MR-Based UAV Route Planning for the Coverage Task

Jia-Hao Wei[1(✉)], Jen-Jee Chen[1], and Yu-Chee Tseng[1,2,3]

[1] College of AI, National Yang Ming Chiao Tung University, Taipei, Taiwan
i860510.cai08g@nctu.edu.tw
[2] Academia Sinica, Taipei, Taiwan
[3] Kaohsiung Medical University, Kaohsiung, Taiwan

Abstract. In this work, we define a new problem called coverage task, which is to plan a flying route for a UAV to inspect a structure surface of interest with a certain visual quality ensurance. Users can use the designed MR interface for a coverage task via intuitive head gaze and hand gesture instructions. In particular, we present a spray light interface for easily visualizing the inspection area of a flying route. We also conduct a user study evaluation to verify our framework on HoloLens. The study shows that our spray light interface performs consistently well in subjective usability and improves over other approaches in performance. The results validate that the semi-immersive interface provides a viable alternative to conventional interfaces for UAV route planning.

Keywords: Mixed reality · HoloLens · Unmanned aerial vehicle · Route planning

1 Introduction

Drones are typically used in military surveillance or reconnaissance [3,15]. Recently, drones are more frequently used in industrial applications, such as quality and condition assessment [11,21], post-disaster exploration [1], and agriculture [9,10].

Route planning of drones is an important topic. There are various route planning interfaces for drones nowadays. The conventional interface uses a joystick combined with a 2D interface [19,20]. Recently, more advanced virtual reality and augmented reality interfaces have been proposed [6,14,22].

In this work, we define a new problem called coverage task, which is to plan a flying route for a UAV to inspect a structure surface of interest with a certain visual quality ensurance. To facilitate this task, we design a set of Mixed Reality (MR) interfaces. We consider MR rather than VR because VR will totally block a pilot's vision on the physical environment. Since our main objective is to understand the spatial information of how well the surface under inspection is covered by the current route, existing interfaces that simply focus on

S. Paiva et al. (Eds.): SmartCity360° 2021, LNICST 442, pp. 293–306, 2022.
https://doi.org/10.1007/978-3-031-06371-8_19

virtual route planning does not meet our need. Our goal is to develop interfaces that combine waypoints and drone camera's field of view to help measure the coverage of a specific range. In the meantime, our interfaces have to perform obstacle avoidance and ensure visual quality. To the best of our knowledge, this is the first work addressing these issues for drone route planning.

Our MR interfaces are developed based on Microsoft HoloLens 1. We use the spatial mapping feature from the HoloLens to construct a 3D room model in an indoor environment. Users are asked to use our interfaces to complete a surface inspection task. The entire system provides a semi-immersive and intuitive interaction for route planning mission with a certain visual quality requirement. A spray light interface is provided to help users verify the image quality under the currently planned route. Users can adjust waypoints by hand gestures with head gazes, which is traditionally hard to accomplish on a 2D screen. The overall performance of our interfaces is evaluated by user study analysis. The analysis results and participants' feedbacks consistently indicate that our interfaces have significant improvements in performance and wide applicability in the coverage task.

The rest of this paper is organized as follows. Section 2 reviews some related works. We formally define coverage task and present the whole architecture of our interface in Sect. 3. In Sect. 4, we describe our user study design and procedure. Our user evaluation results are presented in Sect. 5. We make a conclusion and discuss the future work in Sect. 6.

2 Related Work

Traditional human-drone interaction approaches mainly use a joystick controller with a flat-screen to steer drones. Using touch screen devices to manipulate drones are studied in [8,12]. An innovative interface is developed in [8] on an iPad to intuitively steer drones by simple touch gestures and to take videos in an egocentric view. For visualizing 3D spatial data, a toolkit called Rviz is proposed [7]. The work [4] compares the performance of physical or virtual object interaction in Unity, GAZEBO, and Rviz platforms. However, these works are not for interacting with spatial objects.

For spatial interaction tasks, modeling real objects is a key to conveying human intention. The work [18] uses VR interfaces combined with OptiTrack to transform real scenes and objects to VR world. Another direction is to combine VR interfaces with Kinect [6,22], as Kinect can provide more stereoscopic information and gestures for natural interactions.

