2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

3D Autocomplete: Enhancing UAV Teleoperation with AI in the Loop

Batool Ibrahim, Imad H. Elhajj, Daniel Asmar

***Abstract*— Manually teleoperating a flying robot can be a demanding task, especially for users with limited levels of experience. This is primarily due to the non-linear properties of such robots in addition to the difficulty of controlling various degrees of freedom at the same time. 3D Autocomplete helps mitigate such limitations by assisting the users in teleoperation. It aids in teleoperating 3D motions, such as helical motions, which are more challenging to the users. The proposed framework uses Artificial Intelligence (AI) to predict just-in-time the user's intended motion and then, if the user accepts, completes it autonomously in 3D. The AI component of 3D Autocomplete was presented in our previous work, where we introduced a deep learning model and an algorithm to predict as early as possible the user's desired motion. Moving forward in this work, we focus on synthesizing and completing the user-intended motion autonomously. Also, we introduce a Mixed Reality (MR) user interface for better human-robot interaction. Finally, we evaluate our system subjectively and objectively through human-subject experiments. Autocomplete outperformed traditional method on all criteria with at least 30% improvement in all objective measures.**

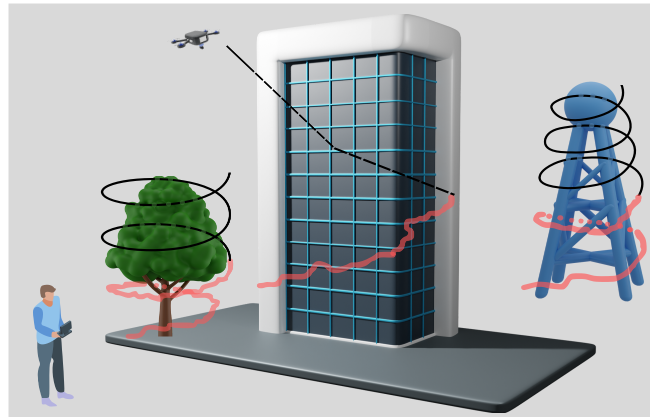


Fig. 1: 3D Autocomplete: the user's manual teleoperated motions are represented in red, whereas the 3D Autocomplete autonomous motions are represented in black. The motion around the tree is considered as a motion around a cylinder, the building as a box, and the antenna as a cone.

I. INTRODUCTION

Human-Robot Interaction (HRI) is an evolving field in robotics [1] and arises from the need to enhance the communication between humans and robots, primarily in application areas that require human-robot collaboration [2]. One of the important areas in HRI is robot teleoperation. In fact, until now, robot teleoperation from a remote location is still a challenging task, especially when controlling Unmanned Aerial Vehicles (UAVs) [3].

Despite the fact that UAVs are already used in various applications such as loads transportation [4], farming [5], and surveillance [6]; their teleoperation remains a demanding task that requires a certain level of experience. This is due to the limited perceptual capability at the operator's end [7], in addition to the high dimensional and nonlinear dynamical properties of such robots. Accordingly, for the purpose of mitigating these limitations and assisting the users while remote controlling a UAV, Autocomplete was proposed in our previous work [8].

Autocomplete aids UAV operators by predicting their desired motions and completing it in an autonomous manner. In more detail, while a person is teleoperating a UAV, Autocomplete uses the entered joystick commands to predict the intended-motion type with the help of Artificial Intelligence (AI). Then, the framework suggests the predicted motion to the user through a User Interface (UI), and if they accept, automatically synthesizes and completes the UAV's motion.

Batool Ibrahim, Imad H. Elhajj, and Daniel Asmar are with the Vision and Robotics Lab, American University of Beirut, Beirut, P.O. Box 11-0236 Lebanon e-mails: {bai04, ie05, da20}@aub.edu.lb.

Initially, our proposed Autocomplete framework relied on a Support Vector Machine (SVM) to classify the user's desired motion as one of predefined motion primitives: straight line, sine, or arc motions. The framework was further developed in [9], where Deep Learning (DL) was used for the prediction of the user's desired motion instead of SVM, and Mixed Reality (MR) was introduced to the system for a better HRI. Moreover, online and incremental learning were implemented in [10] and [11] to improve the performance of the DL model. Throughout the preceding works, Autocomplete focused on predicting and completing 2-dimensional motions; motions in the same plane (arc, line, etc). However, in practice, UAVs are used to navigate in a 3-Dimensional (3D) world which requires executing more complex 3D motions (see Fig. 1). Such as, flying around a tower, a building, or a tree. In such cases, teleoperation becomes more challenging for the operators, and the possibility of collisions increases, thereby motivating the need for an assistive tool.

