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

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) which 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 it intuitive to explore and frame the shots. We leveraged AR technology to explore three 6-DoF video-shooting interfaces on mobile devices: AR keyframe, AR continuous, and AR hybrid, and compared against traditional touch interface in a user study. The results show that AR hybrid is the most preferred by the participants, and costs the least effort among all techniques, while the user 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.5.2 User Interfaces: Input devices and strategies: User-centered design

Author Keywords

Interaction techniques; 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 dual-stick remote controller, which require a non-trivial level of expertise and dexterity to achieve decent quality scenes. To address these control challenges, drone manufacturers and researchers have designed alternative tools (e.g., [3, 6, 7, 2, 15, 16]) that leverage mobile devices—such as smartphones and tablets—to simplify drone controls. Some of them have allowed users to more easily navigate and shoot media in virtual 3D environments.

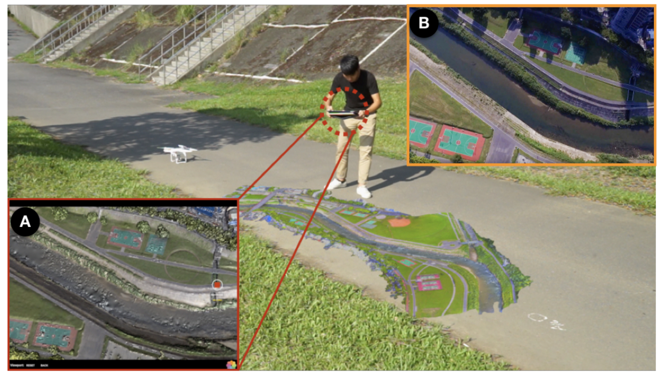

Figure 1. ARPilot is a direct-manipulation tool that facilitates route planning for drones.The user used mobile devices act as the viewfinder of the drone camera, and physically moved it act as the drone moving. (A) ARPilot emulated a camera-like interface. (B) Actual shooting image from (A).

Mobile AR [1] takes advantage of SLAM (Simultaneous Localization And Mapping) technology and has become more and more popular on mobile devices. We use this technology to transform a tablet as a tangible camera control. Users can take images and videos around virtual 3D models by physically moving and rotating their mobile devices. This type of interaction offers a more natural experience for users performing drone videography, since it facilitates the complexity of 6DOF manipulations while directly controlling video cameras. In this context, we propose ARPilot that leverages AR technology as a means to control drones for aerial videography. We designed and implemented three proposed types of AR video-shooting interfaces specifically for drone 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 these interfaces and touch interface for better understanding of how users use these techniques for drone videography. The results show that *AR hybrid* is most preferred among participants. And it costs less effort than the other techniques while *AR continuous* requires less time to finish most of the tasks. The user feedback also suggest that *AR continuous* supports users to create more creative shots. We will detail the findings later. To summarize, the contribution of this paper consist of (i) an intuitive flight planning tool that take advantage of mobile AR for drone videography, (ii) a user study that compare AR interfaces and touch input for taking aerial shots, (iii) insights and recommendation for the design of AR-based videography tools.

RELATED WORK

We developed ARPilot's direct-manipulation interfaces to improve mobile device-driven AR controls for drone videography. Therefore, we explored prior works that proposed solutions and insights for both drone flight planning and tangible virtual camera controls.

Flight Planning Tools

Conventional interfaces that are available to users for drone flight planning include tools such as Ground Station Pro [3], Litchi [6], and Mission Planner [7]. These tools operate by requiring users to plot the drone's waypoints on a 2D map of selected desired locations. Horus [19] further makes use of 2D mapping information and 3D model data of the target city to assist users in planning and previewing drone flight plans. To ensure the feasibility of the trajectory, Robert et al. [23] offers an approach to automatically generate an alternative route if the user proposes one that is not physically feasible. Airways [16] enables users to directly plan the drone's flight trajectory by drawing a 3D path and the tool will assist the user in optimizing the trajectory to ensure feasibility and smoothness.