While VR interfaces only support human-drone interactions in a remote mode, augmented reality (AR) is an option to provide real world scenes of the flying environment and target settings. AR interfaces for displaying virtual drones and targets and piloting by a joystick are proposed in [14,23]. In [13], a 3D scale-down spatial environment is reconstructed by stereo cameras and users can manipulate a drone by using head gazes.

While AR interfaces achieve good performance when a drone is in direct visual range, mixed reality (MR) enables interaction when a drone is in non-light

of sight [2]. Through spatial mapping, the work focuses on drone manipulation in egocentric view or exocentric view and makes a comparison on head gazes and joystick for drone steering. The MR interface [5] allows interaction with drones by 3D rendering and voice control. This work renders a grid map of the flight environment with ground tags for drone localization, so drone steering by voice commands in a grid-by-grid manner is possible. Since the grid map is only 4×4 and needs to align with ground tag manually, the flight paths are limited in specific scenarios.

All the above works have achieved immersive rendering. But pure VR lacks of real understanding on physical world, while AR directly overlays graphics on the physical world. Therefore, we choose to build our interface on MR so that users can simultaneously visualize a physical environment and interact with virtual objects.

3 Propose Approach

3.1 Coverage Task

The *coverage task* for UAV is formally defined as follows. A drone needs to be dispatched to an environment to inspect a surface of interest S of a structure with a certain visual quality ensurance. We define quality ensurance by the distance between the drone's camera and the region under inspection. The inspection routing path of the drone is defined as a sequence of waypoints $R = \{ r_1, r_2, \ldots r_n\}$. At any point in R, the camera's field of view (FOV) on S that is within a distance threshold of δ is considered as covered, and the other FOV is considered as uncovered. Two examples are illustrated in Fig. 1. In Fig. 1 (a), since the drone is too far away from the surface, no FOV is considered as covered. In Fig. 1 (b), the central region $[A, B]$ that is within the distance threshold δ is covered, while the other regions $[C, A]$ and $[B, D]$ are uncovered.

Given a routing path R, it is not hard to mark the regions of S that are covered. Depending on the task, one may specify a required coverage ratio p of S that is covered by R. We assume that the environment has been pre-scanned and thus its 3D model is already known. Our goal is to design a MR tool (such as Hololens) and a set of convenient user interfaces to plan a route R to meet the required coverage ratio p. In order to define visual quality ensurance, let us assume that we want to recognize an object (such as a crack) of length L cm by displaying at least M pixels in an image. In order to guarantee this visual quality, we need to compute the minimum distance threshold δ between the drone's camera and the region under inspection. That is, we need to transform to a pixel unit to its actual distance. We apply the *collinearity equations* for the transformation. To relate camera coordinates in a sensor plane (2D) to ground coordinates (3D), let us consider Fig. 2. Let P = (x, y, y) be a point in the ground coordinate, which is projected to point P' = (x', y', z') on the sensor plane through a projection optical centre C = (x_0, y_0, z_0). To relate P and P', since P', projection optical centre C, and P is collinear, we can use proportional

function (Eq. 1) to get P'. Let the distance from C to the sensor plane be d, and we can get the following equations (Eq. 2 and Eq. 3):

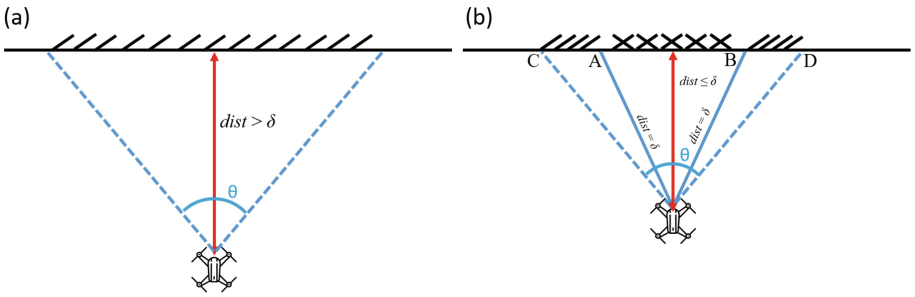

Fig. 1. Examples of covered and uncovered regions (θ = camera view angle).