In this regard, the 3D Autocomplete framework was introduced [12] and supports 3D motions covering different 3D geometrical shapes. These shapes are basic geometric primitives commonly encountered in any 3D environment: box, cylinder, and cone. Handling such primitives enables the system to negotiate 3D objects with similar geometrical properties, like buildings (box) or towers (cylinder); in addition to more complicated geometrical shapes that are formed of combinations of those basic elements (a big tree can be considered as a cylinder followed by a cone).

3D Autocomplete framework uses a real-time algorithm to monitor the UAV's motion, then relies on DL to classify its type [12]. In this paper, we proceed to make the overall framework capable of synthesizing and completing the user's predicted motion. Moreover, we introduce to the system a mixed reality UI for a more efficient teleoperation experience. Finally, we validate 3D Autocomplete's effectiveness, objectively and subjectively, via human-subject experiments conducted by quadrotor simulations on ROS/Gazebo. The main contributions of this paper include:

- We propose a motion synthesis algorithm for 3D Autocomplete that generates the trajectory needed to complete the user's desired 3D motion autonomously.
- We introduce in 3D Autocomplete a mixed reality UI that enables the operator to observe the 3D predicted motions augmented directly to the surrounding environment scene using an MR headset.
- We conduct human-subject testing to validate our proposed framework qualitatively and quantitatively. Such tests are essential to ensure the functionality and usability of 3D Autocomplete in enhancing human-robot interaction.

The remaining part of this paper is structured as follows: Section II provides an overview of related works in the literature. Section III outlines the proposed system. Section IV presents a detailed analysis of the experimental setup and results. Finally, Section V offers conclusions and future directions.

II. Literature Review

This section reviews two fields that are relevant to our proposed framework: assistive teleoperation and MR in HRI.

A. Assistive Teleoperation

Given the widespread usage of teleoperated robots (industrial manufacturing, agriculture, military, etc), assistive teleoperation has been addressed using different approaches. Zhang et al. [13] use Implicit Neural Field (INF), which relies on neural networks to assist humans in teleoperating surgical robots. The system takes as an input the $'$leading hand$'$ commands conducted by the operator, and corrects them with the help of INF to prevent any collisions with the human tissues. In this approach, the users have to teleoperate the robot manually throughout the entire task, while the system is optimizing their path. In contrast, our proposed 3D Autocomplete requires only a partial user motion as a manual input, and then can predict the desired motion type and complete it autonomously.

Maeda in [14] proposes a policy blending with primitives strategy that combines the control policies of the robot and the human using dynamical movement primitives. These primitives allow the robot to execute complex movements by breaking them down into sub-movements while blending the real-time user inputs. A similar approach is introduced in [15], where Gottardi et al. present a system that generates a sequence of safe points to reach a specific goal while considering the user's intentions. The proposed systems are focused on applications that involve targeting multiple goals or dynamic obstacle avoidance. However, it is not suitable for other application types such as surveillance or robot racing.

Ewerton et al. [16] proposes a control strategy that enhances robotic demonstrations using Reinforcement Learning (RL). They address the changes in the robot's surroundings after conducting a demonstration, in addition to non-optimal demonstrations. The introduced RL approach relies on Pearson correlation to determine the relevance functions, which are used to provide online correction and adapt to the new environment. Such a framework is task centered and requires different demonstrations for different tasks. On the other hand, 3D Autocomplete is user-centered; it completes the user's intentions despite the considered task.

Aligning with the focus of our study, Wang et al. [17] address assistive teleoperation in flying robots. They propose a system that utilizes the human gaze and the remote controller commands to generate a path which satisfies the operator's intentions. The gaze captures the targeted position, whereas the remote controller input is used to identify the intended speed. Real world experiments have verified the reliability and robustness of the proposed framework. However, such a system relies on the accuracy of the operator's gaze data which may be affected by several factors such as the lighting conditions or eye movements. Additionally, the system may not be suitable for all types of UAV applications such as long range flights.

B. Mixed Reality in HRI

The recent technologies in MR have allowed humans to blend virtual objects with their real surroundings. This has unveiled a new level of interaction and experience in HRI fields. MR is used in manufacturing [18], medical surgeries [19], education [20], etc. However, only a few works have considered MR in the context of assisted teleoperation. Szczurek et al. [21] introduce a framework that assists operators in teleoperating mobile robots in dangerous environments. They present a mixed reality UI that enables both individual and multiple users to interact with 3D holographic representations of the controlled robot and its surrounding environment. Experiments have proven the effectiveness of the proposed system in enabling stable remote teleoperation. In other similar work, Sun el al. [22] employ an MR interface with an interaction proxy for teleoperating industrial robots. The proposed system utilizes a Head-Mounted Display (HMD) interface with tracking technology for both head and hand gestures in the virtual environment. While these works use MR to enable remote teleoperation within a constructed scene, our proposed system directly augments the user's desired trajectory into the real-world scene of the robot's camera.