Crescenzio, et al. [15] presents an interaction method to operate uninhabited aerial vehicle (UAV) by providing touchscreen operation and 3D rendering scene on display. The touch screen allows user to send high level command to the vehicle using a 2D command panel, and 3D virtual display offers and stereoscopic and augmented visualization of the operating vehicle, allowing better perception of the current states of the vehicle. Copilot [2] similarly provides 3D visualization planning on 3D model data from Google Earth, and utilizes touch gesture interaction for camera control and keyframe setting to enable users to focus on videography shots without regards to the drone's flight trajectory. Lastly, Skywand [9] introduced a VR interface that allowed users to control and plan aerial drone navigation around a virtual environment with two handheld controllers: one for navigation and one for drone point-of-view (POV). Our proposed ARPilot takes insights from these tools to provide more of a natural way for users to record aerial videos. We analogously treat the smartphone's screen as a camera's viewfinder, which allows users capture the 3D scene by simply walking within it.

Tangible and Touch Camera Controls

Researchers have investigated the benefits of navigational controls in 3D space 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

Marzo et al. [20] discovered that the most straightforward approach to manipulate virtual objects with smartphone devices was to map the translation and rotation applied the input device to the translation and rotation of the output object. Furthermore, applying orientation of the device to rotate the object allowed for more accuracy in the rotation when it required small rotations. When comparing tangible input alongside mouse and tactile controls, Besancon et al. [12] reported that users performed spatial navigation tasks with tangible input more quickly with similar levels of accuracy.

3D Data Exploration

Besancon et al.'s [11] and Buschel et al. [13] reported that users perceived spatial interaction as more supportive, comfortable, and preferred over touch interaction for navigating 3D data exploration.

Wall-Display Interaction w/ Physical movement

Jakobsen et al.'s results [18] suggested that moving may not be improving performance, depending on the use of virtual navigation. However, in Radle et al.'s study [22], the results indicated a 47% decrease in path lengths and a 34% decrease in task time in favor of physical navigation than multi touch navigation.

Peephole navigation on mobile devices

Hurst et al. [17] reported that users preferred and performed better with dynamic peephole navigation to view a VR panorama scene.. Spindler et al. [24] enables users to perform 3D interaction tasks in a more accessible manner for tabletop environments. It provided a novel concept for interacting with virtual 3D information spaces that combined tangible interaction, head tracking, and multi-touch techniques. Arvola et al. [10] reported that users found that panning—or spatial interaction along the horizontal plane—in peephole navigation was more engaging than touch interaction for mobile device panoramas.

Currently, there has been lacking in-depth research regarding use of AR and touch methods to conduct drone planning. Therefore, we will analyze how users interact with these interaction methods and provide insight for developing even more user-friend aerial videography planning tool.

ARPILOT

Our goal behind ARPilot is to allows users to intuitively and quickly perform drone videography via AR technology. Thus, we designed and implemented three AR interfaces: *AR keyframe*, *AR hybrid*, and *AR continuous*.

AR keyframe

AR keyframe is a pure AR interface that allows users to physically navigate the camera and take a sequence of critical photos at the desired positions and angles in 3D space. Each keyframe photo will be generated and suspended at the shooting point to inform users of previous locations. After users complete capturing keyframe photos, the system will connect all keyframe photos as a continuous camera path, which the drone would follow to fly. In this design, users can utilize their eye-hand coordination to manipulate the drone's camera as easily as they take regular photos in reality.

AR hybrid

Considering that touch interaction would alleviate the physical efforts for users while navigating a camera to a point [13], we added touch interaction into our design to create *AR hybrid*. With *AR hybrid*, users can not only spatially move and rotate camera but also manipulate the translation and orientation of 3D models by using touch gesture. Referring to Google map's gesture design[5], we provided three gestures as follows:

- One finger to drag 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 Z-axis.

Beyond gestures, the other designs of *AR hybrid* is same as *AR keyframe*.

AR continuous

Inspired by the idea that users can simultaneously and continuously interact with 6DoF via AR technology, we designed *AR continuous* to enable users to create an arbitrary and continuous camera path in 3D scene. In this interface, our system will automatically take keyframes every 0.5 seconds and the keyframes will not be suspended at the captured location to prevent overcrowding in the virtual environment. By this approach, users would act as a flying drone to shoot a video around 3D model.

