis}, ω_{yaw}, ω_{cam_pitch}] from the *Instruction Generator* module and sends instructions to the gimbal and aircraft respectively.

VI. IMPLEMENTATION AND EVALUATION

A. Implementation

We have developed the application and we tested it by using DJI Phantom 4 Pro [7]. Our framework was implemented in Java on a 9.7" Android tablet (ASUS ZenPad 3S 10) and a 5.2" Android phone (Huawei P9). We have used DJI Mobile SDK for Android 4.3.2 [58] and DJI UILibrary for Android 1.0 [59].

Fig. 8 shows the GUI of FlyCam framework. The GUI is displayed on the top of the real-time camera streaming. The top status bar (#1) indicates the information such as the pre-flight aircraft status, GPS and remote controller signal strength, remaining battery power, etc., #2 indicates two buttons for drone taking off/landing and gesture mode activating/deactivating, #3 indicates the camera widget for photo shooting and video recording as well as advanced settings. The dash board widget (#4) provides the aircraft compass, as well as some in-flight information such as distance, altitude, and velocity. The traces in the center of the screen (#5) are examples of multi-touch gesture that have been applied to the framework, in the case of Fig. 8 double finger drag: aircraft throttle up is displayed.

B. System Evaluation

The application provides real-time feedback and the timing of the individual system modules from Fig. 2 is shown in Table I.

Fig. 9. The ground truth photo (left), a photo taken with FlyCam method (middle), and with the traditional RC method (right).

One touch gesture can be classified and turned into corresponding drone commands within a few milliseconds. The task scheduler module executes a command every 20 milliseconds to load and send out commands. The bottleneck of the framework implementation is the communication between the framework and the drone which is a limitation of the hardware and the underlying SDK. The speed of our application is sufficient to provide complete control over the drone.

C. User Study

We conducted a comparative user study between the traditional RC and FlyCam method. All participants were exposed to both approaches and they were also asked to capture the same set of photographs. A post scenario survey was made by using the System Usability Scale (SUS) and The NASA Task Load Index (NASA-TLX). The results were compared and analyzed for four criteria: 1) photo similarities, 2) task time spent, 3) SUS score [60], and 4) NASA-TLX score [61], [62]. These measurements evaluate how quickly and how easily the participants were able to get a desired photograph (Fig. 9).

1) Participants: Our user study included 20 volunteers (50% female and 50% male) of ages 19–33 years with the mean of $\mu = 23$. The participants have background in technology (12), engineering (3), design (3), science (1), and management (1). None of the participants had any prior drone operation or related experience.

2) Apparatus, Setting, and Tasks: The study was conducted outdoors with the drone Obstacle Avoidance (OA) sensors fully activated. The participants were supervised by a certified professional drone operator (guide) for the whole study for safety consideration.

We prepared five photos (ground truth) that were taken in advance on the test site (the photographs as well as the photos taken by the users are available as additional material). The ground truth photographs include significant visual point taken from varying angles, ranges, and compositions. The tasks are to reproduce the given ground-truth photos. The sequence of the ground truth photos was fixed. Without setting any time limit, each study took about 45–60 minutes including demonstration time, drone testing time, talk time, exit survey time, etc.

3) Procedure: For each participant, we randomly decided the order of the two methods to avoid the sequential effect of tested methods as suggested by Yu and Cohen [63].

After the testing order had been decided, a brief introduction of the drone and the tested method was given to the participant. The participant then had three minutes for a test flight in order to get familiar with each method of control. Before the actual

testing, the five tasks were introduced and explained to the participants.

The drone was started by the certified guide, recording was turned on, and then the control device was passed to the participant. The participant was shown the hard copies of the ground truth photos one by one and was asked to reproduce the photos. The average time to complete the tasks for traditional RC method was 7 minutes 34 seconds, and for FlyCam method was 7 minutes 02 seconds. During the tasks, the participant was allowed to ask the guide about the usage if it was needed. When the participant finished the last task, the screen recording was stopped and the drone was landed by the guide.

After both the methods were tested, the participant was asked to complete a web-based exit survey. The survey as well as the testing were anonymous, and included demographic information, SUS questionnaire, and NASA-TLX assessment. The survey took 10–15 minutes to complete. Moreover, we have also recorded the time spent for each photograph. The screen recording video, and the photos taken by the participants were archived.