Hedayati et al. [23] introduce an augmented reality UI to improve the teleoperation of UAVs. In their framework, they integrate information from the robot's visual stream into the user's perspective of the UAV and its surroundings. Additionally, the system provides real-time visual feedback on the UAV camera functionalities. This allows for more effective

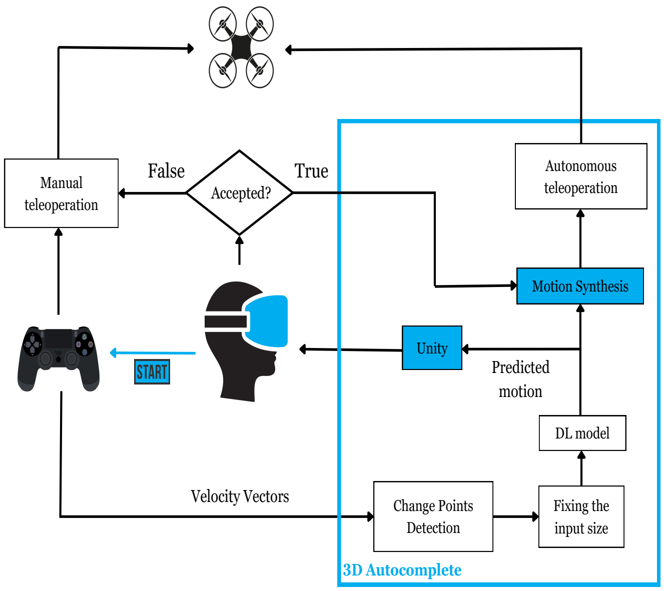


Fig. 2: 3D Autocomplete Framework

robot control and task completion. Both quantitative and qualitative results have validated the effectiveness of their system in enhancing the teleoperation of UAVs. Similar work is presented in [24]. Moreover, Lee et al. [25] introduced an innovative approach to robot teleoperation by incorporating virtual fixtures into the environment scene. They propose that integrating virtual fixtures is aimed at improving user understanding. The framework is suggested to be used in very sensitive challenges such as cleaning up nuclear waste.

The weakness of the works mentioned earlier lies in their exclusive emphasis on enhancing the user's perspective, and presuming constant visibility of the robot within the user's field of view (FOV). However, this approach may fail when the robot is situated far away from the control station. In contrast, our proposed UI adopts a first-person view approach. The operators observe a live video feed from the robot's built-in camera using a mixed reality headset. This setup allows the user to maintain a view of the robot's environment even when the robot is not within the FOV.

III. System Architecture

The system structure of 3D Autocomplete is illustrated in Fig. 2. The framework takes as input the joystick commands conducted by the user and monitors it using a change point detection algorithm (just-in-time). When significant changes are detected, the input motion is forwarded to the DL model to predict its type (as one of predefined motion primitives) after resizing it to match the model's input dimensions. Then, 3D Autocomplete proposes the predicted motion through a UI, giving the user the choice to either accept, decline, or ignore. If the user accepts, the system synthesizes the suggested motion and directs it to the autonomous pilot to execute. The components of 3D Autocomplete are addressed in the following subsections.

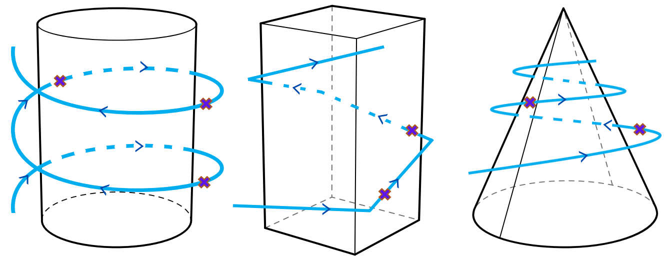


Fig. 3: Spotting important change in the user 3D motion: a change point, presented by an orange cross, is detected when a change in the drone velocity vectors directions takes place.

A. Motion Classifier: Deep Learning Model

Deep learning is employed in our system to predict the user's desired motion type. To achieve this, we collected a dataset of 3D motion primitives, which includes motions around 3D geometrical shapes: cylinders (cylindrical motions), cones (conical motions), and boxes (rectangular motions). Moreover, it accounts for different orientations, dimensions, and directions to ensure its generalizability. Then, the collected dataset was used to train a deep learning model, which is constructed by combining recurrent neural networks and one-dimensional convolutional neural networks. The trained model achieved an 80% training accuracy and a 70% validation accuracy. Additionally, it obtained an 89% F1-score on the testing data. Details of this approach are presented in [12].

