Measuring Human-Robot Team Benefits Under Time Pressure in a Virtual Reality Testbed

Katarina Popović[1], Millicent Schlafly[1], Ahalya Prabhakar[1,2], Christopher Kim[1,3], and Todd D. Murphey[1]

***Abstract*—During a natural disaster such as hurricane, earthquake, or fire, robots have the potential to explore vast areas and provide valuable aid in search & rescue efforts. These scenarios are often high-pressure and time-critical with dynamically-changing task goals. One limitation to these large scale deployments is effective human-robot interaction. Prior work shows that collaboration between one human and one robot benefits from shared control. Here we evaluate the efficacy of shared control for human-swarm teaming in an immersive virtual reality environment. Although there are many human-swarm interaction paradigms, few are evaluated in high-pressure settings representative of their intended end use. We have developed an open-source virtual reality testbed for realistic evaluation of human-swarm teaming performance under pressure. We conduct a user study (n=16) comparing four human-swarm paradigms to a baseline condition with no robotic assistance. Shared control significantly reduces the number of instructions needed to operate the robots. While shared control leads to marginally improved team performance in experienced participants, novices perform best when the robots are fully autonomous. Our experimental results suggest that in immersive, high-pressure settings, the benefits of robotic assistance may depend on how the human and robots interact and the human operator's expertise.**

***Index Terms*—human-robot collaboration, human-robot teaming, multi-robot system, virtual reality, shared control.**

I. Introduction

The technological capabilities of unmanned aeral vehicles (UAVs) have increased, spurring interest in multi-robot systems. The expansion from single UAV deployment to multi-unmanned aerial systems has the potential to benefit a range of civilian applications, such as natural disaster support as well as situational awareness missions. Furthermore, robots are becoming a more integral part of human teams, acting in close physical proximity to humans in the field. Testing novel human-robot interaction algorithms in an experimental environment that resembles end-user scenarios is crucial for transferring effects of swarm assistance in controlled trials to the real world usage.

One of the ways current platforms differ from real-world scenarios is the absence of stressors like those induced in applications such as search and rescue missions [1]–[3], firefighting [4]–[6], and during other natural disasters (i.e., hurricanes, earthquakes, etc.) [7]–[9]. These situations are often time-critical and hazardous with continuously changing exploration goals, necessitating dynamic response and human expertise [3] as environmental conditions evolve. Many prior studies that assess human-swarm interactions assume that the operator is a passive, external observer with no additional involvement in task completion besides swarm management duties [10]–[13]. It is unclear if trends discovered in settings where the human is solely responsible for robot operation will translate to high-pressure, time-sensitive real world applications.

Here, we exploit virtual reality (VR) as a tool to enable the evaluation of human interaction with a swarm of three drones in a simulated environment that mimics features of real-world applications. While we have implemented end-to-end system architecture on hardware and demonstrated feasibility of our control paradigm [14], repeatable real world experimental testing is resource intensive. In addition to extensive logistics necessary to address hardware and networking failures for successful deployment, acquiring physical space usage that complies with legal airspace regulations and safety requirements as well as establishing scientific constraints necessary for rigorous problem isolation is unattainable without extensive collaboration efforts. A VR platform is a feasible representation of many facets of real world usage that addresses safety and regulatory concerns, while enabling scientific rigour for human subject studies with reproducible environmental conditions for every trial. Our open-source VR platform [15]–[17] performs a similar scientific function as field tests, enabling benchmarking of recent [18]–[22] and future human-swarm paradigms experimentally.

A variety of human-swarm algorithms have been proposed and tested. Common swarm command approaches extend one action input to a swarm by relying on leader-follower relationship where remaining robots follow in a formation [22]–[27] or applying the same command input to multiple robots [11], [12], [19], [27]–[30], sometimes using mixed initiative control [31]. To address limitations of one-to-many approaches and increase granularity of user specifications, some allow the operator to select robot(s) to which a command will apply [11], [12], [22], [26]. Other approaches bypass this question by incorporating fully autonomous exploration strategies that do not require any human input [2], [32], [33]. While fully autonomous exploration may be necessary when the human operator is unable to command the swarm, most swarm control strategies are unable to adapt to a new setting and task goal without a human input. User inputs result in more informed swarm exploratory decisions [10], [11], [34].

In this work, we demonstrate the benefits of a distribution based swarm control paradigm [35] that combines human

[1]Northwestern University, Evanston, IL, USA
[2]Swiss Federal Institute of Technology Lausanne, Lausanne, Switzerland
[3]University of Pennsylvania, Philadelphia, PA, USA

This material is based on work supported by DARPA under grants N660011924024 and HR00112020035, and the NSF under grant CNS 1837515. Millicent Schlafly and Katarina Popovic were additionally supported by National Science Foundation Graduate Research Fellowships.

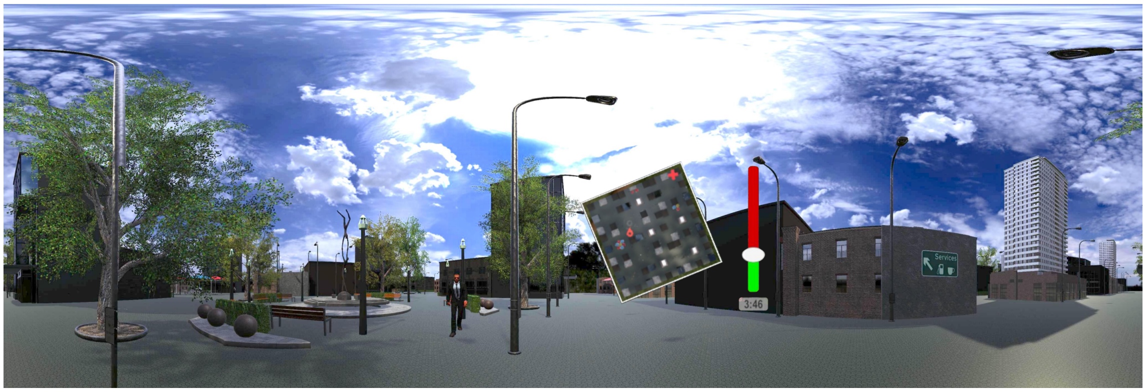

Fig. 1. **Virtual reality environment.** We challenge sixteen participants to complete a timed, treasure-gathering task while being chased by adversaries (wearing black suits). Participants are assisted by a swarm of three drones that can alert the participant to the locations of possible adversaries on an aerial minimap. The location of the treasure is indicated by a red "X" and the person's location by a red pin. Next to the minimap is a life bar and the game time.