Implementation

We developed ARPilot based on ARKit [1] with Unity engine. The app workflow can be referred to Figure 2.

Model Placement

In the beginning, our system requires users to import the 3D virtual model they want to take drone videography on. The 3D model can be built by Pix4D [8] or be loaded from Google earth. After that, the system would automatically perform the plane detection provided by ARKit and place the 3D model on a horizontal surface. To adjust the appropriate model size, we also provide the scale functions to users.

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 devices. Therefore, we further applied midpoint smoothing algorithm [14] to *AR continuous* for stabilizing the shakiness.

Safety Tips

To ensure safety for practical flight, ARPilot implemented two safety tips to alarm users of invalid controls. First, in planning step, the system will forbid users to capture the keyframe or record if they cross 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 occur by virtual camera and model during the path.

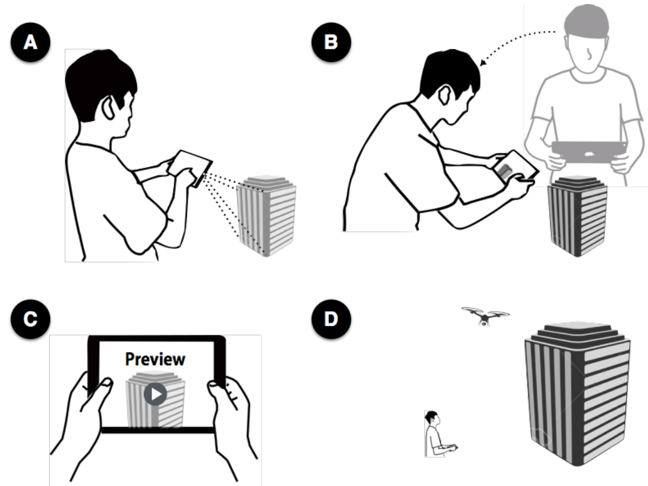

Figure 2. ARPilot consists of the following 4 steps: (A) Place model on the surface being sensed. (B) Manipulate drone's camera on our interfaces for planning videography. (C) Preview before starting drone mission. (D) Drone flights at the path being planned.

Simulation

In ARPilot, flight simulation was developed for users to preview the drone videography. To simplify the interaction procedure and rapidly presenting the simulation in front of users, ARPilot reduced unnecessary settings by users. The velocity of the simulated drone was set as constant 10m/s and the angular velocity would be depended on linear interpolation. When yaw rotation is necessary, the minimum clockwise or counterclockwise rotation would be calculated. All the simulations will be saved into a video gallery for latter use. Although the wind condition or GPS accuracy will change a bit between the previews and the real shots. The simulation can serve as a faithful preview of the real drone videography.

Drone mission

Current implementation of ARPilot supports *DJI phantom 3* as aerial vehicle, we developed it by using DJI Mobile SDK[4]. Once the user confirm to execute the preview shots, ARPilot will convert the contents of keyframes into waypoint mission compatible format, including:

- Drone's coordinate: longitude and latitude (WGS84 format) converted from 3D spatial coordinates. When a 3D model was imported, the original scale as well as the longitude/latitude of the model's origin point would be included. The coordinates in real world can be calculated based on these information.
- Drone's heading and camera's 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 true north of the model might not match that of mobile device camera. Therefore, the correct heading of drone was configured by the relative direction between virtual camera and 3d model. The pitch angle of drone's camera is determined by X-axis rotation of the virtual camera.

After conversions have been completed, the drone will adapts the settings which mentioned in simulation, use 10 m/s constant flying speed, and use minimum distance rotation when yaw rotation is necessary.

USER STUDY

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

Task Design

We consoled a experienced drone operator to analyze twenty award-winning videos from a drone videography competition [1] and find out the fundamental shots that are frequently used. The shots were categorized into five different tasks based on the 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 a 30° angle. This task requires two keyframes by changing position along one axis only.

