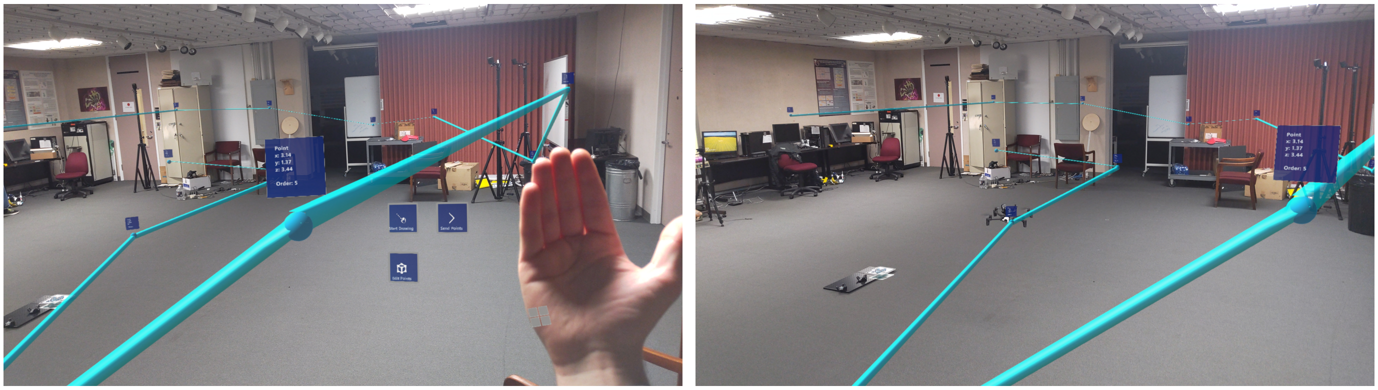
Drone Brush: Mixed Reality Drone Path Planning

Angelos Angelopoulos, Austin Hale, Husam Shaik, Akshay Paruchuri, Ken Liu, Randal Tuggle, and Daniel Szafir
Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, North Carolina, USA
{aangelos,haleau,hshaik,akshay,yiwk321,rtuggle,dszafir}@cs.unc.edu



(a) View of the interface with the hand menu displayed.   (b) View of the drone flying to waypoints.

Fig. 1: First-person views from the HoloLens 2 showing the *Drone Brush* interface and a created path.

***Abstract*—In this paper we present *Drone Brush*, a prototype mixed reality interface for immersive planning of drone paths for tasks such as collaborative photogrammetry and inspection. This interface employs Microsoft's HoloLens 2 to allow users to draw paths for drone navigation in 3D using hand gestures. Users can place waypoints with a simple pinch gesture, and similarly, delete and move existing waypoints. To validate paths, we leverage the HoloLens spatial map to check for potential collisions ahead of time, greatly reducing the likelihood of a collision during drone navigation. Paths are simplified and cleaned up using density-based clustering to prevent complex or redundant drone movement. In this Late-Breaking Report, we present the design and implementation of our system that integrates mixed reality, natural hand gestures, and drone path planning, which we plan to evaluate in a user study in the near future.**

***Index Terms*—Drone path planning; Mixed reality (MR); Augmented reality (AR); Human-drone interaction; Robot navigation**

I. Introduction

Aerial robots (i.e., drones) are becoming indispensable tools for a wide variety of real-world human-robot interaction tasks, such as collaborative photogrammetry, environmental inspection, and search or surveillance activities. In this research, we are interested in exploring new interfaces that may better support humans and drones in effectively communicating with each other to navigate environments and complete such tasks. To date, the majority of drone navigation and path planning interfaces have been designed around joystick controllers and/or smartphone displays. Demonstrations of additional control modalities, such as brain-computer interfaces, voice-based interfaces, or gesture-based interfaces, have been explored in the past decade (e.g., [1]), but have not been adopted due to issues such as the lack of specialized hardware systems and the inability of such interfaces to provide visual information while keeping the real world in focus.

Mixed reality systems enable virtual graphics to be projected directly within the context of the task, which has shown benefits in applications such as guidance in head and neck surgery [2], drone-augmented human vision [3], and collocated robot teleoperation [4]. As a result, we believe mixed reality headsets are suitable candidates to host interfaces as a part of human-robot interaction with collocated robots. Specifically, drone navigation and path planning can potentially become less encumbering and more enjoyable with mixed reality headsets such as the HoloLens 2 (HL2).

