ARPilot: Designing and Investigating AR Shooting Interfaces on Mobile Devices for Drone Videography

Yu-An Chen [1], Te-Yen Wu [1], Tim Chang [2], Jun You Liu [1], Yuan-Chang Hsieh [1], Leon Yulun Hsu [1], Ming-Wei Hsu [1], Paul Taele [3], Neng-Hao Yu [4], Mike Y. Chen [5]
National Taiwan University [1,5], University of California, Santa Barbara [2],
Texas A&M University [3], National Taiwan University of Science and Technology [4]
[1] {ryan149347,teyanwu,junyouliu9,icefich990729,leonorz123,chad1023}@gmail.com,
[2] tinghaur@umail.ucsb.edu, [3] ptaele@cse.tamu.edu, [4] jonesfish@gmail.com, [5] mikechen@csie.ntu.edu.tw

ABSTRACT
Drones offer camera angles that are not possible with traditional cameras and are becoming increasingly popular for videography. However, flying a drone and controlling its camera simultaneously requires manipulating 5-6 degrees of freedom (DOF) that needs significant training. We present ARPilot, a direct-manipulation interface that lets users plan an aerial video by physically moving their mobile devices around a miniature 3D model of the scene, shown via Augmented Reality (AR). The mobile devices act as the viewfinder, making them intuitive to explore and frame the shots. We leveraged AR technology to explore three 6DOF video-shooting interfaces on mobile devices: *AR keyframe*, *AR continuous*, and *AR hybrid*, and compared against a traditional touch interface in a user study. The results show that *AR hybrid* is the most preferred by the participants and expends the least effort among all the techniques, while the users' feedback suggests that *AR continuous* empowers more creative shots. We discuss several distinct usage patterns and report insights for further design.

ACM Classification Keywords
H.5.1 User Interfaces: Artificial, augmented, and virtual realities; H.5.2 User Interfaces: Interaction styles

Author Keywords
Interaction techniques; human-drone interaction; augmented reality; virtual camera control; mobile device; tangible.

INTRODUCTION
The growing affordability of commercial drones and high-resolution cameras have enabled greater accessibility for people to pursue aerial videography. With the portability and maneuverability of camera-equipped drones, videographers can potentially capture professional-looking outdoor cinematic video scenes. Conventionally, users fly a drone and shoot aerial videos by using a dual-stick remote controller, which require a non-trivial level of expertise and dexterity to achieve decent quality scenes. To address these control challenges, researchers and manufacturers have proposed a variety of software solutions (e.g., [2, 3, 7, 8, 17, 19]) that simplify drone controls by leveraging smartphone and tablet devices, including those that introduce navigation and video shooting in virtual 3D environments.

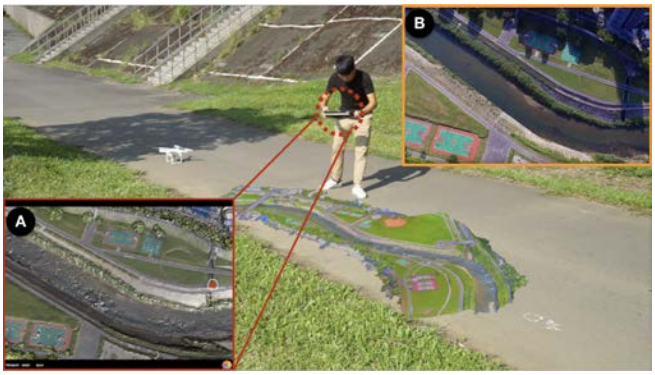

Figure 1. ARPilot is a direct-manipulation tool that facilitates route planning for aerial drones. The user directly views the mobile device's display as a viewfinder of the drone's camera, and physically moves the mobile device as the drone. (A) ARPilot emulated as a camera-like interface. (B) Actual shooting image from (A).

Mobile AR [1] takes advantage of SLAM (Simultaneous Localization And Mapping) technology and has become increasingly more popular on mobile devices. We utilize this technology to transform a tablet as a tangible camera controller. Users can take images and videos around virtual 3D models by physically moving and rotating their mobile devices. This type of interaction can offer a more natural experience for users performing drone videography, since it facilitates the complexity of six degrees-of-freedom (6DOF) manipulations through directly video cameras control. From these contexts, we propose ARPilot as a novel solution that leverages AR technology for controlling drones in aerial videography. We designed and implemented three proposed types of AR video-shooting interfaces specifically for drone

MobileHCI '18, September 3–6, 2018, Barcelona, Spain
© 2018 ACM. ISBN 978-1-4503-5898-9/18/09…$15.00
DOI: https://doi.org/10.1145/3229434.3229475

videography: *AR keyframe*, which takes keyframe photos for a drone path with spatial interactions; *AR hybrid*, which expands from *AR keyframe* by offering touch gesture for users to translate or rotate the 3D model; and *AR continuous*, which continuously records an arbitrary camera path by physically moving their devices.