Pullback

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

Orbiting

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

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

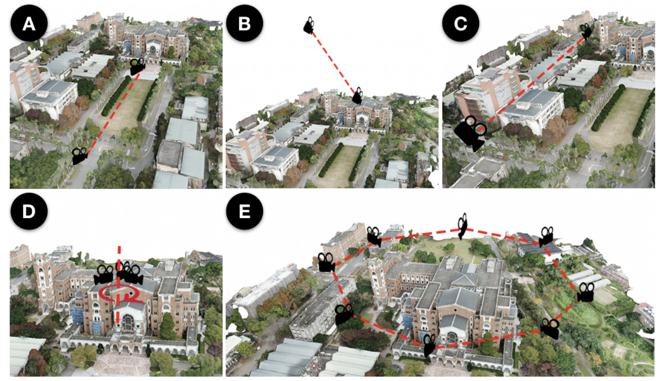

Figure 3. Our five categories for aerial drone videography: (A) forward, (B) pullback, (C) sideway sliding, (D) panorama, and (E) orbiting.

Participants

We recruited 12 participants (2 females) for our user study. Their ages ranged from 20 to 30 years (M = 23.16, SD = 1.26). We recruited all participants on a university campus. 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 existing familiarity working with AR environments.

Apparatus and Implementation

We provided participants with 10.5-inch iPad Pro devices in the study. The 3D model that was displayed to participants was a campus model as shown in Figure 3, which was built using Pix4D [8]. 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 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 the participant to rank by preference and provide feedback of the four techniques.

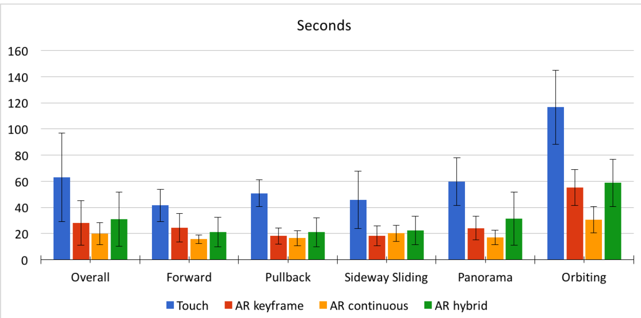

Figure 4. Completion time for each task. AR continuous showed the least amount of time required to complete most of the tasks.

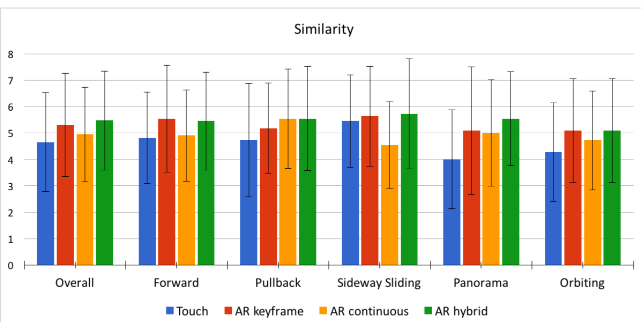

Figure 5. Similarity of the four interfaces in each task. Overall, *AR hybrid* received the highest score from users' subjective rating.

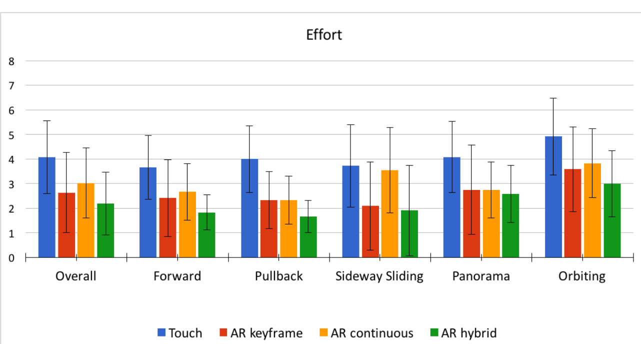

Figure 6. Effort rating of the four interfaces for each task. The score was proportional to effort (e.g., lower score represented lower effort).

RESULT

Performance

We analyzed the completion time of 4 conditions in each task. As shown in Figure 4, the overall time of *AR keyframe (28.9s), AR hybrid (30.8s) and AR continuous (19.5s)* was over 50% faster compared to the *Touch* (64.7s). With Friedman test and Post-hoc Analysis[21], the improvement in time was statistically significant ($p < .05$) in all categories among 3 AR interfaces and touch interface.