This work aims to take a step forward in improving human-robot interactions and presents *Drone Brush*, a mixed reality interface prototype for immersive drone path planning. The *Drone Brush* interface employs a HL2 to keep pertinent, real-world spatial information in view and anchor robot command sequences to the user's environment using virtual graphics. Users draw and modify paths for drone navigation in 3D using intuitive hand gestures such as pinching. Subsequently, paths are validated to ensure collision-free operation of the drone. In our future research, we plan to evaluate *Drone Brush* in user studies that feature tasks that rely on the use of drones collocated in the user's environment. Through this research and future user studies, we aim to answer two research questions of interest:

RQ1: How does immersive path planning in 3D with a mixed reality headset impact task completion speed and accuracy for co-located drone operation?

RQ2: Can users intuitively guide a drone indoors through and past obstacles using a mixed reality headset and natural hand gestures for path planning?

In summary, this work currently contributes the following:

- A mixed reality interface for drone path planning, demonstrated on a mixed reality HMD that captures natural hand gestures.
- A path validation approach that leverages a density-based clustering algorithm, DBSCAN, and the HL2 spatial mapping capabilities to validate drone paths as operable and to avoid collisions.

II. Related Works

User interfaces for robot control have been an active area of research for the past few decades [5]–[7]. In particular, there is a continued need for natural user interfaces for drones to allow unskilled pilots to accurately operate drones without extensive training [1]. Proposed user interfaces for drones have evolved with the adoption of technologies advanced in the past decade, including natural language processing, gesture recognition, and machine learning to better interpret sensor streams that enable new, multi-modal interactions [1]. Examples of such interfaces include voice-based interfaces for drone mission planning [8], gesture-based interfaces for remote robot control [9], and interfaces for sketching and planning paths in detail for applications such as cinematography using drones [10]. Despite various benefits proposed by the aforementioned interfaces, such interfaces are limited by the lack of spatial information for collocated users when planning paths. There has also been progress to explore more natural communication methods [11] and to make drones safer [12].

Mixed reality has also been an active research field for the past few decades with a recently renewed interest in the subject taking place due to significant advancements in computer vision and display hardware [13]. This has opened up opportunities at the intersection of mixed reality and robotics [14]. There has been work with mixed reality headsets toward better understanding robot intent using virtual imagery [15] and exploring the mediation of human-robot interaction using virtual surrogates [16]. These prior works do not explore the utility of natural hand gestures when combined with a mixed reality interface. To the best of our knowledge, there is no prior work involving drone path planning using mixed reality interfaces with natural hand gestures. As a result, we have decided to explore the usage of mixed reality interfaces with natural hand gestures, with the belief that their usage for both path planning and subsequent observation of the drone's intent will enable new human-robot interaction studies.

III. Drone Brush

The *Drone Brush* ecosystem has three primary components, as seen in Figure 2: a HoloLens 2 (HL2) Headset for the interface, a Parrot ANAFI quadrotor drone for navigation, and a host computer for bridging the system and controlling the

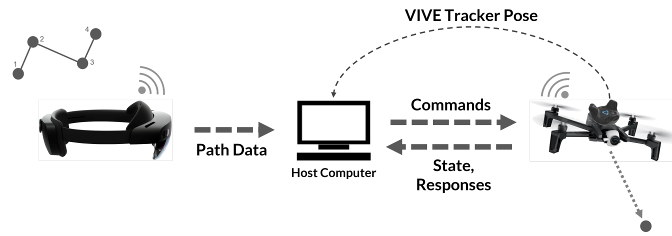


Fig. 2: Diagram of the *Drone Brush* system showing the components and their communication. The HL2 transmits path data to the host computer, which in turn sends commands to the drone for path navigation. The drone sends the computer state information and responses to commands indicating success or failure. The HTC VIVE tracker 2.0 mounted on the drone continuously transmits its pose to the host computer to localize the drone.

drone. The HL2 and the host computer communicate using ROS [17], whereas the Parrot ANAFI communicates through messages via the Parrot Olympe SDK [18]. The user can create, manipulate, and validate paths in mixed reality using the HL2 and use gestures to press a virtual button to send path data to the host computer for execution with the drone. The system requires a live pose of the drone for executing navigation, which in our prototype is provided by an HTC VIVE tracker 2.0 mounted on the drone.

A. Mixed Reality Interface

To aid mobile users in navigating their environment, we aim to create an easily accessible user interface that is always within arm's reach of the user. Thus we utilize a hand menu that quickly appears when the user faces their palm toward the display, in addition to requiring eye gaze and a flat palm to prevent false activation. The main menu allows users to navigate between creating points, editing a path sequence, and sending the ordered points to the drone. To place a point on the path, we use the HL2's built-in hand tracking to calculate the distance between the index finger and thumb locations and set a threshold for registering a pinch motion. To prevent unwanted point creations, the user can specify their pinching hand in the path drawing submenu. Moreover, we set constraints on the minimum distance (10 cm) between two consecutive points and when a user completes the pinching motion.