We evaluate our three proposed AR interface types against a conventional touch interface for improving our understanding of how users interact with these techniques for drone videography. From our results, we demonstrated that *AR hybrid* was the most preferred and the most effortless by all our participants, and that *AR continuous* required less time to complete most of our evaluated tasks. Our user feedback also suggests that *AR continuous* supports users in better producing more creative shots. In summary, our work's contributions include the following: (i) an intuitive flight planning tool taking advantage of mobile AR for drone videography, (ii) a user study that compares three AR interfaces and a conventional touch input for taking aerial shots, and (iii) insights and recommendation for the design of AR-based videography tools.

RELATED WORK

Human-drone interaction

Drones have been used in research projects for taking selfies [15, 18], filming cinematography [21, 25, 26, 29], providing rapid and flexible public display environments [28, 30], serving as a readily-available companion [23, 24, 32, 33], and so on. Here we focus on investigating the interaction techniques for aerial videography.

Mission planner

Conventional tools such as Ground Station Pro [3], Litchi [7], and Mission Planner [8] provide a 2D map for users to plot the drone's waypoints of selected desired locations and require users to set up all the detailed parameters including GPS location, altitude, shooting angle and speed. Although the complex flight missions can be planned with a few taps, users have to imagine the most likely shots from each waypoint. Crescenzio et al. [17] proposed a system that contains a command panel and a vertical display. The command panel allows users to send high-level commands to UAVs and the vertical display offers a pilot view or an external view of the operated UAV with augmented visualization for better perceiving the UAV's current physical state. Copilot [2] let users plan the desired shot by directly manipulating a 3D view from Google Earth [5]. It utilizes touch gesture interactions for camera control and keyframe setting so that the users can focus on framing a shot instead of setting the drone's flight control. Skywand [10] introduced an immersive VR system that allows users to control and plan the drone's views around a virtual city with two handheld controllers: one for navigation and one for the drone's point-of-view (POV). We aim to provide a similar analogy that allows users to walk in the 3D scene like a giant and frame the shot as they hold a real camera in the sky by simply using a mobile device and AR technology.

Trajectory planning

Horus [21, 29] is an interactive tool for designing aerial shots by specifying advanced controls on a 2D map and a 3D view. The system is able to calculate a feasible trajectory for the drone and then executes the videography mission autonomously. Airways [19] enables users to directly plan the drone's flight trajectory by drawing a 3D path, and also assists the user in optimizing the trajectory to ensure feasibility and smoothness. Lastly, Nägeli et al. [25, 26] proposed a real-time motion planning system for aerial videography. The system takes high-level plans as input such as types of shot sizes or shot composition and then generates collision-free trajectories for shooting close-proximity videos in dynamic and cluttered indoor environments. We take insights from these tools and allow users to frame shots from a mobile device's camera. Our system then calculates a feasible trajectory to capture the desired video clips.

Tangible and Touch Camera Controls

Researchers have also investigated the benefits of navigational controls in 3D space, including for tangible user interfaces and their comparisons to traditional computing input. Such controls take advantage of humans' evolved abilities to grasp and manipulate physical objects, and empower users to navigate within the digital world more naturally.

3D object manipulation and data exploration

Marzo et al. [22] compared three manipulation techniques which employ multi-touch, device position and a combination of both to move and rotate a virtual object on the screen. They found that using only the device movement and orientation is more intuitive but combining multi-touch and device movement yields the best efficiency. Furthermore, they observed that applying orientation on input devices allowed for more accuracy in the output object's rotation when users desired smaller rotations. Besancon et al. [13] compared tangible input to alternative mouse and tactile controls, and reported that users performed spatial navigation tasks with tangible input more quickly with similar levels of accuracy.

3D data exploration

Besancon et al. [12] demonstrated that tactile and tangible input together benefitted from interaction precision as well as integrated, multi-sensory, and intuitive 6DOF control, due to their similarity to day-to-day interaction with real objects. Buschel et al. [14] reported that users perceived spatial interaction as more supportive, comfortable, and preferred over touch interaction for navigating 3D data exploration.

Peephole navigation on mobile devices

Hurst et al. [20] reported that users preferred and performed better with dynamic peephole navigation to view a VR panorama scene. Spindler et al. [31] proposed a novel concept for interacting with virtual 3D information spaces that combined tangible interaction, head tracking, and multi-touch techniques, which allowed for users to perform 3D interaction tasks in a more accessible manner for tabletop environments. Arvola et al. [11] reported that users found panning—or spatial interaction along the horizontal plane—in

peephole navigation more engaging than touch interaction for mobile device panoramas.