D. Results

1) Similarity of Photo Composition: We contrasted the photos taken by the participants with the ground truth photos. In order to compare the photograph compositions, we calculated the camera position and orientation (quaternion form) when each photo was taken. This information was recovered with the help of VisualSFM [64], [65] for each photo. We computed the camera position change (Δt) and rotation change (Δr) between each user taken photo and the corresponding ground truth photo. These changes were categorized by methods, and tested with two Matched Pairs t Tests respectively on the population mean differences of Δt and Δr. The results show that the data do not provide evidence of significant differences for the two methods on either Δt or Δr ($\mu_{diff\_\Delta t} : DF = 99$, $t = 1.26, P - value = 0.2106, \alpha = 0.05; \mu_{diff\_\Delta r} : DF$=99, $t = 0.78, P - value = 0.4373, \alpha = 0.05$).

Considering the outdoor environment, the drone position and camera orientation were heavily affected during the tasks by the external conditions such as wind, which created randomness to a certain extent. As fig. 9 shows, the participants were using the same standard to recover the photos with the two tested methods and we did not expect and observe similarity difference in photo composition in the study.

2) Timing: Whereas both methods can achieve the same result, an important measure of the suitability of each method it the actual time spent in achieving this goal. The mean time spending of the 20 participants by using FlyCam method is 422.35 second with a standard deviation of 88.23. The mean time spending of the 20 participants using traditional RC method is 453.95 second with a standard deviation of 111.16.

A Matched Pairs t-Test on the population mean difference of time spent between the two tested methods shows the data provide evidence that there is a significant difference between the time spent on task completion using the two methods ($DF = 19$, $t = -2.10, P - value = 0.0496, \alpha = 0.05$). Based on this, we conclude that FlyCam method shows a better efficiency than the traditional RC method in photo composition tasks. This can be attributed to the fact that FlyCam method combines the aircraft motion and camera operation effectively, which makes the drone reach the target zone more quickly. Also, FlyCam method makes the fine tuning process easier and saves a lot of unnecessary camera pose adjustments. The shooting positions of the five ground truth photos in the experiment are relatively independent to each other. FlyCam method can work more efficiently in continuous scenes for view selections.

3) System Usability Scale: The System Usability Scale (SUS) [60] is widely applied reliable tool for measuring the usability. The SUS consists of 10 item on 5 Likert scale response (strongly disagree, disagree, neutral, agree, and strongly agree) questionnaire in our post scenario survey for both tested methods. The questions we asked were:

1) I think that I would like to use this method frequently.
2) I found this method unnecessarily complex.
3) I thought this method was easy to use.
4) I think that I needed or would need help to recall the usage of this method.
5) I found the various human-drone interactions in this method were well integrated.
6) I thought there was too much inconsistency (unexpected drone poses/behaviors) in this method.
7) I would imagine that most people would learn to use this method very quickly.
8) I found this method very cumbersome to use.
9) I felt very confident using this method.
10) I needed to learn a lot of things before I could get going with this method.

We applied the scoring system suggested by Brooke *et al*. [60]. The mean score of FlyCam method was 75.13, which is higher than the overall score of the traditional RC method that was 67.38. A research on 3500 SUS surveys within 273 studies [66] gave out a total mean score of 69.5, which shows that FlyCam framework is above average and therefore better than the traditional RC method from the system usability perspective.

From the responses of question 7 and question 10, 47.5% of the participants highly agreed that FlyCam method can be learned quickly and easily, while only 35% thought so for the traditional RC method. This reflects that the learning curve of FlyCam method is less steep than the RC method to more users. Moreover, once the user was comfortable with the operations, FlyCam method gains more fidelity. After getting familiar with the methods and completing the tasks, 85% of the participants preferred to use FlyCam method frequently basing on question 1 response.

4) The NASA Task Load Index: Besides system usability, we also evaluated the user workload. The NASA Task Load Index (NASA-TLX) [61], [62] is a subjective multidimensional assessment tool to rate the workload of tasks or system. Our post scenario survey includes NASA-TLX rating scales due to the essence of our research being UAV operations. The workload is detached into six factors in NASA-TLX, which are 1) Mental Demand (MD), 2) Physical Demand (PD), 3) Temporal Demand (TD), 4) Overall Performance (OP), 5) Effort (EF), and 6) Frustration (FR). The overall workload score of FlyCam method is around 36, which is four points less than 40, which is