B. Just-in-Time Detection

3D Autocomplete predicts the user desired motion from the partial motion conducted by user. Initially, a constant time interval was considered to do the motion prediction. However, this could lead to very early or very late predictions depending on the motion type, the size of the shape considered, the velocity of the UAV, etc. Therefore, to determine when to make this prediction, 3D Autocomplete employs a Just-in-Time algorithm to continuously monitor the user input in real-time and detect significant changes within it. As a result, the system predicts the user's motion type after identifying these important changes in the motion (Fig. 3.). These points signify a change in the motion direction, indicating that a portion of the shape is already covered. The algorithm is detailed in our earlier work [12].

C. Variable Motion Vector Length

3D Autocomplete handles motions with varying sizes. This is because it predicts the motion type after detecting change points, which can vary between different motions. However, deep learning models have a fixed input size. Consequently, 3D Autocomplete performs up/down-sampling of motions after detecting change points to ensure they match the input size of the deep model, enabling accurate motion type prediction [12].

D. Motion Synthesis

After predicting the motion type by the deep learning model, Autocomplete synthesizes the user-intended motion, if the user accepts it. Synthesizing the motion generates an estimated 3D trajectory that fits well the user's partial motion and the user's intentions. This allows the UAV to complete the desired motion autonomously by following this generated trajectory. The 3D motions are synthesized by fitting the partial motion done by the user into the predicted motion type: motion around a cylinder, motion around a cone, or motion around a box. These motions are presented as a cylindrical helix, a conical helix, and a rectangular helix, respectively. For example, if the predicted motion is a one around a cylinder, Autocomplete fits the user's partial motion into a cylindrical helix. This will estimate the parameters that define this motion, such as the radius of the helix, the pitch, etc. Thus, the rest of the trajectory can be estimated based on these parameters.

a) Motion Around a Cylinder: Cylindrical Helix: a cylindrical helix has a constant radius and pitch. It is defined by the following parametric equation:

$$\begin{cases} x = r \cdot \cos(t) \\ y = r \cdot \sin(t) \\ z = p \cdot t \end{cases} \tag{1}$$

where p is the pitch of the helix, r is the radius, and $t \in [0, 2\pi)$. Fitting the data into a cylindrical helix involves estimating the parameters r and p. Accordingly, Least Square (LS) optimization is used to find their optimal values. LS fits the input data into the predicted motion by minimizing a sum of the squared errors objective function. The objective function is presented below:

$$f(r,p) = \sum_{i=1}^{n} \left[(x_i - x'_i)^2 + (y_i - y'_i)^2 + (z_i - z'_i)^2 \right] \tag{2}$$

where n is the number of data points. Levenberg-Marquardt (LM) is used to minimize this function. It is one of the most effective algorithms for non-linear LS problems.

b) Motion Around a Cone: Conical Helix: a conical helix has a variable radius which increases/decreases exponentially over time. It is defined by the following parametric equation:

$$\begin{cases} x = r \cdot e^{-k \cdot t} \cdot \cos(t) \\ y = r \cdot e^{-k \cdot t} \cdot \sin(t) \\ z = p \cdot t \end{cases} \tag{3}$$

where p is the pitch of the conical helix, r is the radius, k controls the tightness or wideness of the helix, and $t \in [0, 2\pi)$. Fitting the data into a conical helix involves estimating the parameters r, p, and k. To do so, the same approach is used as that of the cylindrical helix.

c) Motion Around a Box: Rectangular Helix: a rectangular helix has a rectangular cross-section. It is defined by the following parametric equation [26]:

$$\begin{cases} x = a \cdot \max\left(-1, \min\left(\frac{4}{\pi} \arcsin\left(\sin\left(\frac{\pi t}{2} + \frac{\pi}{4}\right)\right), 1\right)\right) \\ y = b \cdot \max\left(-1, \min\left(-\frac{4}{\pi} \arcsin\left(\cos\left(\frac{\pi t}{2} + \frac{\pi}{4}\right)\right), 1\right)\right) \\ z = p \cdot t \end{cases} \tag{4}$$

where p is the pitch of the rectangular helix, a/b are the dimensions of a $2a \times 2b$ rectangle that represents the helix cross-section, and $t \in [0, 2\pi)$. LS and LM are also used to fit the user's partial motion into a rectangular helix by estimating the parameters a, b, and p.