From these prior works, we discovered that there has been lacking in-depth research regarding the use of AR and touch methods to conduct drone planning. Therefore, we will analyze how users interact with these interaction methods and provide insights for developing an aerial videography planning tool with greater user-friendliness.

ARPILOT

Our goal is to allow users to intuitively and easily perform aerial videography without having prior training in flying drones. We propose ARPilot for transforming the mobile device's display as a viewfinder of the drone's camera by leveraging mobile AR technology. As a result, we designed and implemented three proposed AR interfaces: *AR keyframe*, *AR hybrid*, and *AR continuous*.

AR Keyframe Interface

AR keyframe is a pure AR interface that allows users to physically navigate the drone via the camera, and to also take a sequence of keyframe photos at desired positions and angles in 3D space. Each keyframe photo is generated and suspended at the shooting point to inform users of previous locations. After users complete capturing keyframe photos, the interface will connect all keyframe photos as a continuous camera path, which the drone will then follow for its flight. In this design, users can utilize their eye-hand coordination to manipulate the drone's camera for shooting videos as easily as with handheld video cameras.

AR Hybrid Interface

Research work has shown that touch interaction can alleviate a user's physical effort in navigating a camera to a location [14]. From this insight, we incorporated touch interaction into our design to create the *AR hybrid* interface. Users can not only spatially move and rotate the camera, but can also manipulate the translation and orientation of the displayed 3D models with touch gesture interaction. Based on Google Map's gesture design [6], we provide the following three touch gestures:

- One finger to drag the virtual camera's position.
- Two fingers to rotate the virtual camera's y-axis.
- Two fingers to perform zoom-in or zoom-out gesture for translating the virtual lens in the z-axis.

With the exception of gestures, the design aspects of *AR hybrid* are equivalent to *AR keyframe*.

AR Continuous Interface

Inspired by the idea that users can simultaneously and continuously interact with 6DOF via AR technology, we designed the *AR continuous* interface to enable users to create an arbitrary and continuous camera path in the 3D scene. For this interface, our system automatically records keyframes every 0.5 seconds, but the keyframes are not suspended at the captured locations in order to prevent overcrowding in the virtual environment. From this approach, users emulate the role of the flying drone to shoot video around the 3D model.

Implementation

We developed ARPilot based on ARKit [1] with the Unity engine. Our workflow can be found in Figure 2.

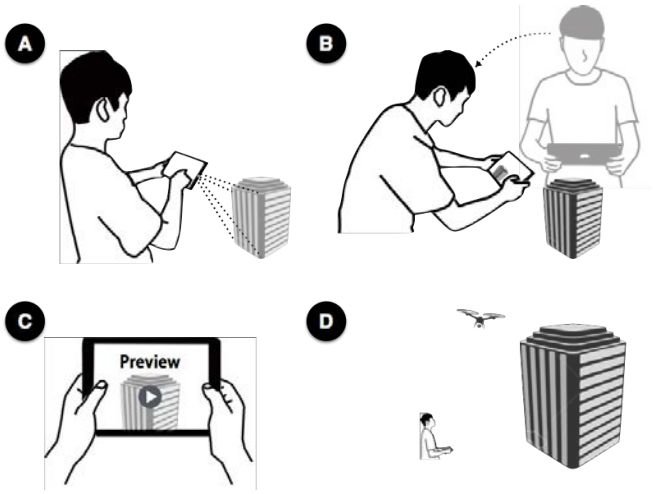

Figure 2. ARPilot consists of the following four steps: (A) Place model on the surface being sensed. (B) Manipulate drone's camera on our interfaces for planning videography. (C) Preview before starting drone mission. (D) Plan drone flights at the path.

Model placement

Initially, our system requires users to import a 3D scene for performing their desired drone videography. The 3D scene can be built in Pix4D [9] or loaded from Google Earth [5]. Afterwards, the system automatically performs plane detection as provided by ARKit, and then places the 3D scene on a horizontal surface. We also provide scaling functions for users to adjust the model to the appropriate size.

Path routing

For *AR keyframe* and *AR hybrid*, our system will route camera paths by linearly interpolating the position and rotation for all keyframes. However, in the case of *AR continuous*, this approach would generate a shaky video, because tremors or jitters may occur while users manipulate the device. Therefore, we further applied a midpoint smoothing algorithm [16] to *AR continuous* for stabilizing this shakiness.

Safety tips

To ensure greater safety for practical flight, ARPilot implemented two safety tips for alerting users of invalid control scenarios. First, in the planning step, the system will forbid users from capturing a keyframe or recording if the drone crosses the model's boundary. This is because the system cannot ensure whether there is a building or an object outside the boundary. Second, after a flight path has been created, the system will check if there is a collision occurrence by the virtual camera and the model during path navigation. That is, if the user captures the keyframe or records outside the models, we cannot guarantee the safety of the drone. To prevent such danger, ARPilot will forbid camera recording and will display warning messages while invalid control is actively occurring.