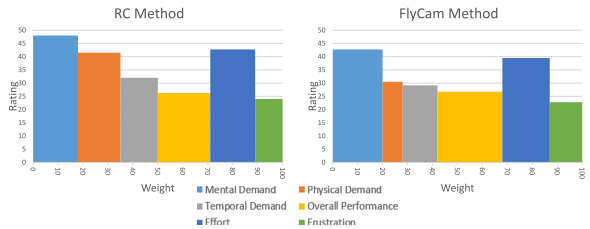

Fig. 10. Comparison of the calculated workload index average weighted rating scores. FlyCam shows lower workload in MD, PD, TD, EF, and FR, whereas the traditional RC method shows a slightly (0.5) lower workload in OP.

the workload score of the traditional RC method. The comparison of the calculated workload index average weighted rating scores is shown as Fig. 10. It shows that except for the OP every other factors of FlyCam method have a lower rate than the traditional RC method. However, the difference in OP is not statistically significant: ($DF = 19, t = -0.3996, P - value = 0.6939, \alpha = 0.05$). With FlyCam method, user shifted more attention from PD to MD and OP. This phenomenon was also evidenced by a study on large multi-touch Ground Control Station of UAVs [51]. The multi-touch gestures free the users from monotonous and repetitive physical operations and allows them to put more efforts in thinking and getting better performance on photo composition. The NASA-TLX rating scores indicates that FlyCam method had lower workload to the participants than the traditional RC method for the drone photography tasks.

VII. Conclusions

We introduced FlyCam, a novel framework that enables users to easily take photographs with drones equipped with gimbal. Our key contribution is in decoupling the flight from the camera operations. The user simply navigates the drone as if it was a flying camera capable of free movements in 3D space and FlyCam framework takes care of the drone and camera control. We introduced six simple touch gestures to utilize this unified control model. We further introduced several novel techniques such as mitigating the swaying of the drone by using sigmoidal velocity control and moving the gimbal in sync with the drone rotation.

We evaluated our system with a user study, where the users were asked to replicate given photographs. Our evaluation shows that FlyCam method outperforms the traditional two joystick control in terms of readiness of completion and easiness of usage. FlyCam method also scored higher in the NASA Task Load Index [61], [62] as well as in System Usability Scale [60].

Our system has several *limitations*. First, there is a communication delay caused by the hardware that causes lagging of the response. We assume this will be addressed by new drones and in a new version of the SDK. Second, the tap hold gesture is not accepted naturally by all users. Two of the users habitually applied double tap hold instead of when they actually intent to do a single tap hold, potentially due to the habit on mouse left button double-click. Besides, the delayed response of tap hold gestures makes the drone position adjustment in close range jerky. A potential replacement gesture could be a pinch, which can zoom in and out the view by driving the drone closer or further to the camera view center along the optical axis. Third, while we aimed at keeping the number of touch gestures minimal, it could be possible to extend the number of gestures as it is not obvious what a good small number of gestures would be.

There are several possible avenues for *future work*. The FlyCam has been tested only on one drone equipped with gimbal camera and it would be interesting to see how this approach can be generalized to different drones. Another future work would include comparison of the gestures on tablets of different sizes. We have observed in our user study that it is not always intuitive for the users to make the mental mapping of the screen size to the desired action of the drone. Another future work would be to include left-handed subjects. The gestures are symmetrical and it should be easy to consider.

Acknowledgements

The authors would like to thank S. D. Rajasekaran for the help on video editing, and B. Smith for useful discussions. The authors also would like to thank the anonymous reviewers for valuable feedback.

References

[1] M. Quigley, M. A. Goodrich, S. Griffiths, A. Eldredge, and R. W. Beard, "Target acquisition, localization, and surveillance using a fixed-wing mini-UAV and gimbaled camera," in *Proc. 2005 IEEE Int. Conf. Robot. Automat.*, 2005, pp. 2600–2605.

[2] O. C. Jakobsen and E. N. Johnson, "Control architecture for a UAV-mounted pan/tilt/roll camera gimbal," in *Proc. AIAA Guidance, Infotech@ Aerospace*, Arlington, VA, USA, 2005, p. 7145.

[3] D. Lee, V. Chitrakaran, T. Burg, D. Dawson, and B. Xian, "Control of a remotely operated quadrotor aerial vehicle and camera unit using a fly-the-camera perspective," in *Proc. 46th IEEE Conf. Decis. Control*, 2007, pp. 6412–6417.

[4] A. E. Neff, D. Lee, V. K. Chitrakaran, D. M. Dawson, and T. C. Burg, "Velocity control for a quad-rotor UAV fly-by-camera interface," in *Proc. IEEE SoutheastCon*, 2007, pp. 273–278.