input with autonomous exploration for experienced users. Our findings indicate that domain expertise impacts whether people can exploit shared control paradigms, with experts taking advantage of shared control and novices benefiting from fully autonomous assistance. Participants complete a high pressure task of collecting treasures while being chased in a VR environment. Throughout the experiment, three drones provide varying types and levels of aerial coverage assistance. The advantages of our paradigm are assessed using measures for task performance, the amount of interaction with the interface, and perceived difficulty.

II. Experimental System Architecture

A. Hardware & Communication

Our experimental setup requires three computers for the tactile tablet human interface, central computer containing the swarm controllers, and Unity VR environment shown in Figure 3. The tablet and VR system are both run on Windows operating systems with the corresponding specs of Intel core i9-9980HK CPU 2.40GHz and i7-8700 CPU 3.20GHz respectively. The central computer is a Linux machine with Ubuntu 18.04 operating system and an Intel core i9-9980HK CPU 2.40GHz processor using ROS [36] (Robot Operating System, version Melodic). The tactile tablet interface and the central computer communicate over a TCP socket. Communication between the central computer and the Unity game engine happens over a ROSbridge websocket developed by Siemens.

B. Virtual Reality Environment

Two environments are created that differ in the building density, and thus, represent areas of low (high-building density) and high (low-building density) spatial visibility. Both environments are built on a 30-by-30 grid, with a 10-by-10 design block of buildings that is repeated to populate the entire plane. The square grid shape simplifies algorithm implementation. For the low-building density environment, 25% of the buildings are removed and replaced with realistic city-like spacious areas such as parks, outdoor dining and public seating areas. In the high-building density urban environment, inanimate objects such as various benches, trash containers, as well as street signs and greenery are added to the rudimentary block structure for a more realistic urban-like representation.

Participants used an HTC Vive headset and controllers to maneuver in the virtual reality (VR) environment. The VR experience is created using Unity 3D software (Unity3D version 2019.3.0 Alpha 8) with asset behaviour and command controls coded in C# using Visual Studio (2017 Visual Studio version 15.9.17). So that the player can always access the tablet interface on the table in front of them, participants complete trials while seated in a chair that could not swivel, primarily using the controllers to move within the VR world. The walking movement is initiated when a controller's trigger button is held with the participants index finger, and the direction of the forward progress is determined based on player's head orientation, as measured by the VR headset. Users could either gradually change direction of motion by turning their head in a desired direction, or perform 30 degree snap turns by pressing buttons on either side of the controller. When the controller is in the left hand, the left side grip button initiates a 30 degrees counterclockwise rotation and the right side grip button initiates a 30 degrees clockwise rotation; rotation buttons are swapped when the controller is held in the right hand. The controller is held with the participant's non-dominant hand so that the participant's dominant hand is available for interaction with the tablet interface. Both the speed of movement as well as the choice and amount of snap rotation is capped to limit the potential side effect of VR motion sickness.

In addition to their direct first-person view of the virtual world, the minimap shown in Figure 3A is displayed to the user. It shows the overhead view of the environment as well as the locations of the target and player at all times. During control paradigm trials that included swarm assistance, the minimap also displays the locations of the drones and temporarily displays the locations of any detected people in the environment for 3 seconds. In the 30x30 unit environment, the drones could detect people within a 2x2 unit square area, but are not given the capability of predicting the future path of an asset. Furthermore, the drone's velocity is capped at 1 unit per second. To aid in spatial orientation in the environment, the minimap display rotated so that the player's view always corresponded to up in the minimap, aligning the player's icon upward. Next to the minimap, the player's health status (number of lives left) and game time are represented via the slider bar and the trial's time counter shown in Figure 1.

C. Tactile Interface for User Commands

To enable players to orient themselves in the virtual reality and send swarm commands while wearing a headset, participants used a TanvasTouch monitor [37]. The TanvasTouch enables the operator to maintain visual situational awareness, an important human-swarm interaction factor which is desirable in real-world scenarios [38]. Given its capability to render surface haptics by modulating the friction underneath the user's fingertip, we characterized two main features: player's location and environmental boundary. Fine textures, resulting in larger vibration, represented the former while coarse textures, resulting in smaller vibrations, outlined the latter. Their representation was dynamically updated to align with the orientation of the minimap displayed to the player in the virtual environment.

Swarm commands are both initiated and completed, sending a command to respective agent(s), by double-tapping the screen. They are followed by an auditory feedback, confirming successful interaction with the tablet as participant's view is obscured with the virtual reality headset. The command input differs based on the control method with users tracing a desired trajectory (waypoint control) for each drone or shading the region of exploratory interest (ergodic control) where transformed coordinates are represented as a distribution and communicated to the all agents. Waypoint control requires additional input (i.e. intermediate taps) to specify which drone will receive the command, with number of taps corresponding to the agent ID (ID = 1, 2, ..., N where N is the total number of drones in the swarm).

III. Control Paradigms

We compared total of four different control paradigms against a baseline of no robot assistance, described in the following subsections.