Video preview

In ARPilot, we developed a flight simulator for users to preview the drone videography. To simplify the interaction procedure and rapidly present the simulation in front of users, we designed ARPilot to reduce the amount of unnecessary setup on the users. The velocity of the simulated drone is set at a constant 10 meters per second, and the angular velocity is dependent on the linear interpolation. When yaw rotation is necessary, the minimum clockwise or counterclockwise rotation is calculated. All simulations are saved into a video gallery for later use. Although the wind conditions and GPS accuracies deviate slightly between preview and actual shots, the simulator can serve as a helpful tool for previewing the actual drone videography.

Drone mission

The current implementation of ARPilot supports *DJI phantom 3* as an aerial vehicle, which we developed by using DJI Mobile SDK [4]. Once the user confirms execution of the preview shots, ARPilot convert the keyframe contents into the waypoint mission's compatible format, including:

- **Drone coordinates.** Longitude and latitude (WGS84 format) is converted from 3D spatial coordinates. When a 3D model is imported, the original scale as well as the longitude/latitude of the model's origin point is included. Real-world coordinates are calculated based on this information.
- **Drone heading and camera pitch angle (gimbal).** While the camera on mobile device is initialized, ARPilot defines the camera-facing direction as true north. After the virtual model is placed, the model's true north may not match that of the mobile device's camera. Therefore, the correct heading of the drone is configured as the relative direction between the virtual camera and 3D model. The pitch angle of the drone's camera is determined by the virtual camera's x-axis rotation.

After the conversion is complete, the drone adapts the settings as mentioned in the simulation, flies at a constant 10 meter per second, and uses minimum distance rotation when yaw rotation is required. In order to prevent unnecessary rotation of the drone, ARPilot first compares the angles of one keyframe to its subsequent keyframe, then adopts the direction that requires lesser rotation.

USER STUDY

For our user study, our goals include understanding: 1) how AR interfaces can assist users in shooting aerial drone videos, and 2) the benefits and weaknesses of AR interfaces compared to traditional touch interfaces. We selected a within-subject design consisting of two independent variables (4×5 factorial design): 1) *Interaction techniques*, which represent touch, *AR keyframe*, *AR continuous*, and *AR hybrid*; and 2) *Task*, which consists of five different fundamental aerial shots.

Task Design

We consulted an experienced drone operator on our task design. This expert was instructed to first analyze twenty award-winning videos from a relevant drone videography competition [1], and then discover fundamental shots that were frequently used from the analyzed videos. The fundamental shots were categorized into five different tasks that are based on the following drone operations (Figure 3): forward, pullback, sideway sliding, panorama, and orbiting.

Forward

The drone flies forward in a straight path, with its camera angled downward at 30°. This task requires two keyframes by changing the drone's position along one axis only.

Pullback

The drone pulls back and upward away from the scene, with its camera angled downward at 45°. This task requires two keyframes by changing the drone's position along two axes.

Sideway sliding

The drone flies sideways along the model at a yaw rotation of 50°, with its camera angled downward at 45°. This task requires two keyframes by changing the drone's position and heading at the same time.

Panorama

The drone is rotated 180° at the yaw axis in a stationary position, with its camera capturing a wild-angle view. This task requires three keyframes by changing the drone's heading only.

Orbiting

The drone flies in a trajectory around the center target, with its camera facing inward and angled downward at 50°. This task requires nine keyframes, which includes eight path vertices and an additional final path vertex that overlaps the initial path vertex.

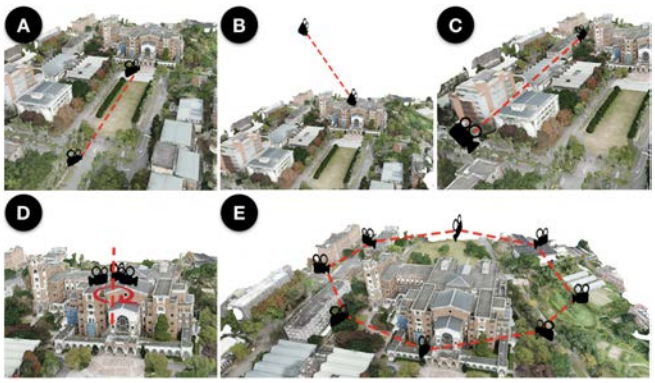

Figure 3. Our five categories for aerial drone videography: (A) forward, (B) pullback, (C) sideway sliding, (D) panorama, and (E) orbiting.

Participants

We recruited 12 participants—2 females—for our user study from a major university campus. Their ages ranged from 20 to 30 years (M=23.16, SD=1.26). From our background questionnaires provided at the beginning of our study, all participants reported having minimal or no drone operating experience prior to the study and having existing familiarity working with AR environments.