Similarity

With The Friedman test, there was not statistically significant among interfaces in each task. The average of users' rating on a 7-point Likert scale for the path similarity were *AR keyframe (5.36), AR hybrid (5.48), AR continuous (4.95), Touch (4.67)*. The result is shown in Figure 5. User ratings for *AR keyframe* and *AR hybrid* both showed a better result than touch interface. Most participants (8) expressed AR interfaces are easier to fine-tuned than touch interface. Likewise, we found that when the paths created by AR interface were easier to stay on the same height, e.g., as shown in Figure 7A, B, and C. On the contrary, the touch interface(Figure 7D) shows a more scattered keyframes in 3D space, so the paths were not as precised.

Effort

The result in Figure 6 showed that touch interface needed the most effort among all interfaces ($p < .05$). The participants pointed out thatit took more trials to capture the right shot in touch interface. On the contrary, *AR keyframe* and *AR continuous* allowed user to physically adjust the desired location and angle at the same time to catch the shot. However, if the desired locations are too far from each other, it might be exhausting due to the large movement. *AR hybrid* got the lowest effort among all tasks because it allowed user to speed up wide range movement with gesture input while adjusting slight frame difference with AR interaction.

Preference

We calculated the final ranking for all conditions from the final questionnaires. The order are shown as following: 1) *AR hybrid*, 2) *AR keyframe*, 3) *AR continuous* and 4) Touch. As result, we found that ranking order is relatively similar to overall scores in Figure 8. Although *AR continuous* shows higher intuition than *AR keyframe*, however, due to the increase in effort, *AR continuous* is less preferable than *AR keyframe* for most participants (9). The result of intuitive, simplicity and satisfaction between 4 interfaces are statistically significant ($p < .05$). 11 participants reported to prefer the *AR hybrid* over the other interfaces because it combines positive feature from both AR element and gesture control. However, there is one participant preferring *AR keyframe* most due to the satisfaction with the AR interaction. *"AR keyframe satisfy most of my needs so I don't feel gesture is necessary."*(P7). Most users dislike the touch interface due to being unable to perform minor detail adjustments. However, *AR hybrid* allow users to move the tablet to adjust the minor change. For example, feedback from user 5, 7, and 9 stated: *AR hybrid maintained the benefit of the AR feature in minor adjustment* *while reducing the physical effort because I do not have to squat down to take closer shots of the model. Instead, I can use gesture to zoom into the model.*

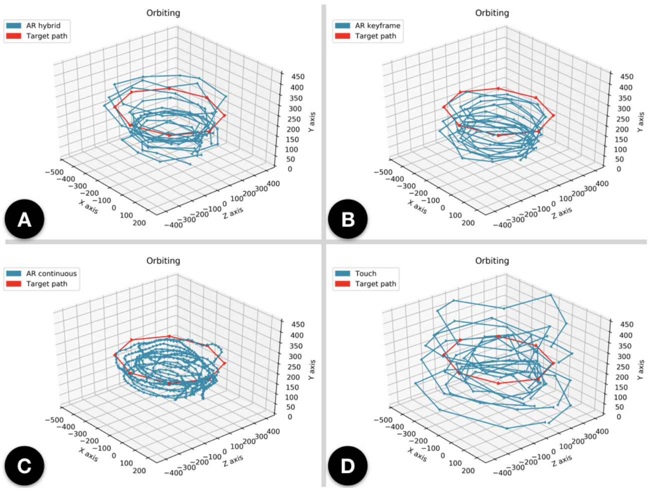

Figure 7. The flight path recorded in the user study(A) AR hybrid. (B) AR keyframe. (C) AR continuous. (D) Touch.

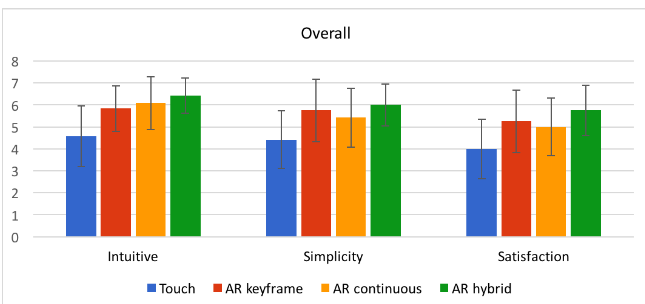

Figure 8. Overall score of each interface. AR hybrid is highly favorite.