A. Ergodic Coverage Control

Ergodic coverage control is a distribution based approach to control multiple robotic agents. Ergodicity is a concept for converting spatially distributed task information into temporally driven motion. It determines the amount of time a robotic agent spends in any particular area of the workspace by generating trajectories that minimize the ergodic metric — a metric that is used to compare the temporal statistics of the robots' response to a desired spatial distribution.

The decentralized implementation of ergodic control was developed in [35] and implemented on a swarm of three rovers for the DARPA OFFSET FX-3 Challenge [39]. While each drone is responsible for full area coverage, it also simultaneously communicates its past and future exploration trajectories to the rest of the swarm — this allows for local coverage prioritization while ensuring task completion regardless of number of active agents. Collective inter-swarm communication is not necessary but beneficial. Communication enables optimal energy expenditure because task space coverage is distributed across the entire swarm. Results from a field test implementation corroborate a persistent and responsive method of swarm command regardless of agent availability [14].

*1) **User**:* In this mode, ergodic assistance from the swarm is generated based on coverage needs specified by the user alone. The tablet interface transmits a set of desired points on the workspace communicated by the user for the swarm to prioritize. The spatial distribution $\phi_u(x)$[1] is generated by assigning the highest priority value of 1 at each of those points in a discretized workspace and a value of 0.005 at every other point to generate minimal coverage over the rest of the workspace.

*2) **Autonomous**:* When specifying autonomous information the aim is to (1) maintain coverage around the operator by covering blind spots and (2) reallocate priority to a given region when task-relevant information is discovered[2].

Given an operator's location, an internal visual coverage model generates a domain $\mathcal{O}$ that represents the operator's blind spots within a local radius. Respective distribution $\phi_o(x)$ decreases in value along a constant slope as one gets further from the operator such that the $\sum_x \phi_o(x) = 1$.

When the locations of task-relevant items of interest are discovered, in this case location of pedestrians within the environment, they are communicated to each agent in the network. An evolving dictionary of all the task-relevant information is constructed $\mathcal{D} = \{(p_i, w_i, \Sigma_i)\}_{i=0}^N$ that consists of a constantly updated list of tuples that represent the locations p of task-relevant items of interest, an importance weight w, and variance Σ_i.

The autonomous information $\phi_a(x)$[1] is generated by parameterizing the distribution as a multimodel sum of Gaussians and combining it with the visual coverage model:

$$\phi_a(x) = 2\phi_o(x) + \frac{1}{\eta}\sum_{\mathcal{D}} \Psi(p_i, w_i)(x) \tag{1}$$

where η is a normalization factor, and

$$\Psi(p_i, w_i)(x) = w_i \exp\left(-\frac{1}{2}\|x - p_i\|^2_{\Sigma_i^{-1}}\right). \tag{2}$$

for all x in which $\|x - p_i\| < \Pi$. The parameter Π is the width of the region of attraction that can be tuned based on the size of the task space and the desired granularity.

This representation generates high importance regions over the operator and any pedestrian (i.e. objects of interest) when discovered by the autonomy. Because these items of interests may move over the course of the task, the list of items of

[1] The resulting distribution is normalized such that the $\sum_x \phi(x) = 1$ and represented using 5 Fourier coefficients in each exploratory dimension. Workspace coordinates are transformed and scaled to a bounding box of size $[0, 1]^2$ for numerical stability.

[2] Prior knowledge of the environment is assumed.

interest are constantly updating, and after a period of time (15 seconds), the item is removed from the list until located again.

*3) **Shared**:* To represent shared knowledge, a target distribution $\phi_s(x)^1$ is defined as a linear combination of the two spatial distributions (the user distribution $\phi_u(x)$ and the autonomous distribution $\phi_a(x)$):

$$\phi_s(x) = w_u \phi_u(x) + w_a \phi_a(x),$$

where w_u, w_a represent the weights of the individual components of the shared control, user commands and autonomous specification respectively. Weights $w_u = w_a = 1$ assign equal contribution to the shared distribution.

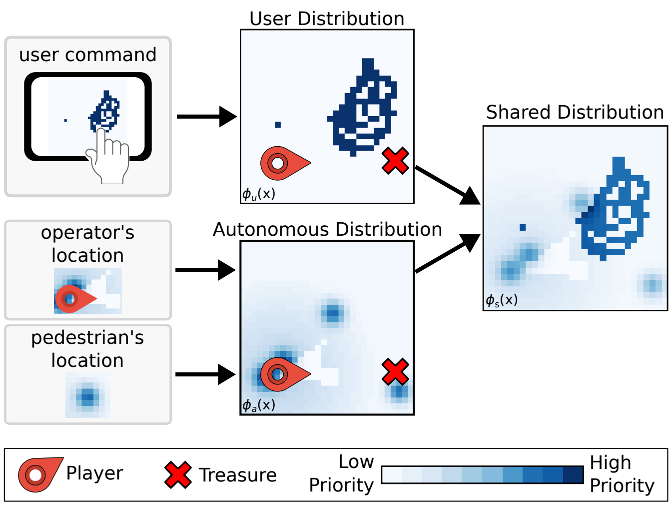

Fig. 2. **Coverage Control Paradigms** During shared coverage control, each robot linearly combines distributions provided separately by the user and the robots. During user and fully autonomous coverage control, the robots only rely on the distribution provided by the user and the robots, respectively. In autonomous trials, the robot identifies the operator's blind spots and prioritizes regions around detected pedestrians, without knowing treasure location.

B. Waypoint

For the waypoint control method, users specify desired trajectories for each individual drone in the swarm. The tablet interface transmits this set of waypoints to the workspace for each corresponding drone to follow. Prior to beginning the task and before learning the starting location of the target and themselves in the virtual world, users are allowed to specify initial paths for individual drones. This initial input accounts for uniform distribution area coverage, a starting point from which trials with ergodic swarm specification begin. The user can update the desired paths for the individual drones at any point during the trial.