[5] Yuneec, "*SkyView User Manual V1.0*," Aug. 2016. [Online]. Available: http://us.yuneec.com/skyview-goggles

[6] DJI, "*DJI Goggles User Guid V1.2*," Aug. 2017. [Online]. Available: https://www.dji.com/dji-goggles

[7] DJI, "*Phantom 4 Pro/Pro+ User Manual V1.6*," Jul. 2017. [Online]. Available: https://www.dji.com/phantom-4-pro

[8] 3D-Robotics, "*SOLO User Manual V9.02.25.16*," Mar. 2017. [Online]. Available: https://3dr.com/solo-drone

[9] Parrot, "*Parrot AR.Drone 2.0 User Guide*," Apr. 2012. [Online]. Available: http://ardrone2.parrot.com

[10] Ehang, "*GhostDrone 2.0 Ehang Play App Manual*," Dec. 2016. [Online]. Available: http://www.ehang.com/ghost2.0.html

[11] J. M. Ettikkalayil, "Design, implementation, and performance study of an open source eye-control system to pilot a Parrot AR. drone quadrocopter," Master's thesis, Dept. of Biomed. Eng., City Univ. New York, NY, USA, Nov. 2013.

[12] J. P. Hansen, A. Alapetite, I. S. MacKenzie, and E. Møllenbach, "The use of gaze to control drones," in *Proc. Symp. Eye Tracking Res. Appl.*, 2014, pp. 27–34.

[13] A. C. Trujillo, J. Puig-Navarro, S. B. Mehdi, and A. K. McQuarry, "Using natural language to enable mission managers to control multiple heterogeneous UAVs," in *Proc. Amer. Inst. Steel Construction*, 2017, pp. 267–280.

[14] M. Landau and S. van Delden, "A system architecture for hands-free UAV drone control using intuitive voice commands," in *Proc. ACM/IEEE Int. Conf. Human-Robot Interaction*, 2017, pp. 181–182.

[15] K. LaFleur, K. Cassady, A. Doud, K. Shades, E. Rogin, and B. He, "Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain–computer interface," *J. Neural Eng.*, vol. 10, no. 4, 2013, Art. no. 046003.

[16] Y. Yu *et al.*, "Flyingbuddy2: A brain-controlled assistant for the handicapped." in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2012, pp. 669–670.

[17] W. S. Ng and E. Sharlin, "Collocated interaction with flying robots," in *Proc. IEEE Int. Conf. Robot Human Interactive Commun.*, 2011, pp. 143–149.

[18] K. Pfeil, S. L. Koh, and J. LaViola, "Exploring 3D gesture metaphors for interaction with unmanned aerial vehicles," in *Proc. Int. Conf. Intell. User Interface*, 2013, pp. 257–266.

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

[20] A. Sarkar, K. A. Patel, R. G. Ram, and G. K. Capoor, "Gesture control of drone using a motion controller," in *Proc. Int. Conf. Ind. Informat. Comput. Syst.*, 2016, pp. 1–5.

[21] M. Chandarana, A. Trujillo, K. Shimada, and B. D. Allen, "A natural interaction interface for UAVs using intuitive gesture recognition," in *Proc. Amer. Inst. Steel Construction*, 2017, pp. 387–398.

[22] J. M. Teixeira, R. Ferreira, M. Santos, and V. Teichrieb, "Teleoperation using Google Glass and AR, drone for structural inspection," in *Proc. 14th Symp. Virtual Augmented Reality*, 2014, pp. 28–36.

[23] L. A. Sandru, M. F. Crainic, D. Savu, C. Moldovan, V. Dolga, and S. Preitl, "Automatic control of a quadcopter, AR. drone, using a smart glove," in *Proc. Int. Conf. Control, Mechatronics Automat.*, 2016, pp. 92–98.

[24] J. Nagi, A. Giusti, G. A. Di Caro, and L. M. Gambardella, "Human control of UAVs using face pose estimates and hand gestures," in *Proc. ACM/IEEE Int. Conf. Human-Robot Interaction*, 2014, pp. 252–253.

[25] J. Nagi, A. Giusti, L. M. Gambardella, and G. A. Di Caro, "Human-swarm interaction using spatial gestures," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 3834–3841.

[26] M. Lichtenstern, M. Frassl, B. Perun, and M. Angermann, "A prototyping environment for interaction between a human and a robotic multi-agent system," in *Proc. ACM/IEEE Int. Conf. Human-Robot Interaction*, 2012, pp. 185–186.