[1] https://www.skypixel.com/events/videocontest2017/winners

Apparatus and Implementation

We provided each participant with a 10.5-inch iPad Pro device for the study. The 3D model that was displayed to participants was built using Pix4D [9], and also modeled the university campus model (Figure 3). Participants used a touch interface similar to the interface designs from Google Maps and Copilots [2], which consisted of four touch gestures to manipulate the virtual camera's position and angle. Three of the gestures—drag, rotate, and pinch-to-zoom—are identical to our AR hybrid interface. The remaining gesture—two-finger slide (pan)—tilts the virtual camera (i.e., rotate based on the camera's horizontal axis) when the user places two fingers together on the screen and simultaneously moves them in parallel.

Procedure

The duration of each of our studies lasted approximately 90 minutes. Participants initially received a background questionnaire, and were then prompted to complete five tasks using our four proposed interfaces. We ordered the sequence of the prompted interfaces using Latin squares to ensure a counterbalanced setup. For each *Interaction technique*, participants were prompted to complete the tasks in the order of: forward, pullback, sideway sliding, panorama, and orbiting. Before each condition, we provided participants a practice period of two minutes in order to familiarize themselves with the interface controls.

Prior to performing their prompted task, participants were first shown a video shot, and then similarly performed that shot with the assigned technique. With the exception of the *AR continuous* interface, we requested that participants confirm the required number of keyframes for each task. To calculate the task completion time, we started timing when we prompted the participant to adjust the camera, and stopped when the participant completed the adjustment. Upon single task completion, we asked participants to grade the similarity and effort of their video task compared to the original video using a 7-point Likert scale. Once the participant completed all tasks with each interaction technique, we gave them a questionnaire with a 7-point Likert scale that prompted their responses for the intuitiveness, simplicity, and satisfaction of their overall performance. After all interaction techniques were completed, we concluded the study with a final questionnaire that prompted participants to both rank by preference and to provide feedback of the four techniques.

RESULT

Performance

We analyzed the completion time of the four conditions for each task. As shown in Figure 4, the overall times of *AR keyframe* (28.9s), *AR hybrid* (30.8s), and *AR continuous* (19.5s) were over 50% faster than *Touch* (64.7s). From the Friedman test and Nemenyi Post-hoc analysis [27], improvement in time was statistically significant ($p<.05$) in all categories among the three AR interfaces and the touch interface.

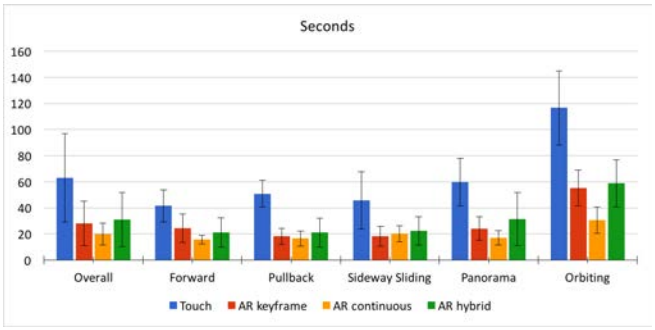

Figure 4. Completion time for each task. *AR continuous* showed the least amount of time required to complete most of the tasks except for Sideway Sliding.

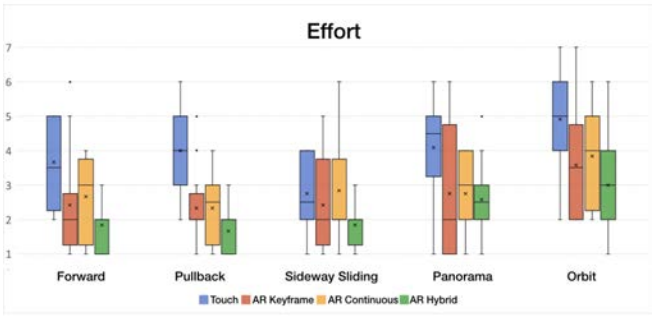

Figure 5. Effort rating of the four interfaces for each task. (lower score represents lower effort).

Similarity

From the Friedman test and Nemenyi Post-hoc analysis, we observed no statistical significance among the interfaces in each of the tasks. The averaged user ratings on a 7-point Likert scale for the path similarity was *AR keyframe (5.36), AR hybrid (5.48), AR continuous (4.95), and Touch (4.67)*, where user ratings for both *AR keyframe* and *AR hybrid* performed with better results than the touch interface. Most participants (8) also expressed that the AR interfaces are easier to make adjustments than the touch interface. Likewise, we discovered that the paths created with the AR interface were easier to maintain at the same height (e.g., Figure 6ABC). On the other hand, the touch interface (e.g., Figure 6D) yielded more scattered keyframes in 3D space, which led to less precise paths.