IV. Methodology

A. Participants

Healthy adults between ages 18 and 32 were recruited at Northwestern University (NU). Participants with poor visual acuity without contacts were excluded from this study due to difficulty in fitting the VR headset over the glasses. Prior video game experience over their lifetime, as determined by a pre-study questionnaire, divided participants group into novices, with less than 1000 hours of video games, and experts. Eight novices completed our experiment. For comparable evaluation, we randomly selected eight of the twenty-four experienced subjects that participated. In total, we present experimental data from sixteen participants in our paper.

B. Experimental Protocol

At the beginning of the experiment, each participant signed a consent form approved by the Northwestern University's Institutional Review Board following an explanation of the study. All participants then completed a training session that lasted at least one hour, composed of a tutorial series that familiarized them with the different parts of the experimental setup and interface. The experiment employs a within-subject design with five randomized control conditions under two varying building densities. Thus, participants performed each of the 10, 5-minute experiment rounds in a randomized order. After completion of each trial round they were asked to rate perceived difficulty of the control paradigm tested on a scale of 1 to 10. Finally, participants would be asked to fill a post-experiment survey to rank their interface preferences.

C. Experimental Task

The participants played a virtual reality game to (1) collect as many treasures as possible and (2) avoid getting caught by adversaries, resulting in loss of one life out of five. The total score for each five-minute trial is summation of one point per collected treasure and three points per leftover life. Treasure location is marked by a red "X" on the minimap. Reaching this position is equivalent to collecting a treasure. This initiates spawning of new treasure at the random location, both in the environment and on the minimap, with set distance requirement between the consecutive placements. Although our experimental task emulates real-world stressors, a game-like description of the task simplified training.

In addition to the player, there are six virtual humanoid characters, three pedestrians and adversarial agents each, with predefined patrol paths. While pedestrians are placed on a continuous trajectory loop for the game duration, adversaries have a capability to switch into "chasing mode" if the player enters their field of view. This behavior is active until either an adversary (1) "loses" sight of a player or (2)"catches" a player, taking their life away. Consequently, an adversary will either return to their route's nearest point or spawn at a random location along the route, with set separation away and unable to "see" the player.

The swarm of three drones assists the player by alerting them of people in the environment. While our scale invariant control paradigm is applicable to swarms of any size, the experimental conditions such as available exploration area and limited scalability of waypoint control constrained the number of drones. Under assumption of perfect object recognition capability, drones can unmistakably locate humans. This information is shared with a player in the minimap display. However, drones cannot perform classification — the player needs to incorporate this knowledge and reason about

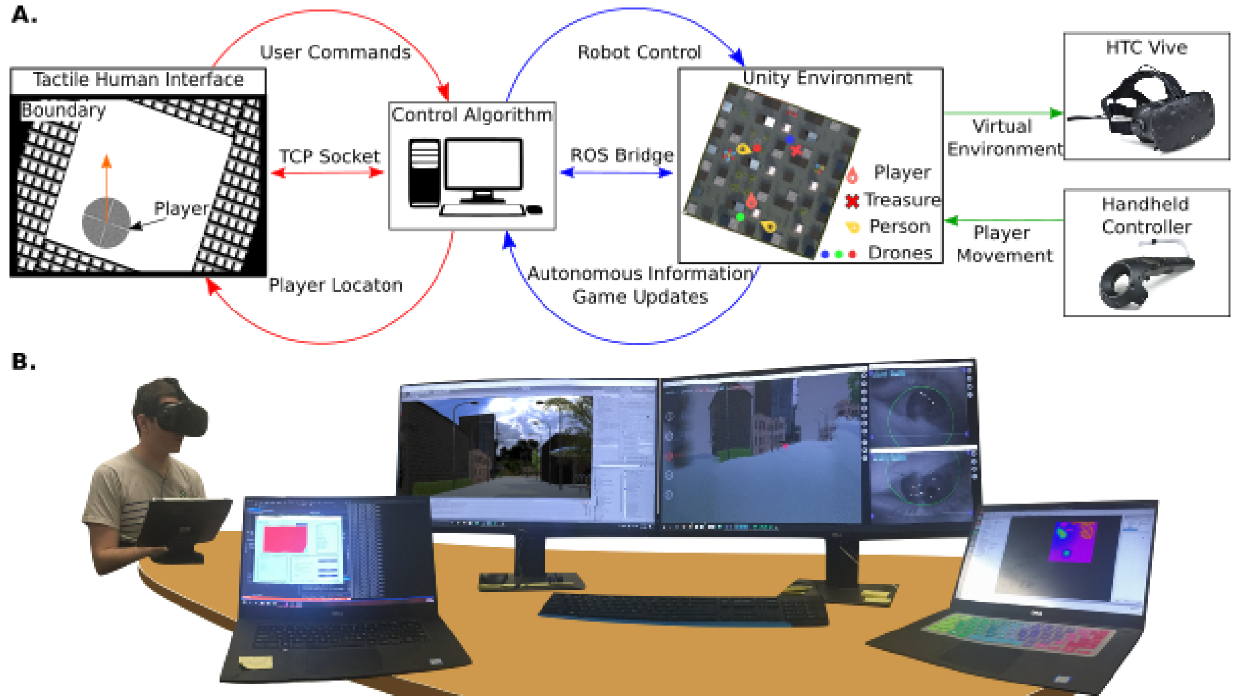

Fig. 3. **Experimental System.** Three computers for the tactile human interface, swarm control algorithms, and virtual reality environment run our experimental software. Intercommunication occurs over websockets indicated by colored arrows in (**A**). The minimap in the Unity environment and TanvasTouch world representation rotate together — forward translates to up on both displays. Participants navigated in the environment with the HTC Vive and one handheld controller. They were seated in a chair during the experiment to avoid motion sickness and ensure access to the tactile interface as shown in (**B**).

possibility of encountering a pedestrians vs. adversaries as they navigate en route to the treasure.