[27] V. M. Monajjemi, J. Wawerla, R. Vaughan, and G. Mori, "HRI in the sky: Creating and commanding teams of UAVs with a vision-mediated gestural interface," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 617–623.

[28] J. R. Cauchard, J. L. E, K. Y. Zhai, and J. A. Landay, "Drone & me: An exploration into natural human-drone interaction," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2015, pp. 361–365.

[29] M. Obaid, F. Kistler, G. Kasparavičiūtė, A. E. Yantaç, and M. Fjeld, "How would you gesture navigate a drone?: A user-centered approach to control a drone," in *Proc. 20th Int. Academic Mindtrek Conf.*, 2016, pp. 113–121.

[30] P. Abtahi, D. Y. Zhao, L. Jane, and J. A. Landay, "Drone near me: Exploring touch-based human-drone interaction," in *Proc. Interactive, Mobile, Wearable Ubiquitous Technol.*, 2017, vol. 1, no. 3, p. 34.

[31] J. L. E, I. L. E, J. A. Landay, and J. R. Cauchard, "Drone & Wo: Cultural influences on human-drone interaction techniques," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2017, pp. 6794–6799.

[32] E. Peshkova, M. Hitz, and D. Ahlström, "Exploring user-defined gestures and voice commands to control an unmanned aerial vehicle," in *Proc. Int. Conf. Intell. Technol. Interactive Entertainment*, 2017, pp. 47–62.

[33] R. A. S. Fernández, J. L. Sanchez-Lopez, C. Sampedro, H. Bavle, M. Molina, and P. Campoy, "Natural user interfaces for human-drone multi-modal interaction," in *Proc. Int. Conf. Unmanned Aircr. Syst.*, 2016, pp. 1013–1022.

[34] A. O. Abioye *et al.*, "Multimodal human aerobotic interaction," in *Smart Technol. Appl. Bus. Environ.*, 2017, pp. 39–62.

[35] E. Peshkova, M. Hitz, and B. Kaufmann, "Natural interaction techniques for an unmanned aerial vehicle system," *IEEE Pervasive Comput.*, vol. 16, no. 1, pp. 34–42, Jan. 2017.

[36] J. Fleureau, Q. Galvane, F.-L. Tariolle, and P. Guillotel, "Generic drone control platform for autonomous capture of cinema scenes," in *Proc. Workshop Micro Aerial Veh. Netw. Syst. Appl. Civilian Use*, 2016, pp. 35–40.

[37] N. Joubert, M. Roberts, A. Truong, F. Berthouzoz, and P. Hanrahan, "An interactive tool for designing quadrotor camera shots," *ACM Trans. Graph.*, vol. 34, no. 6, 2015, Art. no. 238.

[38] M. Roberts and P. Hanrahan, "Generating dynamically feasible trajectories for quadrotor cameras," *ACM Trans. Graph.*, vol. 35, no. 4, 2016, Art. no. 61.

[39] C. Gebhardt, B. Hepp, T. Nägeli, S. Stevšić, and O. Hilliges, "Airways: Optimization-based planning of quadrotor trajectories according to high-level user goals," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2016, pp. 2508–2519.

[40] T. Ngeli, 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 Robot. Autom. Letters*, vol. 2, no. 3, pp. 1696–1703, Jul. 2017.

[41] T. Nägeli, L. Meier, A. Domahidi, J. Alonso-Mora, and O. Hilliges, "Real-time planning for automated multi-view drone cinematography," *ACM Trans. Graph.*, vol. 36, no. 4, Jul. 2017, Art. no. 132.

[42] Yuneec, "*Typhoon H User Manual RS V1.2*," Sep. 2016. [Online]. Available: http://us.yuneec.com/typhoon-h-overview

[43] S. Buchanan, B. Floyd, W. Holderness, and J. J. LaViola, "Towards user-defined multi-touch gestures for 3D objects," in *Proc. ACM Int. Conf. Interactive Tabletops Surf.*, 2013, pp. 231–240.

[44] C.-J. KUa and L.-C. Chen, "A study on the natural manipulation of multi-touch gestures for 3D object rotation using a large touch screen," in *Universal Design 2014: Three Days of Creativity and Diversity: Proc. UD*, vol. 35. Amsterdam, The Netherlands: IOS Press, 2014, p. 279.