After adding points to the path, a procedural cylinder mesh connecting the points generates. The mesh creates a visual line that represents the path. We offer editing tools to delete, clear all, and move points. The mesh regenerates after creating, moving, or deleting a point and when the updated spatial mesh processes every few seconds. Path validation occurs before regenerating the mesh, so users can safely send their drawn path to the drone at any time. If a path is invalid, we notify the user that drone navigation cannot happen due to collisions with the environment. We built the interface using Unity with the Mixed Reality Toolkit (MRTK) [19].

(a) (b)

Fig. 3: (a) Main menu of the hand-attached UI for adding points, sending ordered point locations in the drone coordinate space to the drone, and editing created points. (b) Editing options to move, clear, or delete points on an existing path.

B. Localization

For a drone to follow a 3D path drawn in mixed reality, we need the precise position and orientation of the drone relative to the HL2. This is achieved by aligning the VIVE tracker coordinate space V^3 and drone coordinate space D^3 to the HL2 coordinate space H^3 using the calibration apparatus seen in Figure 4.

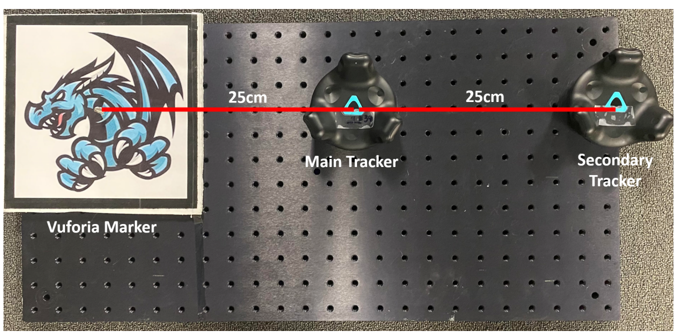


Fig. 4: Calibration apparatus to align the drone and HL2.

We first rigidly attach a HTC VIVE Tracker 2.0 (VT) called Drone Tracker (DT) to the drone as seen in Figure 5 and then use forward kinematics to extrinsically calibrate and transform the position and orientation of the DT and drone to H^3 [20]. The calibration apparatus has two rigidly attached VTs (Main Tracker and Secondary Tracker) on the same vertical and horizontal axis with known offsets relative to each other, thus leaving only one rotational axis (yaw) to be solved to complete the calibration.

When the RGB camera on the HL2 first recognizes the Vuforia marker (VM) we use the inverse position of the VM to get its position relative to the HL2 and generate the following kinematic chain: Main Tracker (MT) $\Rightarrow$ Secondary Tracker (ST), Drone Origin (DO) $\Rightarrow$ Drone Tracker, Path Points (note that the DO is the first local transform of the DT). We then use the vector between MT and ST to solve for yaw and align V^3 and H^3. Next, we use the local transform of the drone using DT relative to DO to align D^3 to H^3 thus getting the position and orientation of the drone relative to HL2. Since the path points are also relative to DO, no additional computation is needed for the drone to fly towards a specified point in space.

Fig. 5: The Parrot ANAFI drone used with the HTC VIVE tracker 2.0 mounted on the top.

C. Path Validation

Validating planned paths can increase the safety and efficiency of drone navigation. Path validation can prevent complex or redundant motions the drone should not perform.

We first want to remove cluttered points and simplify the path. Through a density-based clustering algorithm, DBSCAN [21], we detect point clusters based on the number of neighbors a point has within a certain radius. We manually set this radius to 10 cm, but in future iterations giving the user the ability to adjust this threshold is important so the system can adjust for different levels of task precision. We then clear dense clusters by replacing them with a point located at the cluster's mean. We only change a point's position if it is not in the same cluster as the previous point, and we delete it from the path otherwise. This way, we can eliminate rapid and consecutive direction changes, creating a straightened path while maintaining point ordering.

Spatial mapping allows us to identify real-world surfaces in the environment that act as potential collision points for the drone, as seen in Figure 6. For validating existing paths, we first create a width and height buffer for the drone to prevent it from being too close in proximity to the spatial mesh. Facing in the direction from one point to the next, we send five ray casts along the path, one from the point origin and one for each edge on the width and height constraint's 2D plane, to check for collisions with the spatial mesh layer. In the presence of collisions, we apply incorrect red color vertices to the path's mesh between these two points.