D. Statistical Analysis

For the number of interface interactions, repeated measures ANOVAs with within-participant factors for control paradigm and building density and a between-participant factor for expertise (experienced vs. novice) is performed in R ($\alpha = 0.05$). Assumptions are tested using Shapiro-Wilk test for normality and Mauchly's sphericity test. To help determine which control paradigm is different from the rest, post-hoc, pairwise, and two-way t-tests with a Bonferroni correction for multiple comparisons is performed.

Perceived difficulty rating survey was collected from questionnaire administered at the end of the experiment. Given that data is not normally distributed according to the Shapiro-Wilk test for normality and following standard practices in human-robot interaction studies [40], we use the non-parametric rank test to test for statistical significance. The alternatives to an ANOVA and a t-test used are respectively the Friedman Test with blocks for "participant" and the Wilcoxon Signed Rank Sum Test.

The final score measure had an implicit cap on performance due to the authors setting the initial number of lives and the player's speed, thereby restricting the number of targets that could be obtained. Therefore, we find that the performance data is not normally distributed according to the Shapiro-Wilk test for normality. Alternatively, we fit the data to a generalized linear mixed-effects model (GLM) using the glmer function in R, with the experimental factors as predictors. We chose to parameterize the data using a Poisson distribution because a participant's final score is comprised of the number of times particular events happen during a particular trial (e.g. a life is lost or a target is acquired). After fitting GLMs to our data, we use Wald chi-square tests to evaluate for statistical significance; similar to an ANOVA, the Wald chi-square test evaluates whether a given factor explains some of the variation in an outcome measure. Post-hoc Tukey tests for multiple comparisons are performed to look for significant differences between different control paradigm pairs.

V. Results

As part of the demographic intake form, participants estimated number of hours playing video games over their lifetime. Data demonstrates a natural split in expertise around 1000 hours with no participants having between 580 and 1000 hours of video game experience. For the purpose of analysis, the participants are split into two groups: 8 experienced participants with $\geq$ 1000 hours of video game experience and 8 novice participants with $<$ 1000 hours of video game experience. In particular, three different measures of performance are examined across novice and experienced participants: (1) overall game performance, (2) number of interface interactions, and (3) perceived difficulty rating.

A. Overall performance

The primary metric for evaluating participants' overall performance is the final game score for each experimental condition. Participants were asked to maximize final game score — a predefined summation of the number of lives

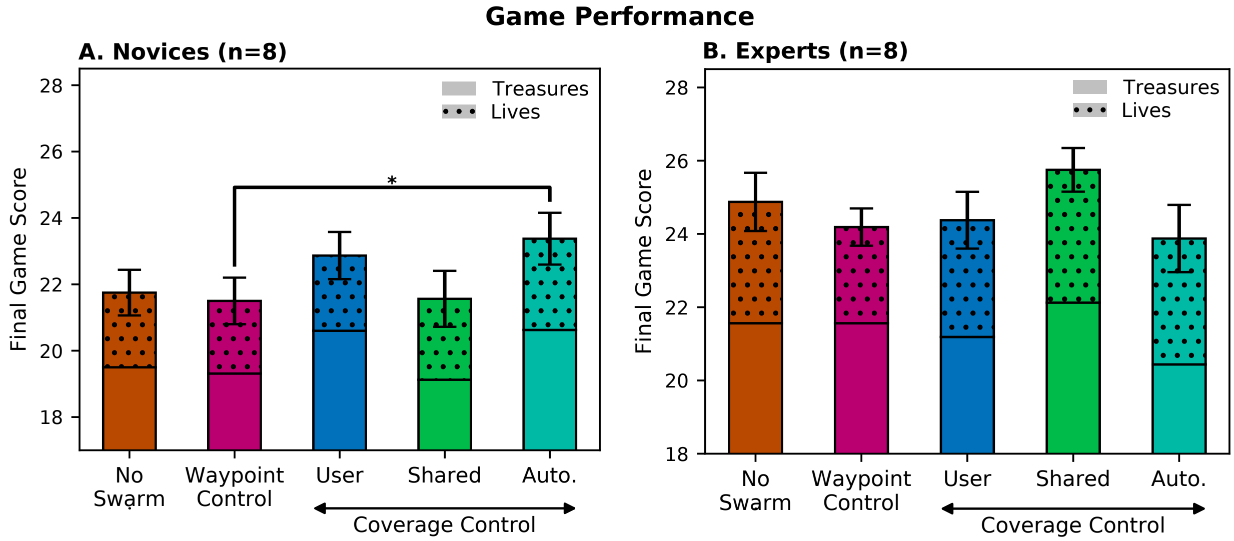

Fig. 4. **Game performance results.** Final game score, a formula provided to participants prior to data collection, is a summation of the number of treasures collected and three times the number of leftover lives. The novices perform better when more assistance is provided (A) while shared coverage control modestly improves game performance in experts (B). The asterisk indicates statistical significance ($^*p < 0.05$) from pairwise t-tests with a Tukey correction.

remaining and treasures ($score = 3N_{lives} + N_{treasures}$) — which was explicitly provided to them.

The statistical analyses reveal that expertise is a statistically significant factor ($\chi^2(4) = 7.00$, $p = 8.12\text{e}{-3}$) with experts scoring higher than novices. There is an interaction effect between control paradigm and expertise ($\chi^2(4) = 8.82$, $p = 1.73\text{e}{-2}$) indicating that control paradigm affects experienced and novice participants differently. Therefore, we separate experienced and novice participants.

