

# Supplemental Materials: Learning Visualization Policies of Augmented Reality for Human-Robot Collaboration

**Kishan Chandan**  
Computer Science Department  
SUNY Binghamton United States  
kchanda2@binghamton.edu

**Jack Albertson**  
Computer Science Department  
SUNY Binghamton United States  
jalbert5@binghamton.edu

**Shiqi Zhang**  
Computer Science Department  
SUNY Binghamton United States  
zhangs@binghamton.edu

In the supplementary materials, we provide additional details on the implementation of the *Visualizations for Augmented Reality using Imitation Learning* (VARIL) framework. The last section contains additional details on the baseline implementations, and finally we describe the subjective analysis results.

## 1 System Instantiation

In this section, we describe our implementation of the VARIL framework in a simulated warehouse environment (Figure 1). A key part of VARIL is an AR agent that learns to help a human worker visualize the status of a multi-robot team towards effective human-robot collaboration.

### 1.1 Warehouse Environment

We developed a platform for simulating warehouse environments using Unity [1], where human-robot teammates work together to complete delivery tasks. Figure 1 shows the simulated warehouse environment. The warehouse environment comprises of the following components:

- A warehouse room
- Shelves with boxes
- A virtual human
- Drop stations
- Turtlebot robots

The robots and human work together in the shared warehouse room to complete collaborative delivery tasks. The robots move to different shelves to pick up boxes based on their tasks. After picking the boxes, the robots move to the drop stations to deliver the objects and wait there until they get help from the human to unload the box.

We have created a virtual human in Unity that can be teleoperated by a real human to collaborate with the robots. While the robots are waiting at the drop station, the virtual human can be moved to a drop station to help the robots unload the objects. For simplicity, when the virtual human is in a radius of one meter from the robot waiting at the drop station, the robot automatically drops the box.

Additionally, our warehouse environment can be dynamically scaled by changing the parameters based on the desired complexity of the environment. Figure 1 shows an example of the two different scales of the warehouse environments. With the ability to choose the size of the environment, we can vary the complexity of the human-robot tasks.

**Multi-robot Task Allocator ( $\mathcal{T}^a$ ):** We designed a multi-robot task allocator that assigns tasks to the team of robots at runtime. To allocate the tasks,  $\mathcal{T}^a$  uses a task pool ( $\mathcal{P}^t$ ) that contains all the tasks available for the robots. We use a random seed to initialize a pseudo-random number generator that is used to select the index of the task to be allocated to a particular robot.

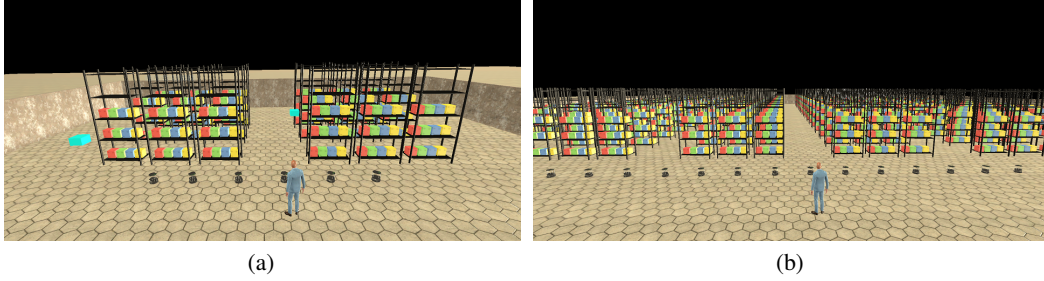


Figure 1: (a) A miniature version of the warehouse environment with 18 shelves and 6 Turtlebot robots. (b) A larger version of the warehouse environment with 225 shelves and 15 Turtlebot robots.

**Motion Planner ( $\mathcal{P}^m$ ):** We use Unity NavMesh (short for Navigation Mesh) [2] for robot navigation and obstacle avoidance. We first generate a NavMesh, which is a data structure to describe the navigable surfaces of the environment, and allows the robot to find a path from one navigable location to another in the environment. Every robot is considered as a NavMesh agent in Unity, and every NavMesh agent can avoid each other while moving towards their goal. All the agents in Unity reason about the environment using the NavMesh, and they know how to avoid each other as well as moving obstacles.

## 1.2 AR Agent for the AR Interface

In our instantiation of the VARIL system, we used four different types of visualizations in the AR interface to enable the human to track the robot status in the warehouse environment. The different visualizations are as follows:

- **Trajectory** shows the motion intentions of the robot. Additionally, the width of the trajectory increases with the distance of the goal from that point on the trajectory. Due to the presence of multiple robots in the environment, the trajectories of different robots are overlaid using different colors so that the human can distinctly identify different robot plans.
- **Solid Avatar** shows the live location of the robot. As the robot moves towards the goal location, the position of the solid avatar is updated accordingly.
- **Transparent Robot Avatar** shows the direction of the motion of the robot by moving the avatar constantly over the trajectory.
- **Floating Balloon** enables the human to locate the drop station from a long distance.