Additionally, to account for the orientation and the position of the considered motion, we introduce a rotation matrix and a translation vector into all the parametric equations. This involves incorporating the Euler angles within a rotation matrix and the components of a 3D translation vector into the optimization problem. As a result, the system will have the estimated parameters of the user's motion in addition to its orientation and position. Accordingly, to estimate the future desired trajectory, the system sets the variable 't' to an 'upcoming' value, typically within the range of $[2\pi, 4\pi)$, and calculates the corresponding values for x, y, and z. This process generates target points relative to the UAV's current position, allowing autonomous completion of the user's desired motion.

E. Mixed Reality User Interface

Integrating mixed reality into teleoperation enhances the user's understanding of the robot's actions and its surrounding environment. This strengthens the interaction between humans and robots and thus improve the performance of the system [21]. Accordingly, we introduce MR UI to 3D Autocomplete. First, the real-time video stream from the UAV's onboard camera is presented to the user through a VR headset rather than the traditional screen. This experience eliminates distractions from the physical surroundings and enhances the user's focus during teleoperation. While 3D Autocomplete is active, the predicted 3D motion is augmented directly into the user's field of view. For example, if the predicted motion is a helix, 3D Autocomplete will display a helical motion within the augmented environment (see Fig. 4).

To take into account the motion direction while displaying the predicted class (a helix to the left, a helix to the right, etc.), we utilize Principle Component Analysis (PCA) to identify the major axis of the motion. Since the primary aim is to convey to the operator the 'rough' direction of motion without displaying the precise predicted motion, we consider eight orientations for each motion: $0°, 45°, 90°, 135°, 180°, 225°, 270°$, and $315°$. The augmented motion is then selected based on the nearest angle. For