Novice Participants. Novices perform worse using shared coverage control than either user coverage control or autonomous coverage control but without significant difference (Figure 4A). However, they scored significantly better using autonomous coverage control than waypoint control ($p = 4.15\text{e}{-2}$). Control paradigm does not significantly affect game performance ($\chi^2(4) = 7.24$, $p = 1.24\text{e}{-1}$).

Experienced Participants. Experienced Participants have modest performance improvement using shared coverage control (Figure 4B). Control paradigm has a marginally significant effect on game performance ($\chi^2(4) = 8.59$, $p = 7.19\text{e}{-2}$).

B. Number of Interface Interactions

Control paradigm has a statistically significant effect on the number of commands provided by participants during the five minute trial ($F(2, 26) = 15.59, p = 3.55\text{e}{-5}$). The post-hoc pairwise t-test shows the greatest number of commands are given during control paradigms in which the drones solely rely on information from the user (Figure 5), with shared coverage control having significantly lower input count compared to both user coverage control ($p = 1.59\text{e}{-2}$) and waypoint control ($p = 1.45\text{e}{-6}$). The difference between the number of inputs during waypoint control and user coverage control trials is marginally significant ($p = 5.13\text{e}{-2}$) with participants providing more inputs during the waypoint control.

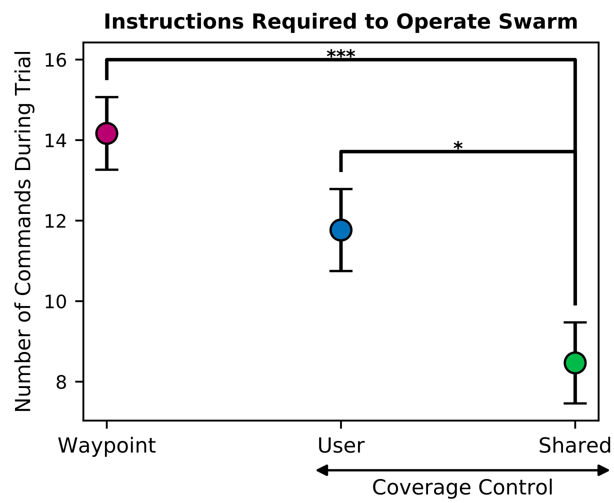

Fig. 5. **Interface interactions results.** Average number of commands given by users (n=16) during a five minute trial. The asterisks indicate statistical significance with $^*p < 0.05$ and $^{***}p < 0.01$.

Since there is no significant interaction effect between expertise and control paradigm ($F(2, 26) = 0.206, p = 8.52\text{e}{-1}$), we present data for all participants combined. Experienced participants provide marginally more commands than novices ($F(1, 13) = 3.94, p = 6.84\text{e}{-2}$). The waypoint control paradigm trials include an additional three commands, one given to each drone, before the start of the trial. This is not necessary for the user and shared coverage control trials because the drones began each trial by following a uniform distribution.

C. Perceived Difficulty Rating

After completion of each experimental trial, participants independently rated the trial's difficulty. Trials were numerically ordered as "Trial 1," "Trial 2,"..."Trial 10" and scored on a scale from 1 to 10 with 1 indicating the trial is difficult and 10 indicating the trial is easy. Given that perceived difficulty rating trend holds for both novice and experienced participants, the results are combined. While the coverage trials are overall perceived easier to use (Figure 6), only the shared coverage control is significantly easier ($p = 1.59\text{e}{-2}$) compared to the no swarm experimental condition based on the Wilcoxon Signed Rank Sum Test. Furthermore, the Friedman test showed that control paradigm is statistically significant ($\chi^2(4) = 9.84$, $p = 4.31\text{e}{-2}$) for the high building density and marginally significant ($\chi^2(4) = 9.15$, $p = 5.73\text{e}{-2}$) for the low building density environment.

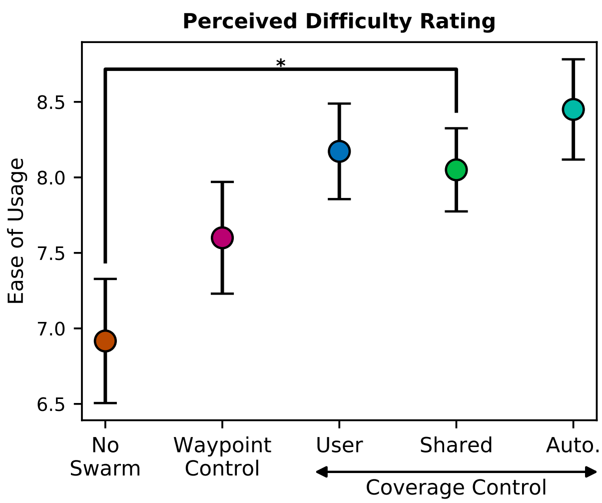

Fig. 6. **Participants perceived coverage control trials as easier.** Participants rated the trial's difficulty on a scale $1 - 10$, with increase in score corresponding to decrease in difficulty. Error bars indicate standard error. Asterisks indicate statistical significance with $^*p \leq 0.05$.

Our post-experiment survey results on control methods indicates that participants overwhelmingly (93.3%) preferred specifying control commands on the table via area shading — an essential component of the user and shared ergodic control paradigms. Only one novice (6.7%) preferred task ergodic trial instances that required no control input at all.

VI. Discussion

A. Benefits of Shared Coverage Control

Directing the swarm's response by distribution specification in ergodic coverage control paradigms enables scalability – a single user input is sufficient to generate trajectories for a swarm of arbitrary size [41]. In contrast, waypoint control commands grow proportional to the swarm size. These differences are reflected in the interaction results (Figure 5) with further implication for quality of commands and higher density of information conveyed through coverage control. Shared coverage control further reduces the number of instructions necessary to operate the swarm compared to user coverage control, indicating participants are relying on the autonomy.