## 1.3 State-Action Space of AR Agent

We learn a visualization strategy for two types of visualizations, one for the visualization of robots, and other for the visualization of the drop station. The state-space of our AR agent for the robot consists of the following:

- humanState: {close, moderate, far}
- robotWaitingTime: {short, medium, long}
- robotTaskState: {picking, dropping}
- nearbyRobots: {few, many}
- robotRemainingTasks: {few, many}
- nearbyRobotVizStatus: {few, many}

The above state representation consists of all the key states that represent the entire human-multi-robot system from each agent’s perspective. The action space of the AR agent depends on the number of different visualizations. In our case for robots, we have three different visualizations, which are:

- liveLocation: {on, off}
- transparentAvatar: {on, off}
- trajectory: {on, off}

where each of them can be turned on or off. The state-space of our AR agent for each drop station consists of the following:

- humanState: {close, moderate, far}
- robotsWaitingTimeDS: {short, medium, long}

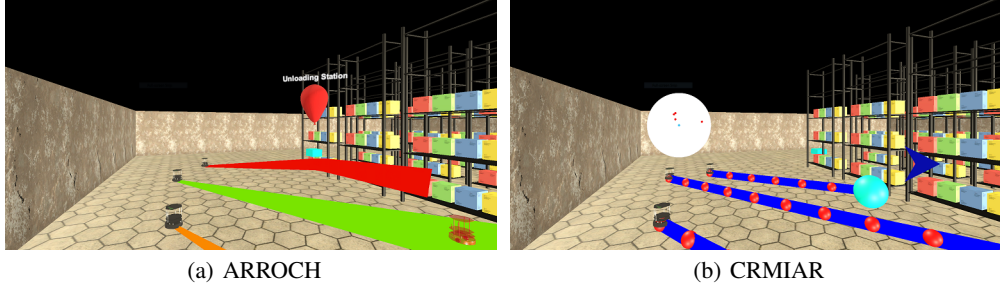


Figure 2: Our implementation of two existing methods from literature, ARROCH [3] and CRMIAR [4], showing their AR visualizations for enabling the humans to track the status of robots.

- $nRobotsAtDropStation$ : {few, many}

The action space for each drop station consists of enabling or disabling the balloon on the drop station, hence it is represented as follows:

- balloon: {on, off}

The above state and action space represents our implementation of the AR agent, and we use the PolicyUp function (Algorithm ??), which is a part of the VARIL framework to learn a visualization policy for human-multi-robot teams.

## 2 Additional Results

### 2.1 Baselines

Figure 2(a) is our implementation of the ARROCH system, which has the following visualizations: the trajectories to show the motion intentions, robot avatar to show the live location, and a transparent robot avatar to show the direction of the motion of the robot. On the other hand, CRMIAR was a single robot visualization system, but we have scaled it to enable tracking the team of robots in our warehouse environment. In Figure 2(b), we show our implementation of the CRMIAR system, where we have combined the NavPoints, and Utilities designs of CRMIAR to provide the following visualizations: the trajectories of robots to show their planned motion intentions, a map to show the relative location of the robots with respect to the human, and arrows to point to the robots that are not in the field of view.

**Questionnaire for Subject Analysis:** The questions listed in the questionnaire included: Q1, *The tasks were easy to understand*; Q2, *The tasks were not mentally demanding, e.g., remembering, deciding, thinking, etc.*; Q3, *I enjoyed using the system and would like to use such a system in the future.*; Q4, *It was easy to keep track of robot status.*; Q5, *The visualizations proved helpful in completing the tasks.*; and Q6, *The visualizations were not distracting.* Note that Q1 is a verification question to evaluate if the participants understood the tasks, and is not directly relevant to the evaluation of our hypotheses. The response choices were: 1 (Strongly disagree), 2 (Somewhat disagree), 3 (Neutral), 4 (Somewhat agree), and 5 (Strongly agree).

Figure 3 (presented in the supplementary materials) shows the average scores from the questionnaires. Results show that VARIL produced higher scores on questions Q3-Q6, where we observed **significant improvements** in all of the four questions with the  $p$ -values  $< 0.05$ . The significant improvements observed from the study support Hypothesis-II on user experience that the visualizations in VARIL provided a better user experience. We did not observe a significant difference in scores for Q2, and one possible reason we thought is the way the question was framed. Since we ask about the demanding nature of tasks, and the re-

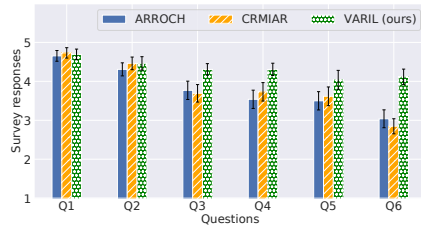


Figure 3: Questionnaire scores

sponse of Q1 confirms that the tasks were easily understood by the participants, the scores make sense. On the other hand, if the question was about the demanding nature of processing the visualization, we think the question would have been useful for the evaluation of different visualization strategies.

## References

- [1] A. Juliani, V.-P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, and D. Lange. Unity: A general platform for intelligent agents, 2020.
- [2] M. Kallmann and M. Kapadia. Navigation meshes and real-time dynamic planning for virtual worlds. In *ACM SIGGRAPH 2014 Courses*, SIGGRAPH '14, New York, NY, USA, 2014. Association for Computing Machinery.
- [3] K. Chandan, V. Kudalkar, X. Li, and S. Zhang. ARROCH: Augmented Reality for Robots Collaborating with a Human. In *2021 International Conference on Robotics and Automation (ICRA)*, 2021.
- [4] M. Walker, H. Hedayati, J. Lee, and D. Szafir. Communicating robot motion intent with augmented reality. In *Proceedings of the International Conference on Human-Robot Interaction*, 2018.