$$\frac{x' - x_0}{x_0 - x} = \frac{y' - y_0}{y_0 - y} = \frac{-d}{z_0 - z}. \tag{1}$$

$$x' - x_0 = -d\frac{x - x_0}{z - z_0}. \tag{2}$$

$$y' - y_0 = -d\frac{y - y_0}{z - z_0}. \tag{3}$$

Assuming the distance of P and Q is L cm and P, Q are on the same plane, and we can get the pixel distance of the two points by the above equation. With pixel unit, we can derive the minimum distance threshold δ.

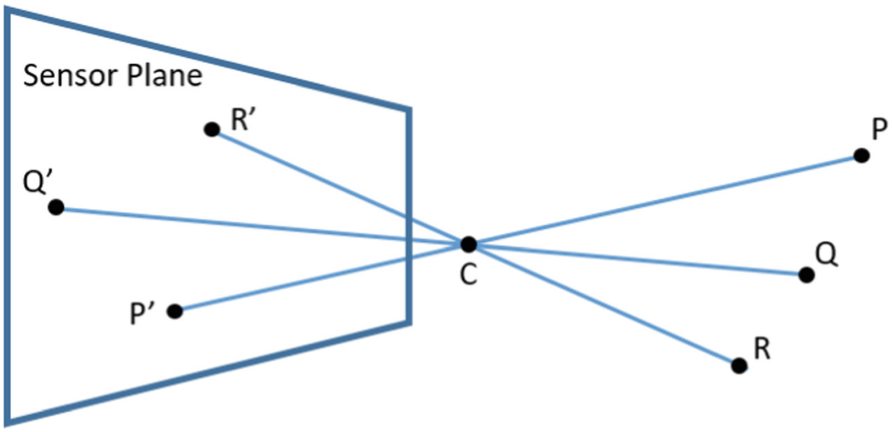

Fig. 2. Point P is (x, y, z), and the relative point P' on the sensor plane is (x', y', z'), projection optical centre is (x_0, y_0, z_0).

3.2 System Architecture

Next, we propose our MR-based model to handle the coverage task. The overall architecture is shown in Fig. 3. We use Microsoft HoloLens 1 in this work. The interface is developed on Unity 2018.4 and Visual Studio Code platform. The task is given as (S, p) and it is for an indoor room environment, of which the 3D spatial model has been pre-scanned by HoloLens.

On the user interface part, there are three components. The spatial information is processed by HoloLens functions to construct the 3D model. HoloLens already provides some head gaze gestures for clicking on the MR world by air tap [17]. We also provide button clicking by head gaze air tap for waypoints selection. The routing path between two adjacent waypoints is connected automatically. There are also interfaces for editing waypoints. As the coverage ratio of the current route is important for a user to understand how well the coverage task is conducted, besides the current routing path, we also provide convenient visualization by rendering the current coverage regions on S based on the distance threshold δ.