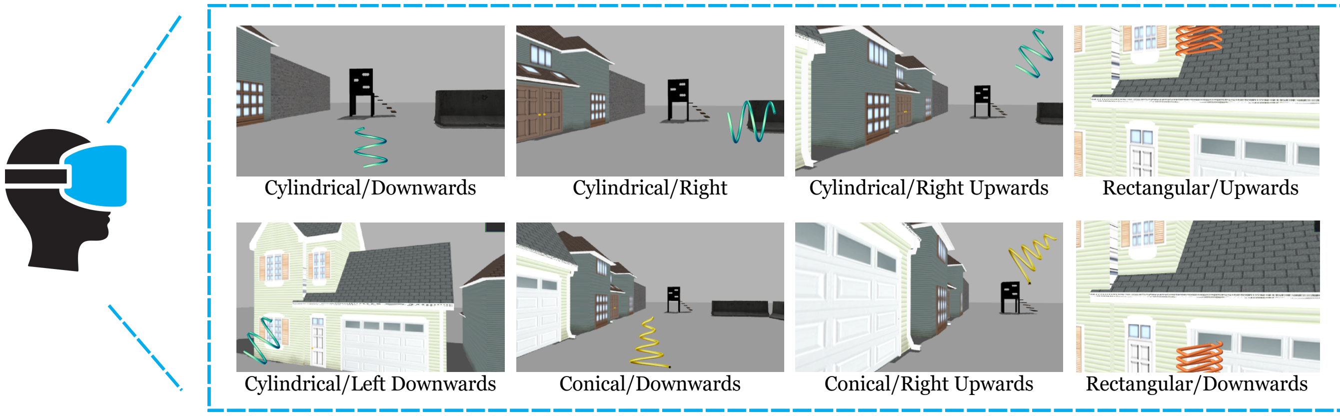


Fig. 4: Proposed User Interface for 3D Autocomplete

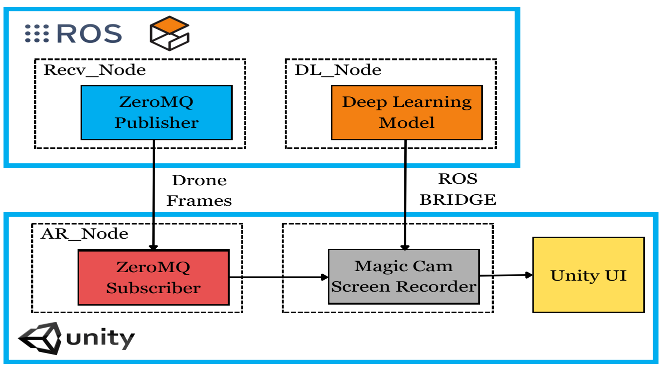


Fig. 5: Components of the Proposed MR user Interface

implementation, Unity with Vuforia Engine SDK was used to augment the 3D motions into the user's scene. To do so, we emulate the existence of a physical camera using ZeroMQ (ZMQ) network protocol and Magic Camera software. This is because augmenting an object within the physical world through Unity, necessitates the presence of an AR Camera Object. Which is typically done when a physical camera is linked to the host PC. Accordingly, the ZMQ publisher on the ROS PC, where the UAV is controlled, transmits the frames captured by the simulated camera on Gazebo to the ZMQ subscriber on the Unity host PC. This will open a real-time video stream window utilizing openCV and captured by a Magic Camera screen recording tool. Then, the recorded video feed is transformed into a Virtual Webcam identified and utilized by Unity. Finally, to transmit the predicted motion class from ROS to Unity, it is transmitted from the DL node to Unity via ROS bridge (see Fig. 5).

IV. Results

This section details the conducted experiments to corroborate the benefits of 3D Autocomplete, in addition to the achieved subjective and objective results.

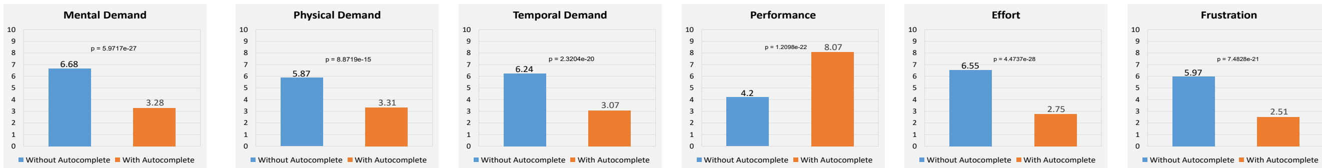
A. Experiments

To test 3D Autocomplete in real-time, we conducted human-subject testing within Gazebo virtual environment integrated with ROS. The primary objective of these experiments was to assess the effectiveness and efficiency of 3D Autocomplete compared to the traditional teleoperation without autocomplete. All human-subject testing was approved by the university's Institutional Review Board (IRB); approval ID SBS-2023-0163. All invited participants were above 18 years old and familiar with driving flying robots, but with limited levels of experience.

Each experiment (per operator) is divided into five trails. In each trail the operator is asked to fly a simulated AR.Drone 2.0. quadrotor around a 3D shape (a box, a cone, or a cylinder) twice (once through manual teleoperation and once using 3D Autocomplete). The shapes and their dimensions were chosen randomly, but in each new trail, a new shape with new dimensions is considered. The experimental procedures are as follows: (1) The participants are introduced to the experimental tasks and are given some time to adapt to the system and its components (joystick, VR headset), (2) a 3D shape is chosen randomly, in addition to randomly deciding which case to start with (3D Autocomplete or traditional teleoperation), (3) after each trial, the operator completes a survey to assess their experience subjectively. Note that in manual teleoperation, the user controls the drone using only the joystick. The same procedure is applied in all the experiments. The number of participants was 14, where we reached a total of 140 trials divided equally between Autocomplete and manual teleoperation.

B. Results

1) Subjective Evaluation: we subjectively validated our system using the NASA Task Load Index (NASA-TLX) which assesses a task by evaluating the perceived workload rates [27]. This assessment is essential in our study, given that the primary objective of 3D Autocomplete is to reduce user workload during teleoperation. Thus, after each trial,



(a) (b) (c) (d) (e) (f)

Fig. 6: NASA-TLX Survey Scores Obtained from Human-Subject Experiments

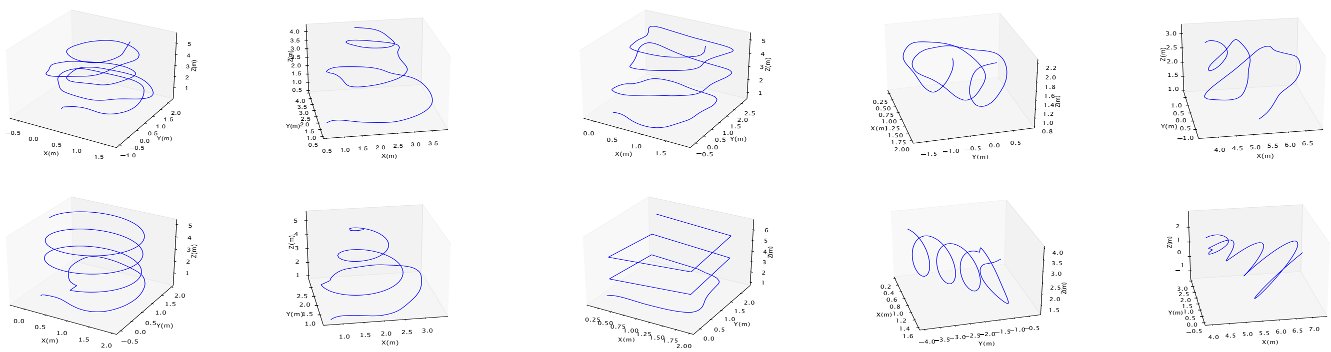


Fig. 7: 3D trajectories using only manual teleoperation (up); and trajectories with 3D Autocomplete (down)