DISCUSSION

Effort in Physical Movement

While providing a intuitive way to perform drone videography, AR interface required user to squat down in order to record a closer shot. This might increase the effort and be tiring, especially *AR continuous*. In sideway sliding, participants reported to be more tired due to the necessity of getting closer to the model to get a closer shot. *"It is more tiring to take a shot at the lower elevation."*(P4). On the contrary, the *AR hybrid* allowed users to use the gesture input to adjust the height placement of the model instead and reduced the effort as a result. Therefore, we believe that if users are able to adjust the height placement of the model, it would decrease the effort required during recording.

Creative task

After the study, we invite six aerial photographers to try out ARPilot freely to take any shot they like. In the freeform shot task we observed several behaviors. In order to find a special shot, some participants frequently walked around the miniature scene and tried different view angles while using the AR interfaces. Compared to touch interface, AR interfaces not only enable users efficiently create shots with body movements but also encourage users to explore the miniature scene without tedious touch interactions. Participants reported that it's relative easy to find the expected shot by manipulating the smartphone camera tangibly. Some participants used *AR continuous* to create a reveal shot such as having the drone start flying towards the building with the camera facing down, then slowly tilt up to reveal the building. This shot is hard to be generated by keyframe-based methods but easy create by continuously recording the camera movements. One participant also reported that ARPilot can promote teamwork and cooperation. Users can try different shots as many as possible and then discuss with team members in preview mode. Once the creative shots are selected, the drone then executes the selected missions one by one. Some experts suggested to add speed control either in preview or in capturing mode. With this function, users can create cinematic tension in the aerial video. This function has already been implemented in [19]. We plan to add it in ARPilot in the future.

LIMITATIONS AND FUTURE WORK

Our vision for ARPilot aims to allow users 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 the smoothing algorithm to stabilize the 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 the unnecessary shaking in the scene. P1, P2, and P12 stated that there are lacking of editing function after the path planning and wishing for a post-editing session after the program has simulated the path. ARPilot is looking into developing custom algorithms that can detect the type of mission. Upon detection, it will suggest a smoother flight path for the user to decide. In addition, users will be able to manually modify their desired flight path.

CONCLUSION

In this paper, we present ARPilot – a novel approach for drone video and flight planning. We designed three AR interfaces: *AR keyframe*, *AR hybrid* and *AR continuous* and implemented for drone's actual flight. Second, as of yet, there is a lack of studies using hybrid interaction (AR+Touch) for drone videography. Therefore, we designed a study to investigate the usage of AR interfaces and traditional touch interface in the basic aerial shots: forward, pullback, sideway sliding, panorama, and orbiting. In previous works, spatial interaction was perceived as more supportive, comfortable and overall preferable to touch input, however, traditional touch still has its advantages. Combining from both positive features, *AR hybrid* carries out a more robust technique in aerial videography. We will improve this technique by applying the findings from our study in the future. We also envision ARPilot to be extended to other virtual camera planning tasks such as computer animations or interior walk through videos.

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 Map gestures. https://developers.google.com/maps/documentation/android-api/controls#map_gestures
6. Litchi for DJI Mavic. https://flylitchi.com
7. Mission Planner. http://ardupilot.org/planner/
8. Pix4D. https://pix4d.com/
9. Skywand. https://skywand.com/
10. 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.
11. 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.
12. 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.
13. 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.
14. 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.
15. 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.
16. 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.
17. 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.
18. Mikkel R. Jakobsen and Kasper Hornbæk. 2015. Is Moving Improving?: Some Effects of Locomotion in Wall-Display Interaction. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. 4169–4178.
19. 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.
20. 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)*. 13–16.
21. Marija Norusis. 2006. *SPSS 14.0 Guide to Data Analysis*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
22. Roman Rädle, Hans-Christian Jetter, Simon Butscher, and Harald Reiterer. 2013. The Effect of Egocentric Body Movements on Users' Navigation Performance and Spatial Memory in Zoomable User Interfaces. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces (ITS '13)*. 23–32.
23. 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.
24. 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.