[45] D. Mendes, M. Sousa, A. Ferreira, and J. Jorge, "Thumbcam: Returning to single touch interactions to explore 3D virtual environments," in *Proc. ACM Int. Conf. Interactive Tabletops Surf.*, 2014, pp. 403–408.

[46] D. Fiorella, A. Sanna, and F. Lamberti, "Multi-touch user interface evaluation for 3D object manipulation on mobile devices," *J. Multimodal User Interfaces*, vol. 4, no. 1, pp. 3–10, Mar. 2010.

[47] F. R. Ortega, "3D navigation with six degrees-of-freedom using a multi-touch display," Ph.D. dissertation, Dept. of Comput. Sci., Florida Int. Univ., Miami, FL, USA, Nov. 2014.

[48] J. Jankowski, T. Hulin, and M. Hachet, "A study of street-level navigation techniques in 3D digital cities on mobile touch devices," in *Proc. IEEE Symp. 3D User Interfaces*, 2014, pp. 35–38.

[49] F. D. Crescenzio, G. Miranda, F. Persiani, and T. Bombardi, "A first implementation of an advanced 3D interface to control and supervise UAV (uninhabited aerial vehicles) missions," *Presence: Teleoperators Virtual Environ.*, vol. 18, no. 3, pp. 171–184, 2009.

[50] J. Haber, "Enhancing the functional design of a multi-touch UAV ground control station," Master's thesis, Dept. of Aerospace Eng., Ryerson Univ., Toronto, ON, Canada, Jan. 2015.

[51] J. Haber and J. Chung, "Assessment of UAV operator workload in a reconfigurable multi-touch ground control station environment," *J. Unmanned Vehicle Syst.*, vol. 4, no. 3, pp. 203–216, 2016.

[52] G. Paravati, A. Sanna, F. Lamberti, and C. Celozzi, "A reconfigurable multi-touch framework for teleoperation tasks," in *Proc. IEEE Int. Conf. Emerg. Technol. Factory Autom.*, 2011, pp. 1–4.

[53] Y. Sugiura *et al.*, "An operating method for a bipedal walking robot for entertainment," in *Proc. SIGGRAPH ASIA*, 2009, pp. 79–79.

[54] M. Micire, J. L. Drury, B. Keyes, and H. A. Yanco, "Multi-touch interaction for robot control," in *Proc. Int. Conf. Intell. User Interface*, 2009, pp. 425–428.

[55] Y.-L. Chen, W.-T. Lee, L. Chan, R.-H. Liang, and B.-Y. Chen, "Direct view manipulation for drone photography," in *Proc. SIGGRAPH Asia*, 2015, p. 23.

[56] L. Gross, "Multi-touch through-the-lens drone control," Master's thesis, Dept. of Elect. Eng. and Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, Jun. 2016.

[57] Z. Lan, M. Shridhar, D. Hsu, and S. Zhao, "Xpose: Reinventing user interaction with flying cameras," in *Proc. Robot. Sci. Syst.*, Cambridge, MA, USA, Jul. 2017.

[58] DJI, "DJI mobile SDK for android," 2017. [Online]. Available: https://github.com/dji-sdk/Mobile-SDK-Android

[59] DJI, "DJI uilibrary for android," 2017. [Online]. Available: https://github.com/dji-sdk/Mobile-UILibrary-Android

[60] J. Brooke *et al.*, "Sus-a quick and dirty usability scale," *Usability Eval. Ind.*, vol. 189, no. 194, pp. 4–7, 1996.

[61] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (task load index): Results of empirical and theoretical research," *Adv. Psychol.*, vol. 52, pp. 139–183, 1988.

[62] S. G. Hart, "Nasa-Task load index (NASA-TLX); 20 years later," *Proc. Human Factors Ergonom. Soc. Annu. Meeting*, vol. 50, no. 9., pp. 904–908, 2006.

[63] A. J. Yu and J. D. Cohen, "Sequential effects: Superstition or rational behavior?" in *Proc. Adv Neural Inf Process Syst*, 2009, pp. 1873–1880.

[64] C. Wu., "Visualsfm: A visual structure from motion system," 2011. [Online]. Available: http://ccwu.me/vsfm/

[65] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, "Multicore bundle adjustment," in *Proc. IEEE Conf., Comput. Vision Pattern Recognit.*, 2011, pp. 3057–3064.

[66] A. Bangor, P. Kortum, and J. Miller, "Determining what individual SUS scores mean: Adding an adjective rating scale," *J. Usability Stud.*, vol. 4, no. 3, pp. 114–123, 2009.