users were asked to complete the NASA-TLX questionnaire (see Fig. 6). The results indicate that 3D Autocomplete outperforms traditional teleoperation by effectively reducing the demands across the considered subscales, while sustaining a high level of performance. This is because in 3D Autocomplete the users only have to perform a partial motion, instead of the entire motion, and the autopilot completes the remaining motion, which reduces the workload and effort. Also, the MR UI allows the users to be more focused on the task and isolated from the surrounding noises, which is reflected in the system performance. Each user experienced at least one collision during manual control (the trial was restarted in such cases), highlighting the challenges of UAV teleoperation. In contrast, collisions rarely occurred while using 3D Autocomplete and resulted from the user deviation from the targeted shape in their conducted partial motion.

2) Objective Evaluation: to evaluate our proposed system in an objective manner, we consider three metrics: the duration taken to complete the assigned task, the distance traveled by the UAV, and the smoothness of the conducted path (the degree of regularity). The distance reflects the user's convergence or divergence from the intended trajectory, while time and smoothness are significant parameters in optimizing the effectiveness of teleoperation missions. Consequently, we compare the results of the two teleoperation methods: traditional teleoperation and 3D Autocomplete.

The average time, distance, and smoothness of 70 tests per method (total of 140 iterations per shape) are shown in Table 1. Autocomplete outperforms manual teleoperation in average time, distance, and smoothness: approximately 35%, 30% and 50% less average time, distance, and smoothness angle for all shapes. Moreover, all obtained p-values (subjective and objective) are very small ($<<<< 0.005$) which suggests strong evidence contradicting the null hypothesis.

Fig. 7. presents some of the motions tested using manual teleoperation and 3D Autocomplete. The figure illustrates the challenges faced during user teleoperation, where the user's motions appear irregular and occasionally deviate from the desired trajectory. In contrast, when employing the 3D Autocomplete, the autopilot smoothly completes the user's intended motion after predicting its type from partial inputs.

TABLE I: Quantitative Results of the human-subject tests.

	Average time[sec]		Average Distance[m]		Average Smoothness[radians]	
	Manual	Auto	Manual	Auto	Manual	Auto
Cylinders	39.12	25.12	29.624	20.73	6450.493	3134.975
Cones	36.6	23.86	30.618	20.603	5010.52	2919.78
Boxes	36.68	22.5	27.45383	16.5568	6137.663	3125.872

V. CONCLUSIONS

In this paper, we introduced '3D Autocomplete' to assist users in teleoperating UAVs in a 3D environment. Our proposed system leverages AI to predict the user's desired motion as early as possible and autonomously completes it. Additionally, it employs mixed reality to enhance the users' interactions. The results obtained from human subject testing in ROS/Gazebo have validated our proposed system, demonstrating its quantitative and qualitative superiority over traditional manual teleoperation methods. In future work, we plan to extend our evaluation to real-life test scenarios.

ACKNOWLEDGMENT

This work was supported by the University Research Board (URB) at the American University of Beirut.

REFERENCES

[1] T. B. Sheridan, "Human–robot interaction: status and challenges," *Human factors*, vol. 58, no. 4, pp. 525–532, 2016.

[2] K. Hambuchen, J. Marquez, and T. Fong, "A review of nasa human-robot interaction in space," *Current Robotics Reports*, vol. 2, no. 3, pp. 265–272, 2021.

[3] H. T. Nguyen, T. V. Quyen, C. V. Nguyen, A. M. Le, H. T. Tran, and M. T. Nguyen, "Control algorithms for uavs: A comprehensive survey," *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 7, no. 23, pp. e5–e5, 2020.

[4] D. K. Villa, A. S. Brandao, and M. Sarcinelli-Filho, "A survey on load transportation using multirotor uavs," *Journal of Intelligent & Robotic Systems*, vol. 98, pp. 267–296, 2020.

[5] A. D. Boursianis, M. S. Papadopoulou, P. Diamantoulakis, A. Liopa-Tsakalidi, P. Barouchas, G. Salahas, G. Karagiannidis, S. Wan, and S. K. Goudos, "Internet of things (iot) and agricultural unmanned aerial vehicles (uavs) in smart farming: A comprehensive review," *Internet of Things*, vol. 18, p. 100187, 2022.