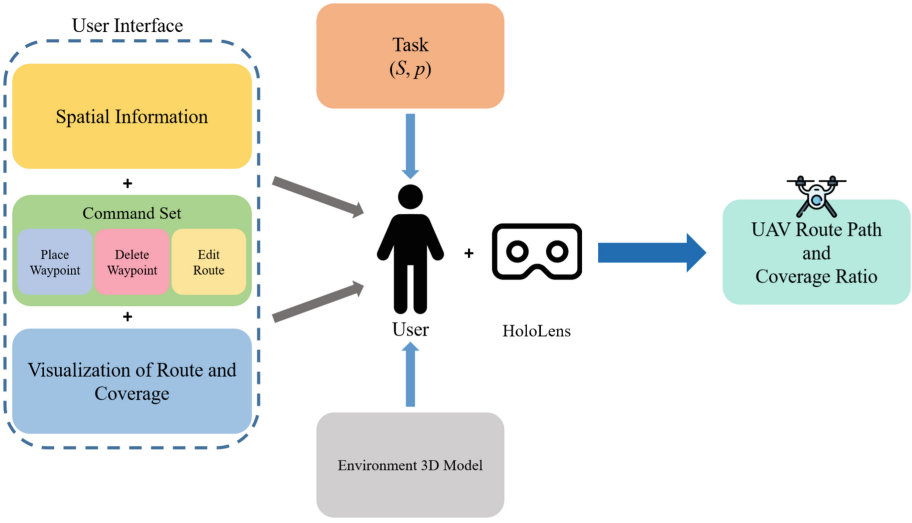

Fig. 3. System architecture.

3.3 Interface Design

Spatial Mapping. First, we scan the space to get its spatial information and construct a 3D space model. This spatial mapping can be done by the visual SLAM algorithm and onboard Lidar of HoloLens [16]. The scanning range is about twenty square meters. It can recognize surface planes, such as table, ceiling, floor, and wall. Detecting and classifying the complicated objects in a messy

indoor environment is a challenge. An indoor space is generally surrounded by rectangular ceilings, floors, and walls. Figure 4 (a) shows a real scene in our experiment. After scanning by HoloLens, a triangular mesh model can be constructed by Unity, as illustrated in Fig. 4 (b). Then, ceilings and floors are classified based on their mesh sizes and locations. Further, vertical meshes greater than a certain size (such as three square meters) are classified as wall surfaces. The other meshes are classified as obstacles in the room, which are shown as occlusion objects by HoloLens and should be avoided in route planning. After these processing, we obtain a cleaner room model as shown in Fig. 4 (c).

After the 3D space model is constructed, we designate a region S of wall areas as our inspection targets. That is, users need to plan a route for a drone to inspect the region S. We use translucent gray materials overlaid on the top of large surfaces for visualization. Target surface S is bounded by pink lines. Then users can conduct route planning in the space. We also use spray light of bright color to cover the areas that have been covered by the current route. The uncovered areas, however, remain in translucent gray. So it is hard for users to not observe the uncovered areas. Figure 5 shows our interface.

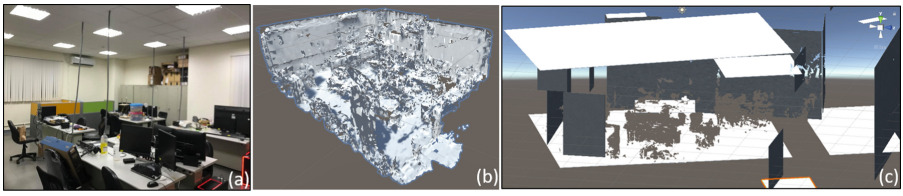

Fig. 4. Spatial mapping steps: (a) a real room scene, (b) 3D triangular mesh model, and (c) surface model.

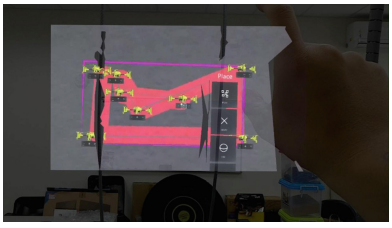

Fig. 5. Wall, target inspection surfaces S, and spray light of covered areas.

Waypoint Editing Interfaces. Next, we introduce our interfaces for waypoint editing. We design three buttons, *Place*, *Delete*, and *Edit*, in our mixed reality UI, as shown in Fig. 6. Controlling the cursor, which is on the center of field of vision, is done by head moving. Clicking on any button results in corresponding actions on waypoints. The straight line between two adjacent waypoints is drawn by a white color line. Whenever a new waypoint is added, we render on S the newly covered area by adding more spray light in pink. Also, the current coverage ratio will be shown immediately on screen. The *Place* button is to add a new waypoint at the end of the current route. The new path is shown by white. If there is any physical obstacle exists in the path, the part of the path will be changed to red to remind users. The *Delete* button is to remove a misplaced waypoint. If the deleted waypoint is between two other waypoints, the two waypoints will be connected by a new white line. The *Edit* button is to adjust the position of an existing waypoint. A user can pick the drone and release it at a new position. The new flight route and spray light will be updated automatically. Since this action may change the covered areas, one can visualize the effect by the size of spot light immediately.