To further validate paths, we have the capability to visualize a simulated drone flying in mixed reality along the drawn path to each waypoint. The virtual drone uses the pose from the Parrot Sphinx simulation to model realistic drift and motion.

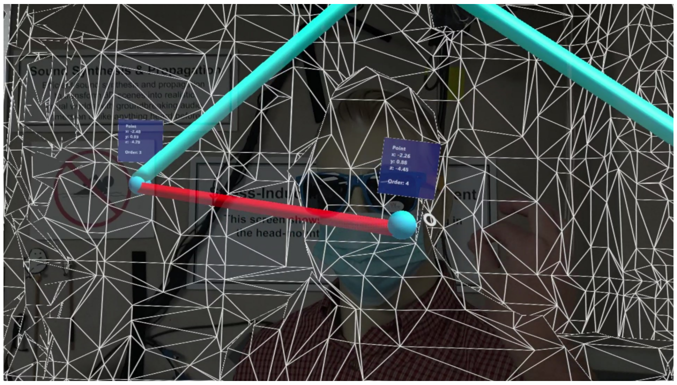


Fig. 6: An invalid path colored red between two points collides with the spatial mesh given a size constraint for the drone.

This feature can allow users to preview the drone motion on created paths, enabling another method for the detection and correction of potential collisions prior to execution with the physical drone.

D. Drone Navigation

Given a sequence of points, we want to navigate the drone to each one in order. We implemented proof-of-concept drone navigation to waypoints using the Parrot ANAFI drone, the Parrot Olympe SDK, and ROS. We also used the Parrot Sphinx simulator [22], which is based on Gazebo [23], to simulate the drone before using the physical system. The host computer connected to the drone via Wi-Fi controls the drone via Olympe commands. This computer is also part of the ROS network, allowing it to communicate with the HL2. We use the Unity ROS TCP connector and TCP endpoint to connect Unity with the ROS network. We connect devices to a dedicated local router for low latency in the ROS network.

The drone d starts at an origin pose d_0. When the host computer receives waypoints $W = \{w_i\}_{i=1}^{n}$ from the HL2, the computer assumes control of the drone. The drone takes off and waits a few seconds to stabilize. Subsequently, the drone navigates to each waypoint by first rotating to face it and then moving towards it. Since the drone faces the waypoint, it only needs to move forward or vertically, making the motion more predictable and ahead of time path validation useful. We can further increase motion predictability by optionally decomposing simultaneously horizontal and vertical movement. When both horizontal and vertical displacement is required and vertical displacement exceeds a threshold, we can insert an intermediate waypoint to decompose the drone's motion into its vertical and horizontal steps. The decomposition maintains rectilinear motion and avoids the curved trajectories that the Parrot ANAFI would otherwise make to move diagonally while changing altitude, which are harder to validate.

The computation of Δ_{yaw}, the minimum rotation necessary to face a target waypoint w_i works as follows: First, we compute $\theta = tan^{-1}(\frac{w_{i,y}-d_y}{w_{i,x}-d_x}) - d_{yaw}$. If $|\theta| \geq \pi$, then $\Delta_{yaw} = -\frac{\theta}{|\theta|}(2\pi - |\theta|)$, otherwise, $\Delta_{yaw} = \theta$. To control for error in the on-board sensors of the drone, we continuously compute the Euclidean distance of the drone (based on the VIVE tracker position) from w_i. Movement continues until a configurable distance threshold is met.

IV. Discussion

Drone Brush is currently a functional prototype demonstrating the use of gesture-based control for mixed-reality interaction with a drone. There are many ways to improve the system. Point annotations (e.g., to rotate the drone at a given point and take a picture) enable photogrammetry and inspection. Automatic mission-preserving path correction for avoiding collisions while maintaining path structure and direction can save time. Lastly, extensions that allow collaborative path planning in a shared space can make planning longer paths easier. Work also remains on the drone control system. The overall solution would improve with real-time obstacle avoidance, better drone localization, and more robust drone navigation. All of these are open problems in drone research.

We believe that *Drone Brush* can aid the automation of tasks such as photogrammetry and inspection for enterprises. For example, users can define paths for automatically patrolling specific areas or routinely navigating to certain equipment to take pictures from certain angles in industrial settings. Users wearing a HL2 can walk around areas of interest to create, view, and edit paths. Azure spatial anchors can enable large-scale path planning in larger environments. The system could also be adapted to work with miniaturized versions of large environments such as building exteriors.