[6] X. Li and A. V. Savkin, "Networked unmanned aerial vehicles for surveillance and monitoring: A survey," *Future Internet*, vol. 13, no. 7, p. 174, 2021.

[7] T. Fong, C. Thorpe, and C. Baur, *Collaborative control: A robot-centric model for vehicle teleoperation*. Carnegie Mellon University, The Robotics Institute Pittsburgh, 2001, vol. 1.

[8] M. K. Zein, A. Sidaoui, D. Asmar, and I. H. Elhajj, "Enhanced teleoperation using autocomplete," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9178–9184.

[9] M. K. Zein, M. Al Aawar, D. Asmar, and I. H. Elhajj, "Deep learning and mixed reality to autocomplete teleoperation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4523–4529.

[10] M. H. Hussein, I. H. Elhajj, and D. Asmar, "Personalized autocomplete teleoperation: Real-time user adaptation using transfer learning with partial feedback," in *IEEE International Conference on CYBER Technology in Automation, Control, and Intelligent Systems*. IEEE, 2021.

[11] M. H. Hussein, B. Ibrahim, I. H. Elhajj, and D. Asmar, "Incremental learning for enhanced personalization of autocomplete teleoperation," in *2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[12] B. Ibrahim, M. H. Hussein, I. H. Elhajj, and D. Asmar, "Autocomplete of 3d motions for uav teleoperation," in *2023 IEEE/RSJ Intelligent Robots and Systems (IROS*, 2023.

[13] H. Zhang, L. Zhu, J. Shen, and A. Song, "Implicit neural field guidance for teleoperated robot-assisted surgery," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 6866–6872.

[14] G. Maeda, "Blending primitive policies in shared control for assisted teleoperation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9332–9338.

[15] A. Gottardi, S. Tortora, E. Tosello, and E. Menegatti, "Shared control in robot teleoperation with improved potential fields," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 3, pp. 410–422, 2022.

[16] M. Ewerton, G. Maeda, D. Koert, Z. Kolev, M. Takahashi, and J. Peters, "Reinforcement learning of trajectory distributions: Applications in assisted teleoperation and motion planning," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4294–4300.

[17] Q. Wang, B. He, Z. Xun, C. Xu, and F. Gao, "Gpa-teleoperation: Gaze enhanced perception-aware safe assistive aerial teleoperation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5631–5638, 2022.

[18] P. Wang, S. Zhang, M. Billinghurst, X. Bai, W. He, S. Wang, M. Sun, and X. Zhang, "A comprehensive survey of ar/mr-based co-design in manufacturing," *Engineering with Computers*, vol. 36, pp. 1715–1738, 2020.

[19] A. J. Lungu, W. Swinkels, L. Claesen, P. Tu, J. Egger, and X. Chen, "A review on the applications of virtual reality, augmented reality and mixed reality in surgical simulation: an extension to different kinds of surgery," *Expert review of medical devices*, vol. 18, no. 1, pp. 47–62, 2021.

[20] Z. Pan, A. D. Cheok, H. Yang, J. Zhu, and J. Shi, "Virtual reality and mixed reality for virtual learning environments," *Computers & graphics*, vol. 30, no. 1, pp. 20–28, 2006.

[21] K. A. Szczurek, R. M. Prades, E. Matheson, J. Rodriguez-Nogueira, and M. Di Castro, "Multimodal multi-user mixed reality human–robot interface for remote operations in hazardous environments," *IEEE Access*, vol. 11, pp. 17 305–17 333, 2023.

[22] D. Sun, A. Kiselev, Q. Liao, T. Stoyanov, and A. Loutfi, "A new mixed-reality-based teleoperation system for telepresence and maneuverability enhancement," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 1, pp. 55–67, 2020.

[23] 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.

[24] 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.

[25] D. Lee and Y. S. Park, "Implementation of augmented teleoperation system based on robot operating system (ros)," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 5497–5502.

[26] J. M. ain39;t a mathematician (https://math.stackexchange.com/users/498/j-m-aint-a mathematician), "Equation of a rectangle," Mathematics Stack Exchange, uRL:https://math.stackexchange.com/q/69134 (version: 2017-04-13). [Online]. Available: https://math.stackexchange.com/q/69134

[27] S. G. Hart, "Nasa-task load index (nasa-tlx); 20 years later," in *Proceedings of the human factors and ergonomics society annual meeting*, vol. 50, no. 9. Sage publications Sage CA: Los Angeles, CA, 2006, pp. 904–908.