Fig. 6. Editing interface: place button, delete button, and edit button.

Visualization of Route and Coverage. In order to visualize the drone route in different circumstances, we use several presentation method. The original route is shown in Fig. 7 (a), where waypoints are connected by white lines. We observe that some users prefer to fine-tune waypoints at the final stage of a task. The original route is obstacle-avoiding before their adjustment. In this case, they should be notified by the color change caused by route change. When

obstacles exist in an adjusted part, the part of route will be marked in red to remind users to further revise the part. In Fig. 7 (b), we show a case where an obstacle (whiteboard) exists between two waypoints and the color of the part turns to red. Besides, the size of spray light may change when waypoints are adjusted. This feature is very helpful for users to fine-tune a route to fulfill the target coverage ratio p. In Fig. 7 (c), the size of spray light decreases and then increases because of the middle waypoint changes its position. When a waypoint is too far from S, the corresponding spray light may disappear to warn the users.

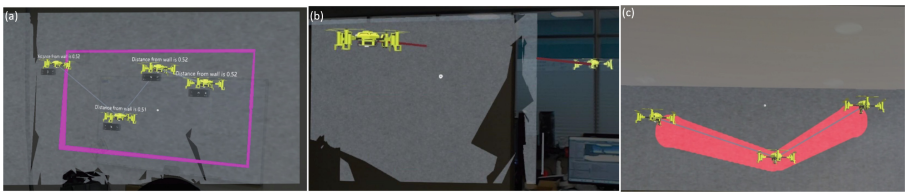

Fig. 7. (a) A typical obstacle-free route, (b) a route marked by red when crossing an obstacle, (c) change of spray light when one moves the middle waypoint.

4 User Study

We have conducted experiments to evaluate the efficiency and usability of our interfaces by observing human users to complete a series of coverage tasks. These tasks are meant to represent inspection applications requiring a drone to navigate specific areas and following obstacle avoidance and quality ensurance requirements.

4.1 Participants

We recruited ten participants (one female) from a local university between ages twenty and thirty, and they all had normal or corrected-to-normal vision. Two participants had previous experiences with AR or VR HMDs.

4.2 Study Design

We used a within-subjects design and the distance between the designated surface S and participants is at least one meter. The participants were given tasks of route planning and these tasks are divided into spray light enabled and disabled states (see Fig. 8 for a comparison).

The inspection tasks may involve *large*, *medium*, or *small cracks*, which may lead to different threshold values for δ (δ = 30–50, 50–70, 70–90). To remove the effect of learning effect, the order of δ values is random for every participant. In

every state, two shapes of S but of the same size are evaluated (Fig. 8). That is, each participant performed twelve trials ($two\ states \times two\ shapes \times$ three δ). A participant proceeds to the next trial only when he/she thinks that the current coverage task is finished and we use OpenCV to get the coverage ratio. The average total trial time is about 1 h (Fig. 9).

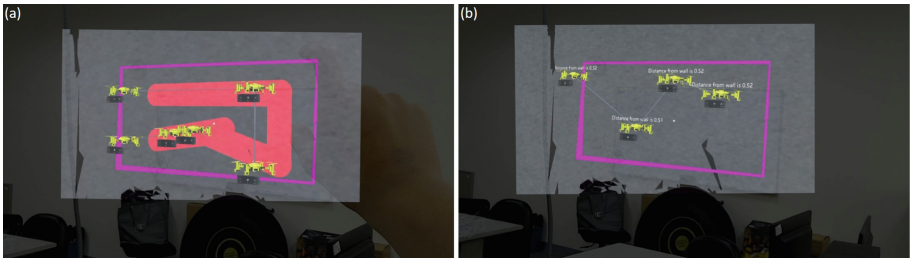

Fig. 8. (a) Spray light enabled state and (b) disabled state.