Effort

The result in Figure 5 demonstrated that the touch interface required the most effort among all the interfaces, and that this effort was statistically significant using the Friedman test and Nemenyi Post-hoc analysis ($p<.05$). Participants pointed out that it was difficult to perform minor detail adjustments using the touch interface. That is, the touch interface required more trials to capture the correct shot in the touch interface. On the other hand, *AR keyframe* and *AR continuous* allowed users to physically adjust the desired location and angle at the same time in order to capture the desired shot. We note however that if the desired locations are too distant from each other,

users may feel more exhausted due to the larger movements necessary to correctly capture the shot. Of the interfaces, *AR hybrid* received the lowest effort for all the tasks, since this interface allowed users to accelerate their wide-range movements with gesture input, while also adjusting slight frame differences with AR interaction.

Preference

We calculated the final ranking for all the conditions from the final questionnaires. The order from most to least preferred was as follows: 1) *AR hybrid*, 2) *AR keyframe*, 3) *AR continuous* and 4) *Touch*. From our results, we discovered that ranking order is relatively similar to overall scores in Figure 7. Although *AR continuous* demonstrated higher intuition than *AR keyframe*, due to the increase in effort, *AR continuous* was less preferable than *AR keyframe* for most participants (9). The results for intuitiveness, simplicity, and satisfaction between the four interfaces were statistically significant ($p<.05$) on the Friedman test and Nemenyi Post-hoc analysis. 11 participants reported preferring *AR hybrid* over the other interfaces, because it combined positive features from both AR element and gesture control.

However, one participant expressed preference towards *AR keyframe* the most due to the satisfaction of its AR interaction: *AR keyframe satisfied most of my needs, so I didn't feel that gesture was necessary* (P7). Most users also disliked the touch interface due to being unable to perform minor detail adjustments. However, *AR hybrid* allowed users to move the mobile computing device (i.e., tablet) to make minor adjustments. For example, feedback from one of the users that was shared among users 5, 7, and 9 expressed: *AR hybrid maintained the benefits of the AR feature for minor adjustments while reducing physical effort, since I did not have to squat down to take closer shots of the model. Instead, I can use a gesture to zoom into the model.*

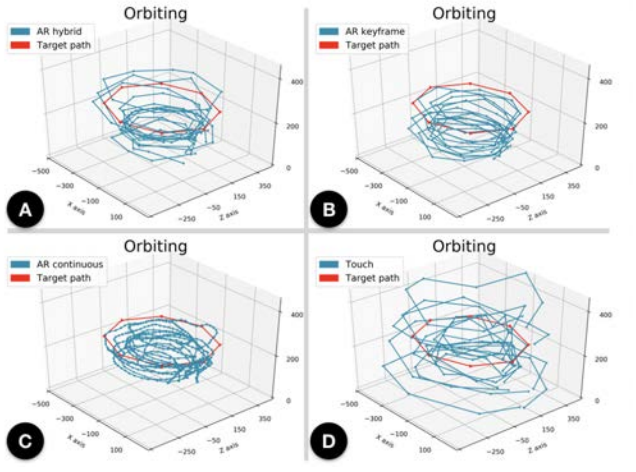

Figure 6. The flight path recorded in the user study: (A) *AR hybrid*, (B) *AR keyframe*, (C) *AR continuous*, and (D) *Touch*.

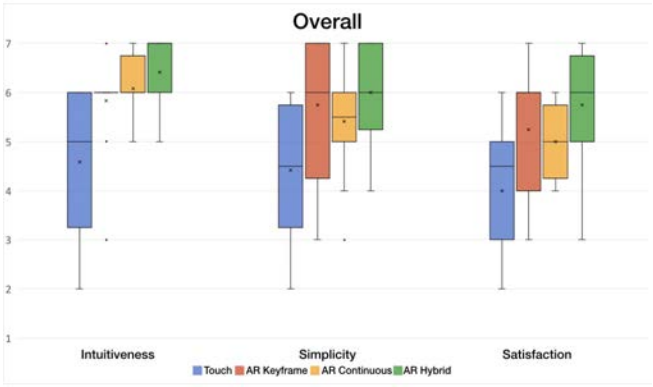

Figure 7. Overall score for each interface. *AR hybrid* was the most favorite from the participants.

DISCUSSION AND FUTURE WORK

Physical Movement Effort

In this work, we demonstrate that our AR interface provides an intuitive way to perform drone videography. However, we observed that users who wish to record closer shots with our AR interface would instinctively squat down in a hunching position in order to achieve more desirable shots. We believe that users assuming this physical posture to record such more desirable closer shots elevates their effort and subsequently become more tired, especially with *AR continuous*. Furthermore, when participants expressed feeling more fatigued when moving laterally (i.e., sliding their bodies side-to-side) to achieve more desirable closer shots. *"It felt more tiring to take a shot at a lower position."* (P4). On the other hand, we observed that *AR hybrid* allowed participants to instead use the gesture input to adjust the height placement of the model, which subsequently reduced their effort. Therefore, we believe that if users are able to adjust the height placement of the model, then their required physical movement effort would decrease while recording, especially for achieving more desirable closer shots.