Regardless of whether the user is providing swarm commands at a given moment, the autonomy side of the shared control is persistently updating the swarm's exploration goals in response to evolving task conditions. This characteristic of ergodic coverage, as demonstrated in previous field tests [14], allows the operator to fully allocate their resources to a more pressing task at hand (e.g., running away from an adversary) without hindering swarm exploration. Thus, the swarm continuously provides the operator with information about the environment based on aerial coverage.

Continuous support is reflected in interface interaction results (Figure 5) showing that participants provide fewer swarm commands using the shared control paradigm. By relying on the autonomy, experts exploit the incorporation of autonomous knowledge in shared coverage control to supplement swarm commands. Thus, participants can shift their focus from operating the interface to more strategically important aspects of the game requiring critical thinking (e.g., obtaining treasures), consequently affecting team performance.

A final benefit of shared coverage control is that it enables the human and the autonomy to both contribute to the robots' exploration goals. Participants often choose to specify the desired regions of exploration as areas between themselves and their goal (i.e., the treasure). The autonomy does not know the participant's goal nor does it have any information about the participant's future path but can quickly generate a distribution that prioritizes locations close to the operator and focuses on people detection. The target distribution during shared coverage control leverages both the human and autonomy's knowledge about where to find relevant information, possibly explaining improved game performance in experts.

B. Perspective on Novice Participants

Regardless of their exposure to video games, all participants received the same training and exposure to the experimental setup. However, many video games require players to quickly reason about the uncertain information and autonomous character behaviour under pressure. Prior experience may be relevant to the experimental task, translating to a better understanding of how to maneuver in the game-like environment. It is not clear how much novices' performance using shared coverage control would improve with additional practice as well as deeper understanding of autonomy's role.

VII. Conclusions and Future Work

This work evaluates multiple approaches to generating swarm control that incorporate user input and task specification against the baseline — no task assistance. A stressful, dynamically-changing and time-sensitive experimental task was designed to capture characteristics of real-world situations an operator may face. The results of the human subject study demonstrate that the importance of interaction type in a human-robot systems is skill dependant. In particular, modest performance benefits arise for expert participants from expressing task specific intent via distributions in a shared control setting that provides persistent autonomous assistance towards goal completion. The novice participants with less video games experience perform better in a fully autonomous control setting. For both expert and novice participants, shared coverage control requires significantly fewer swarm instructions from the operator than waypoint control, and is preferred method for command control specification for nearly all.

Further research questions in human-swarm teaming are apparent and now attainable. While the initial results are encouraging, a greater number of participants would allow us to understand effects of control paradigms with more certainty. Expanding both training and testing period over multiple session could reveal effects of autonomy levels, particularly for novice users with little to no prior task related experience. Lastly, evaluating how cognitively expensive different swarm control paradigms are on the user's mental state could be used to quantify their operational cost.

References

[1] M. Bernard, K. Kondak, I. Maza, and A. Ollero, "Autonomous transportation and deployment with aerial robots for search and rescue missions," *Journal of Field Robotics*, vol. 28, no. 6, pp. 914–931, 2011.

[2] Y. Tian, K. Liu, K. Ok, L. Tran, D. Allen, N. Roy, and J. P. How, "Search and rescue under the forest canopy using multiple UAVs," *The International Journal of Robotics Research*, vol. 39, no. 10-11, 2020.

[3] Y. Liu and G. Nejat, "Multirobot cooperative learning for semiautonomous control in urban search and rescue applications," *Journal of Field Robotics*, vol. 33, no. 4, pp. 512–536, 2016.

[4] O. Bjurling, R. Granlund, J. Alfredson, M. Arvola, and T. Ziemke, "Drone swarms in forest firefighting: A local development case study of multi-level human-swarm interaction," *Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society*, 2020.

[5] A. M. Naghsh, J. Gancet, A. Tanoto, and C. Roast, "Analysis and design of human-robot swarm interaction in firefighting," *International Symposium on Robot and Human Interactive Communication*, 2008.

[6] J. J. Roldán-Gómez, E. González-Gironda, and A. Barrientos, "A survey on robotic technologies for forest firefighting: Applying drone swarms to improve firefighters' efficiency and safety," *Applied Sciences*, vol. 11, no. 1, p. 363, 2021.

[7] O. Fernandes, R. Murphy, D. Merrick, J. Adams, L. Hart, and J. Broder, "Quantitative data analysis: Small unmanned aerial systems at Hurricane Michael," *International Symposium on Safety, Security, and Rescue Robotics*, pp. 116–117, 2019.

[8] J. Qi, D. Song, H. Shang, N. Wang, C. Hua, C. Wu, X. Qi, and J. Han, "Search and rescue rotary-wing UAV and its application to the lushan ms 7.0 earthquake," *Journal of Field Robotics*, vol. 33, no. 3, 2016.

[9] K. Nagatani, S. Kiribayashi, Y. Okada, K. Otake, K. Yoshida, S. Tadokoro, T. Nishimura, T. Yoshida, E. Koyanagi, M. Fukushima *et al.*, "Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots," *Journal of Field Robotics*, vol. 30, no. 1, pp. 44–63, 2013.

[10] J. R. Cody, K. A. Roundtree, and J. A. Adams, "Human-collective collaborative target selection," *ACM Transactions on Human-Robot Interaction*, vol. 10, no. 2, pp. 1–29, 2021.

[11] C. C. Ashcraft, M. A. Goodrich, and J. W. Crandall, "Moderating operator influence in human-swarm systems," *International Conference on Systems, Man and Cybernetics*, pp. 4275–4282, 2019.