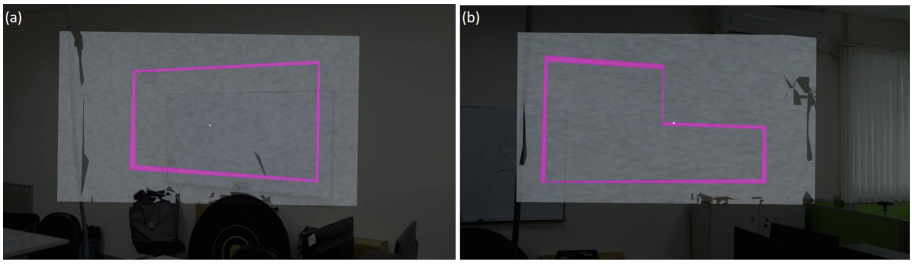

Fig. 9. Two shapes of target surface S.

4.3 Task Execution Procedure

After we explained how to use our interfaces, participants were allowed to practice as much as they felt comfortable. The participants performed two shapes of tasks for each given spray light state and were told to perform each task as fast as possible and try to fill S as much as possible. If needed, they were allowed to take a break between trials. We measured the time, waypoint number, route distances and coverage ratio. Participants then filled out a usability questionnaire (Fig. 10) and we interviewed them after all tasks were completed.

	Question	Score
Q1	I quickly adapted to this interface	1-10
Q2	Controlling the system with this interface came naturally to me.	1-10
Q3	This interface was pleasant to use.	1-10
Q4	This interface was confusing.	1-10
Q5	Without this interface, I cannot finish those task.	1-10

Fig. 10. Usability questionnaire.

4.4 Evaluation Results

Among the ten recruited participants, eight were able to complete the whole procedure. Thus, the analyses are based on these eight participants. To quantify the results, we calculate participants' performance by:

$$Performance = \frac{Coverage\ Rate(\%)}{Complete\ Time + No.Waypoints \times P}. \tag{4}$$

The denominator is considered as penalty. First, the more waypoints there are, the more turning overhead it will incur. Second, we observe that some participants prefer to use more waypoints to cover an area. Therefore, we set $P = 0.5$ s as an time overhead. The evaluations of pairwise t-test are shown in Fig. 11 (a). As can be seen, there are significant advantage in using spray light to guide a user. In the small δ case, the enabled state has mean = 73.72 and standard deviation = 15.17, versus the disabled state's mean = 55 and standard deviation = 11.45. In medium and large δ cases, we see the similar trend. In all cases, the p-value is smaller than 0.001. The analysis shows that participants took significantly less time to complete a task when spray light is enabled. We also plot the completion time of pairwise t-test in Fig. 11 (b). In the small δ case, the enabled state has mean = 131 and standard deviation = 28.24, versus the disabled state's mean = 150.67 and standard deviation = 31.67. In medium and large δ cases, we see the similar trend. In all cases, the p-value is smaller than 0.001. The results validate that participants took longer time in disabled state than enabled state in every task.

In Fig. 12, we compare the routing path length as it reflects drones' battery consumption. The enabled state has mean = 38 and standard deviation = 11.84, versus the disabled state's is mean = 62.42 and standard deviation = 18.81. The p-value is also smaller than 0.001. In summary, the analysis reveals that the enabled state is more efficient than to disabled state in all aspects.

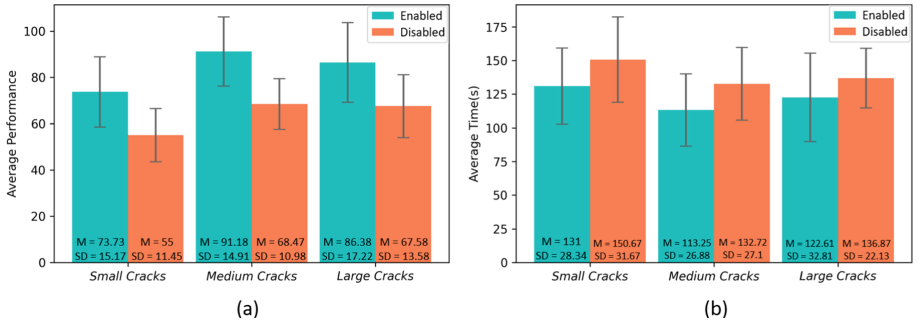

Fig. 11. (a) Performance measurements and (b) Time measurements.