Creative Tasks

After our user study, we invited six aerial photographers to use ARPilot for taking freeform shots without any constraints. In this freeform shot task, we observed at least several behaviors.

1. In order to achieve their desired shot, some participants frequently walked around the miniature scene and experimented with different viewing angles while using the AR interfaces. Compared to the touch interface, we observed that the AR interfaces not only allowed users to more efficiently create shots with body movements, but also encouraged users to explore the miniature scene without tedious touch interactions.

2. Participants reported that it was relatively easy to achieve desired shots by manipulating the smartphone camera tangibly. Some participants used *AR continuous* to create a revealing shot such as having the drone start flying towards the building with the camera facing down, then slowly tilting up to reveal the building. This shot is hard to generate with

keyframe-based methods, but easy to create by continuously recording the camera movements.

3. One participant also reported that ARPilot can potentially promote teamwork and cooperation. Users can attempt as many different shots as they wish, and then discuss these shots with their team members in preview mode. Once these creative shots are selected, the drone can then execute the selected shooting assignments one-by-one.
4. Some experts suggested adding a speed control in either preview or capture mode. With this function, users can create cinematic tension in the aerial video. This function has already been implemented in [21], and we plan to add this in a future version of ARPilot.

Further Next Step Considerations

Our vision for ARPilot is to allow users to efficiently and easily generate drone footage while reducing the amount of retakes needed. However, there are currently several technical limitations. First, in *AR continuous*, although we applied a smoothing algorithm to stabilize shakiness, it is not effective if users created unnecessary movement. Second, our prototype currently requires users to rely on their own senses for height and distance in the AR world, in order to faithfully create smooth and desired footage. We aim to enhance the algorithm to prevent unnecessary shaking in the scene. Third, participants P1, P2, and P12 stated that there is a lack of editing function after the path planning, and desired a post-editing session after the program has simulated the path. We are investigating custom algorithms for ARPilot that can detect this type of assignment. Upon detection, it would suggest a smoother flight path for the user to decide. Moreover, users would be able to manually modify their desired flight path.

CONCLUSION

In this paper, we present ARPilot, a direct-manipulation approach for drone video shooting and flight planning. We designed three proposed AR interfaces that were implemented for use in actual drone flight videography: *AR keyframe*, *AR hybrid* and *AR continuous*. Furthermore, there has been a lack of studies investigating hybrid interaction (AR+Touch) for drone videography. Therefore, we designed a study to investigate the usage of AR interaction and traditional touch interaction in basic aerial shots. Combining both the positive features of spatial interaction and touch input, we demonstrated that *AR hybrid* performs more robustly as a technique in aerial videography. From our work, we also envision ARPilot's strong potential in expanding into other virtual camera planning tasks such as computer animations and interior walk through videos.

ACKNOWLEDGEMENTS

This study was partially supported by the National Science Council, Taiwan, under grant MOST105-2221-E-004-009-MY2 and MOST106-3114-E-002-012.

REFERENCES

1. ARKit. https://developer.apple.com/videos/play/wwdc2017/602/
2. Copilot. http://freeskies.co/copilot.html
3. DJI GS PRO. https://www.dji.com/ground-station-pro
4. DJI Mobile SDK. https://developer.dji.com/mobile-sdk/
5. Google Earth. https://www.google.com/earth/
6. Google Map Gestures. https://developers.google.com/maps/documentation/android-api/controls#map_gestures
7. Litchi for DJI Mavic. https://flylitchi.com
8. Mission Planner. http://ardupilot.org/planner/
9. Pix4D. https://pix4d.com/
10. Skywand. https://skywand.com/
11. Mattias Arvola and Anna Holm. 2014. Device-orientation is More Engaging Than Drag (at Least in Mobile Computing). In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational (NordiCHI '14)*. ACM, New York, NY, USA, 939–942.
12. Lonni Besançon, Paul Issartel, Mehdi Ammi, and Tobias Isenberg. 2017a. Hybrid Tactile/Tangible Interaction for 3D Data Exploration. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 881–890.
13. Lonni Besançon, Paul Issartel, Mehdi Ammi, and Tobias Isenberg. 2017b. Mouse, Tactile, and Tangible Input for 3D Manipulation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 4727–4740.
14. Wolfgang Büschel, Patrick Reipschläger, Ricardo Langner, and Raimund Dachselt. 2017. Investigating the Use of Spatial Interaction for 3D Data Visualization on Mobile Devices. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces (ISS '17)*. ACM, New York, NY, USA, 62–71.
15. Jessica R. Cauchard, Jane L. E, Kevin Y. Zhai, and James A. Landay. 2015. Drone & Me: An Exploration into Natural Human-drone Interaction. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)*. ACM, New York, NY, USA, 361–365.
16. Daniel Cohen-Or, Chen Greif, Tao Ju, Niloy J. Mitra, Ariel Shamir, Olga Sorkine-Hornung, and Hao (Richard) Zhang. 2015. *A Sampler of Useful Computational Tools for Applied Geometry, Computer Graphics, and Image Processing*. A.K. Peters, Ltd., Natick, MA, USA.
17. Francesca De Crescenzio, Giovanni Miranda, Franco Persiani, and Tiziano Bombardi. 2009. A First Implementation of an Advanced 3D Interface to Control and Supervise Uav (Uninhabited Aerial Vehicles) Missions. *Presence: Teleoperators & Virtual Environments* 18, 3 (jun 2009), 171–184.