[12] D. S. Brown, M. A. Goodrich, S.-Y. Jung, and S. Kerman, "Two invariants of human-swarm interaction," *Journal of Human-Robot Interaction*, vol. 5, no. 1, pp. 1–31, 2016.

[13] J. Zhou, H. Zhu, M. Kim, and M. L. Cummings, "The impact of different levels of autonomy and training on operators' drone control strategies," *ACM Transactions on Human-Robot Interaction*, vol. 8, no. 4, 2019.

[14] A. Prabhakar, I. Abraham, A. Taylor, M. Schlafly, K. Popovic, G. Diniz, B. Teich, B. Simidchieva, S. Clark, and T. Murphey, "Ergodic specifications for flexible swarm control: From user commands to persistent adaptation," *Robotics: Science and Systems*, 2020.

[15] K. Popovic and M. Schlafly, "Open-source code for unity virtual reality environments," Feb. 2023. [Online]. Available: https://bit.ly/3ESCnDN

[16] M. Schlafly, "Open-source code for tactile human interface," https://github.com/mschlafly/darpahaptic, Nov. 2021.

[17] M. Schlafly and A. Prabhakar, "Open-source code for swarm control algorithm," Nov. 2021. [Online]. Available: https://github.com/mschlafly/VR_exp_ROS

[18] Y. Lee, W. Do, H. Yoon, J. Heo, W. Lee, and D. Lee, "Visual-inertial hand motion tracking with robustness against occlusion, interference, and contact," *Science Robotics*, vol. 6, no. 58, p. eabe1315, 2021.

[19] D. Zhou, Z. Wang, and M. Schwager, "Agile coordination and assistive collision avoidance for quadrotor swarms using virtual structures," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 916–923, 2018.

[20] M. Santos, G. Notomista, S. Mayya, and M. Egerstedt, "Interactive multi-robot painting through colored motion trails," *Frontiers in Robotics and AI*, vol. 7, p. 143, 2020.

[21] S. Rajappa, H. Bülthoff, and P. Stegagno, "Design and implementation of a novel architecture for physical human-UAV interaction," *The International Journal of Robotics Research*, vol. 36, no. 5-7, 2017.

[22] J. Cacace, A. Finzi, V. Lippiello, M. Furci, N. Mimmo, and L. Marconi, "A control architecture for multiple drones operated via multimodal interaction in search & rescue mission," *International Symposium on Safety, Security, and Rescue Robotics*, pp. 233–239, 2016.

[23] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, 2018.

[24] D. Qian and Y. Xi, "Leader–follower formation maneuvers for multi-robot systems via derivative and integral terminal sliding mode," *Applied Sciences*, vol. 8, no. 7, p. 1045, 2018.

[25] S. Musić and S. Hirche, "Control sharing in human-robot team interaction," *Annual Reviews in Control*, vol. 44, pp. 342–354, 2017.

[26] S. Clark, K. Usbeck, D. Diller, and R. E. Schantz, "CCAST: A framework and practical deployment of heterogeneous unmanned system swarms," *GetMobile: Mobile Computing and Communications*, vol. 24, no. 4, pp. 17–26, 2021.

[27] A. Franchi, C. Secchi, M. Ryll, H. H. Bulthoff, and P. R. Giordano, "Shared control: Balancing autonomy and human assistance with a group of quadrotor UAVs," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 57–68, 2012.

[28] C. Vasile, A. Pavel, and C. Buiu, "Integrating human swarm interaction in a distributed robotic control system," *International Conference on Automation Science and Engineering*, pp. 743–748, 2011.

[29] J. W. Crandall, N. Anderson, C. Ashcraft, J. Grosh, J. Henderson, J. McClellan, A. Neupane, and M. A. Goodrich, "Human-swarm interaction as shared control: Achieving flexible fault-tolerant systems," 2017.

[30] A. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis, "Human interaction with robot swarms: A survey," *Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 9–26, 2015.

[31] C. Nam, P. Walker, H. Li, M. Lewis, and K. Sycara, "Models of trust in human control of swarms with varied levels of autonomy," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 3, 2019.

[32] K. McGuire, C. De Wagter, K. Tuyls, H. Kappen, and G. C. de Croon, "Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment," *Science Robotics*, vol. 4, no. 35, 2019.

[33] N. Ayanian, "Dart: Diversity-enhanced autonomy in robot teams," *The International Journal of Robotics Research*, vol. 38, no. 12-13, 2019.

[34] M. Q. Azhar and E. I. Sklar, "A study measuring the impact of shared decision making in a human-robot team," *The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 461–482, 2017.

[35] I. Abraham and T. D. Murphey, "Decentralized ergodic control: distribution-driven sensing and exploration for multiagent systems," *Robotics and Automation Letters*, vol. 3, no. 4, pp. 2987–2994, 2018.

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

[37] "Tanvas - Surface haptic technology and products," https://tanvas.co/.

[38] A. Hussein and H. Abbass, "Mixed initiative systems for human-swarm interaction: Opportunities and challenges," in *2018 2nd Annual Systems Modelling Conference (SMC)*. IEEE, 2018, pp. 1–8.

[39] DARPA, "Offset swarm systems integrators demonstrate tactics to conduct urban raid." [Online]. Available: https://www.darpa.mil/news-events/2020-01-27

[40] M. L. Schrum, M. Johnson, M. Ghuy, and M. C. Gombolay, "Four years in review: Statistical practices of likert scales in human-robot interaction studies," pp. 43–52, 2020.

[41] J. Meyer, A. Prabhakar, A. Pinosky, I. Abraham, A. Taylor, M. Schlafly, K. Popovic, G. Diniz, B. Teich, B. Simidchieva, S. Clark, and T. Murphey, "Scale-invariant specifications for human-swarm systems," *Field Robotics*, vol. 3, pp. 368–391, 2023.