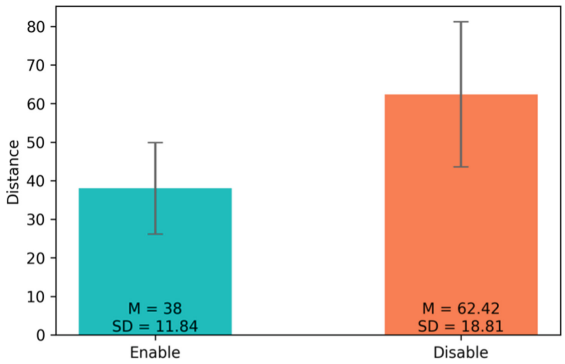

Fig. 12. Distance measurements.

4.5 Interview Comments

In interviews, we receive the similar comments that when the spray light function is enabled, it is much easier to adjust waypoints and visualize the area that are not yet covered. Some participants mentioned that they finished their tasks in the enabled state with less fatigue, as quoted below:

"Comparing to disabled state, I think the interface (enabled state) greatly facilitate manipulation and adjustment." (P2)

"The tasks can be finished in enabled state easily without hesitation and frazzle." (P5)

Additionally, all users can well understand the concept of spray light easily. They did recognize how the coverage areas are reflected by spray light, are quoted below:

"It is hard for me to finish the tasks without spray lighting path." (P1)

"Using enabled one is easy and convenient." (P7)

From the questionnaire results, we conduct post-experiment comparison on usability and flexibility. Note that the fourth question in the questionnaire reverses the valence to test if users are paying attention. In the analysis, this reverse question's scores are inverted to match the other scales. To interpret the survey question as a measuring parameter, we use one-way repeated-measures ANOVA to analyze the results. The spray light enabled state improves an average of 5.35 Likert points over the disabled state (with p-value $= 0.0329 < 0.05$). While this brief survey certainly does not cover all aspects of usability, it does demonstrate a preference for spray light. In addition, we did not observe a significant difference across all metrics when comparing those users with VR experience to those without VR experience. We speculate the reason is that users can still interact with real objects to get spatial information in semi-immersive environment.

5 Conclusions and Future Work

In this paper, we have defined a new coverage task for drone route planning and presented a spray light interface to facilitate users to arrange waypoints while visualizing the covered and yet-to-be-covered areas in a MR environment. A set of design considerations and insights that are unique to the routing planning interaction problem are presented. Our user evaluation results validate the effectiveness and efficiency of our design. In conclusion, the spray light design provides a high degree of freedom and applicability. The interface can be integrated with various interaction techniques and devices in other applications and circumstances. As to future work, we have validated our design for indoor environments, but outdoor environments of larger scales deserve further study. Manipulating multiple drones also deserves further research.

Acknowledgements. This research is co-sponsored by ITRI, Pervasive Artificial Intelligence Research (PAIR) Labs, and Ministry of Science and Technology (MoST). This work is also financially supported by "Center for Open Intelligent Connectivity" of "Higher Education Sprout Project" of NYCU and MOE, Taiwan.

References

1. Aljehani, M., Inoue, M.: Performance evaluation of multi-UAV system in post-disaster application: validated by HITL simulator. IEEE Access **7**, 64386–64400 (2019)
2. Erat, O., Isop, W.A., Kalkofen, D., Schmalstieg, D.: Drone-augmented human vision: exocentric control for drones exploring hidden areas. IEEE Trans. Visual Comput. Graphics **24**(4), 1437–1446 (2018)