18. Jane L. E, Ilene L. E, James A. Landay, and Jessica R. Cauchard. 2017. Drone & Wo: Cultural Influences on Human-Drone Interaction Techniques. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 6794–6799.

19. Christoph Gebhardt, Benjamin Hepp, Tobias Nägeli, Stefan Stevšić, and Otmar Hilliges. 2016. Airways: Optimization-Based Planning of Quadrotor Trajectories According to High-Level User Goals. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 2508–2519.

20. Wolfgang Hürst and Tair Bilyalov. 2010. Dynamic Versus Static Peephole Navigation of VR Panoramas on Handheld Devices. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia (MUM '10)*. ACM, New York, NY, USA, 25:1–25:8.

21. Niels Joubert, Mike Roberts, Anh Truong, Floraine Berthouzoz, and Pat Hanrahan. 2015. An Interactive Tool for Designing Quadrotor Camera Shots. *ACM Transactions on Graphics* 34, 6 (2015), 238:1–238:11.

22. Asier Marzo, Benoît Bossavit, and Martin Hachet. 2014. Combining Multi-touch Input and Device Movement for 3D Manipulations in Mobile Augmented Reality Environments. In *Proceedings of the 2Nd ACM Symposium on Spatial User Interaction (SUI '14)*. ACM, New York, NY, USA, 13–16.

23. Florian Mueller, Eberhard Graether, and Cagdas Toprak. 2013. Joggobot: Jogging with a Flying Robot. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13)*. ACM, New York, NY, USA, 2845–2846.

24. Florian 'Floyd' Mueller and Matthew Muirhead. 2015. Jogging with a Quadcopter. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2023–2032.

25. Tobias Nägeli, Javier Alonso-Mora, Alexander Domahidi, Daniela Rus, and Otmar Hilliges. 2017a. Real-Time Motion Planning for Aerial Videography With Dynamic Obstacle Avoidance and Viewpoint Optimization. *IEEE Robotics and Automation Letters* 2, 3 (feb 2017), 1696–1703.

26. Tobias Nägeli, Lukas Meier, Alexander Domahidi, Javier Alonso-Mora, and Otmar Hilliges. 2017b. Real-time Planning for Automated Multi-view Drone Cinematography. *ACM Trans. Graph.* 36, 4 (jul 2017), 132:1–132:10.

27. Marija Norusis. 2006. *SPSS 14.0 Guide to Data Analysis*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

28. Hiroki Nozaki. 2014. Flying Display: A Movable Display Pairing Projector and Screen in the Air. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14)*. ACM, New York, NY, USA, 909–914.

29. Mike Roberts and Pat Hanrahan. 2016. Generating Dynamically Feasible Trajectories for Quadrotor Cameras. *ACM Transactions on Graphics* 35, 4 (jul 2016), 61:1–61:11.

30. Jürgen Scheible, Achim Hoth, Julian Saal, and Haifeng Su. 2013. Displaydrone: A Flying Robot Based Interactive Display. In *Proceedings of the 2Nd ACM International Symposium on Pervasive Displays (PerDis '13)*. ACM, New York, NY, USA, 49–54.

31. Martin Spindler, Wolfgang Büschel, and Raimund Dachselt. 2012. Use Your Head: Tangible Windows for 3D Information Spaces in a Tabletop Environment. In *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces (ITS '12)*. ACM, New York, NY, USA, 245–254.

32. Daniel Szafir, Bilge Mutlu, and Terrence Fong. 2014. Communication of Intent in Assistive Free Flyers. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction (HRI '14)*. ACM, New York, NY, USA, 358–365.

33. Daniel Szafir, Bilge Mutlu, and Terry Fong. 2015. Communicating Directionality in Flying Robots. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction (HRI '15)*. ACM, New York, NY, USA, 19–26.