Overall, the mixed reality gesture-based approach for this problem may be more intuitive and easier to learn and master as it is more natural. The ability to leverage mixed reality features such as real-time spatial mapping can offer real-time path validation and simulation that is likely impossible with other approaches, increasing reliability. We expect that this method has advantages over other solutions using computer monitors, keyboard and mouse, gamepad, or VR with VR controllers. We plan to perform a user study to evaluate *Drone Brush* and compare it with other approaches. This study would help uncover how easy each interface is to learn, how intuitive it is to use, and its effectiveness in aiding task accomplishment.

V. Conclusion

We present *Drone Brush*, a mixed reality system for drone path planning using hand gestures. This interface offers an immersive way to create drone paths with simple hand gestures using the HL2. We use the HL2 spatial map to check for potential drone collisions with the environment ahead of time. Dynamic path validation can allow the use of simpler drones that may not have real-time collision avoidance to execute planned paths. We also implemented a proof-of-concept drone navigation system for the Parrot ANAFI drone. We believe there is much promise for further design, work, and study on this system and mixed reality interfaces in general, and we intend to collect and share results from future experiments that evaluate our proposed system against more traditional drone teleoperation systems.

REFERENCES

[1] D. Tezza and M. Andujar, "The state-of-the-art of human–drone interaction: A survey," *IEEE Access*, vol. 7, pp. 167 438–167 454, 2019.

[2] A. S. Rose, H. Kim, H. Fuchs, and J.-M. Frahm, "Development of augmented-reality applications in otolaryngology–head and neck surgery," *The Laryngoscope*, vol. 129, pp. S1–S11, 2019.

[3] O. Erat, W. A. Isop, D. Kalkofen, and D. Schmalstieg, "Drone-augmented human vision: Exocentric control for drones exploring hidden areas," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 4, pp. 1437–1446, 2018.

[4] H. Hedayati, M. Walker, and D. Szafir, "Improving collocated robot teleoperation with augmented reality," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 78–86.

[5] T. B. Sheridan, "Teleoperation, telerobotics, and telepresence: A progress report," *Control Engineering Practice*, vol. 3, pp. 1–8, 1992.

[6] T. Fong, C. Thorpe, and C. Baur, "Multi-robot remote driving with collaborative control," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 4, pp. 699–704, 2003.

[7] P. G. De Barros and R. W. Lindeman, "A survey of user interfaces for robot teleoperation," *Worcester Polytechnic Institute*, 2009.

[8] B. Huang, D. Bayazit, D. Ullman, N. Gopalan, and S. Tellex, "Flight, camera, action! using natural language and mixed reality to control a drone," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6949–6956.

[9] J. DelPreto and D. Rus, "Plug-and-play gesture control using muscle and motion sensors," *2020 15th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 439–448, 2020.

[10] Q. Galvane, C. Lino, M. Christie, J. Fleureau, F. Servant, F. o.-l. Tariolle, and P. Guillotel, "Directing cinematographic drones," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 3, pp. 1–18, 2018.

[11] J. R. Cauchard, J. L. E, K. Y. Zhai, and J. A. Landay, "Drone & me: an exploration into natural human-drone interaction," in *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*, 2015, pp. 361–365.

[12] P. Abtahi, D. Y. Zhao, J. L. E, and J. A. Landay, "Drone near me: Exploring touch-based human-drone interaction," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, pp. 1–8, 2017.

[13] M. Billinghurst, A. Clark, and G. Lee, "A survey of augmented reality," 2015.

[14] D. Szafir, "Mediating human-robot interactions with virtual, augmented, and mixed reality," in *International Conference on Human-Computer Interaction*. Springer, 2019, pp. 124–149.

[15] M. Walker, H. Hedayati, J. Lee, and D. Szafir, "Communicating robot motion intent with augmented reality," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 316–324.

[16] M. E. Walker, H. Hedayati, and D. Szafir, "Robot teleoperation with augmented reality virtual surrogates," in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2019, pp. 202–210.

[17] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," *ICRA workshop on open source software*, vol. 3, no. 3.2, p. 5, 2009.

[18] Parrot SA, "Olympe," Accessed December 2 2021. [Online]. Available: https://developer.parrot.com/docs/olympe/overview.html

[19] Microsoft Corporation, "MRTK-Unity," Accessed December 2 2021. [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/?view=mrtkunity-2021-05l

[20] N. Rewkowski, A. State, and H. Fuchs, "Small marker tracking with low-cost, unsynchronized, movable consumer cameras for augmented reality surgical training," in *2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE, 2020, pp. 90–95.

[21] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *KDD*, vol. 96, no. 34, 1996, pp. 226–231.

[22] Parrot SA, "Sphinx," Accessed December 2 2021. [Online]. Available: https://developer.parrot.com/docs/sphinx/index.html

[23] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154 vol.3.