3. Giese, S., Carr, D., Chahl, J.: Implications for unmanned systems research of military UAV mishap statistics. In: 2013 IEEE Intelligent Vehicles Symposium (IV), pp. 1191–1196. IEEE (2013)
4. Hoenig, W., Milanes, C., Scaria, L., Phan, T., Bolas, M., Ayanian, N.: Mixed reality for robotics. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5382–5387. IEEE (2015)
5. Huang, B., Bayazit, D., Ullman, D., Gopalan, N., Tellex, S.: Flight, camera, action! Using natural language and mixed reality to control a drone. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 6949–6956. IEEE (2019)
6. Jie, L., Jian, C., Lei, W.: Design of multi-mode UAV human-computer interaction system. In: 2017 IEEE International Conference on Unmanned Systems (ICUS), pp. 353–357. IEEE (2017)
7. Kam, H.R., Lee, S.H., Park, T., Kim, C.H.: RViz: a toolkit for real domain data visualization. Telecommun. Syst. **60**(2), 337–345 (2015)
8. Kang, H., Li, H., Zhang, J., Lu, X., Benes, B.: FlyCam: multitouch gesture controlled drone gimbal photography. IEEE Robot. Autom. Lett. **3**(4), 3717–3724 (2018)
9. Kellenberger, B., Marcos, D., Lobry, S., Tuia, D.: Half a percent of labels is enough: efficient animal detection in UAV imagery using deep CNNs and active learning. IEEE Trans. Geosci. Remote Sens. **57**(12), 9524–9533 (2019)
10. Kitano, B.T., Mendes, C.C., Geus, A.R., Oliveira, H.C., Souza, J.R.: Corn plant counting using deep learning and UAV images. IEEE Geosci. Remote Sens. Lett. (2019)
11. Kuang, Q., Jin, X., Zhao, Q., Zhou, B.: Deep multimodality learning for UAV video aesthetic quality assessment. IEEE Trans. Multimedia **22**(10), 2623–2634 (2019)
12. Lan, Z., Shridhar, M., Hsu, D., Zhao, S.: XPose: reinventing user interaction with flying cameras. In: Robotics: Science and Systems, pp. 1–9 (2017)
13. Liu, C., Shen, S.: An augmented reality interaction interface for autonomous drone. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 11419–11424. IEEE (2020)
14. Mashood, A., Mohammed, M., Abdulwahab, M., Abdulwahab, S., Noura, H.: A hardware setup for formation flight of UAVs using motion tracking system. In: 2015 10th International Symposium on Mechatronics and its Applications (ISMA), pp. 1–6. IEEE (2015)
15. Ma'Sum, M.A., et al.: Simulation of intelligent unmanned aerial vehicle (UAV) for military surveillance. In: 2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS), pp. 161–166. IEEE (2013)
16. Microsoft: HoloLens 1st hardware-Microsoft docs (2018). https://docs.microsoft.com/zh-tw/hololens/hololens1-hardware
17. Microsoft: Microsoft 2018a. Gestures-mixed reality (2018). https://docs.microsoft.com/en-us/windows/mixed-reality/gestures
18. Paterson, J.R., et al.: Improving usability, efficiency, and safety of UAV path planning through a virtual reality interface. In: Symposium on Spatial User Interaction, pp. 1–2 (2019)
19. Rothwell, C.D., Patzek, M.J.: An interface for verification and validation of unmanned systems mission planning: communicating mission objectives and constraints. IEEE Trans. Hum.-Mach. Syst. **49**(6), 642–651 (2019)
20. Wilde, G.A., Murphy, R.R.: User interface for unmanned surface vehicles used to rescue drowning victims. In: 2018 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), pp. 1–8. IEEE (2018)

21. Wu, W., Qurishee, M.A., Owino, J., Fomunung, I., Onyango, M., Atolagbe, B.: Coupling deep learning and UAV for infrastructure condition assessment automation. In: 2018 IEEE International Smart Cities Conference (ISC2), pp. 1–7. IEEE (2018)
22. Wu, Y., Song, J., Sun, J., Zhu, F., Chen, H.: Aerial grasping based on VR perception and haptic control. In: 2018 IEEE International Conference on Real-time Computing and Robotics (RCAR), pp. 556–562. IEEE (2018)
23. Yu, Y., Wang, X., Zhong, Z., Zhang, Y.: ROS-based UAV control using hand gesture recognition. In: 2017 29th Chinese Control And Decision Conference (CCDC), pp. 6795–6799. IEEE (2017)