ARROCH: Augmented Reality for Robots Collaborating with a Human

Kishan Chandan, Vidisha Kudalkar, Xiang Li, Shiqi Zhang

***Abstract*— Human-robot collaboration frequently requires extensive communication, e.g., using natural language and gesture. Augmented reality (AR) has provided an alternative way of bridging the communication gap between robots and people. However, most current AR-based human-robot communication methods are unidirectional, focusing on how the human adapts to robot behaviors, and are limited to single-robot domains. In this paper, we develop *AR for Robots Collaborating with a Human* (ARROCH), a novel algorithm and system that supports bidirectional, multi-turn, human-multi-robot communication in indoor multi-room environments. The human can see through obstacles to observe the robots' current states and intentions, and provide feedback, while the robots' behaviors are then adjusted toward human-multi-robot teamwork. Experiments have been conducted with real robots and human participants using collaborative delivery tasks. Results show that ARROCH outperformed a standard non-AR approach in both user experience and teamwork efficiency. In addition, we have developed a novel simulation environment using Unity (for AR and human simulation) and Gazebo (for robot simulation). Results in simulation demonstrate ARROCH's superiority over AR-based baselines in human-robot collaboration.**

I. Introduction

Robots are increasingly present in everyday environments, such as warehouses, hotels, and airports, but human-robot collaboration (**HRC**) is still a challenging problem. As a consequence, for instance, the work zones for robots and people in warehouses are separated [1]; and hotel delivery robots barely communicate with people until the moment of delivery [2]. When people and robots work in shared environments, it is vital that they communicate to collaborate with each other to avoid conflicts, leverage complementary capabilities, and facilitate the smooth accomplishment of tasks. In this paper, we aim to enable people and robot teams to efficiently and accurately communicate their current states and intentions toward collaborative behaviors.

Augmented reality (**AR**) technologies focus on visually overlaying information in an augmented layer over the real environment to make the objects interactive [3]. AR has been applied to human-robot systems, where people can visualize the state of the robot in a visually enhanced form [4]. Most existing research on AR-based human-robot collaboration focuses on the visualization of robot status, and how people leverage the augmented information to adapt to robot behaviors, resulting in unidirectional communication [5], [6]. Another observation is that many of those methods from the literature were limited to human-single-robot, in-proximity scenarios [7], [8]. In this paper, we focus on **human-multi-robot**, **beyond-proximity** settings, where the robots need to collaborate with each other and with people at the same time, in indoor multi-room environments. Our developed algorithm (and system) is called **ARROCH**, short for *AR for robots collaborating with a human*.

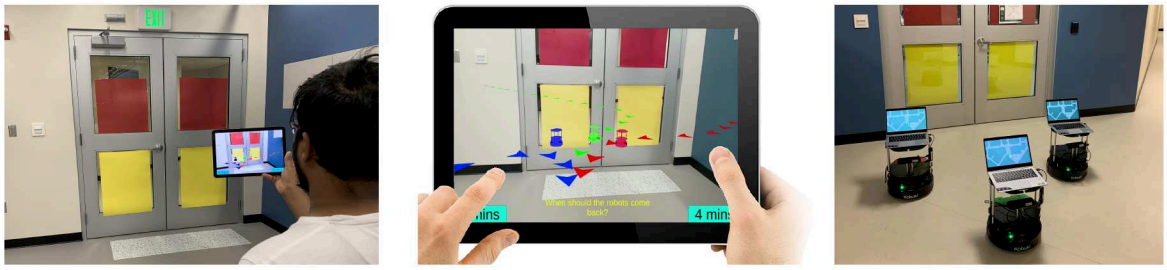
ARROCH supports **bidirectional** human-robot communication. On the one hand, ARROCH enables the human to visualize the robots' current states, e.g., their current locations, as well as their intentions (planned actions), e.g., to enter a room. For instance, a human might see through an opaque door via AR to "X-ray" a mobile robot waiting outside along with its planned motion trajectory. On the other hand, leveraging a novel AR interface, ARROCH allows the human to give feedback to the robots' planned actions, say to temporarily forbid the robots from entering an area. Accordingly, the robots can incorporate such human feedback for replanning, avoiding conflicts, and constructing synergies at runtime. ARROCH supports beyond-proximity communication by visually augmenting the robots' states and intentions, which is particularly useful in indoor multi-room environments.

ARROCH (algorithm and system), as the **first contribution** of this paper, has been evaluated in simulation and using real robots. The robots help people move objects, and the human helps the robots open doors (the robots do not have an arm, and cannot open the doors). Results from the real-world experiment suggest that ARROCH significantly improves the efficiency of human-robot collaboration, compared with a standard non-AR baseline. The simulation platform is constructed using Unity [9] for simulating the AR interface and human behaviors, and using Gazebo [10] for simulating robot behaviors. To the best of our knowledge, this is the first open-source simulation platform for simulating AR-based human-multi-robot behaviors, which is the **second contribution** of this research. In the simulation, ARROCH significantly improved human-robot teamwork efficiency in comparison to baseline methods with different communication mechanisms.

II. Related Work

Human-Robot Communication Modalities: Humans and robots prefer different communication modalities. While humans employ natural language and gestures, the information in digital forms, such as text-based commands, is more friendly to robots. Researchers have developed algorithms to bridge the human-robot communication gap using natural language [11], [12], [13], [14] and vision [15], [16], [17]. Despite those successes, augmented reality (AR) has its unique advantages of providing a communication medium

Chandan, Kudalkar, and Zhang are with SUNY Binghamton.
Email: {kchanda2, vkudalk1, zhangs}@binghamton.edu
Li is with OPPO US Research Center.
Email: xiang.li@oppo.com



(a) Visualization of robots' states and intentions (b) AR interface of ARROCH (c) Robots waiting to enter a room

Fig. 1. The AR interface of ARROCH enables the human to visualize the robots' current states (their current locations in this example) using 3D robot avatars, and their intentions (entering the room) using trajectory markers. ARROCH also enables the human to give feedback to robot plans. In this example, the human can use the interactive buttons in the bottom corners to indicate he could not help open the door in a particular time frame, say "4 minutes."

with potentially less ambiguity and higher bandwidth [18]. AR helps in elevating coordination through communicating spatial information, e.g., through which door a robot is coming into a room and how (i.e., the planned trajectory), when people and robots share a physical environment. Researchers are increasingly paying attention to AR-based human-robot systems [19]. We use an AR interface for human-robot communication, where the human can directly visualize and interact with the robots' current and planned actions.

Projection-based Communication: One way of delivering spatial information related to the local environment, referred to as "projection-based AR," is through projecting the robot's state and motion intention to the humans using visual cues [6], [20], [21]. For instance, researchers used an LED projector attached to the robot to show its planned motion trajectories, allowing the human partner to respond to the robot's plan to avoid possible collisions [22]. While projection-based AR systems facilitate human-robot communication about spatial information, they have the requirement that the human must be in close proximity to the robot. We develop our AR-based framework that inherits the benefits of spatial information from the projection-based systems while alleviating the proximity requirement, and enabling bidirectional communication.

Unidirectional AR-based: More recently, researchers have developed frameworks to help human operators visualize the motion-level intentions of unmanned aerial vehicles (UAVs) using AR [5], and visualize a robot arm's planned actions in the car assembly tasks [7]. One common limitation of those systems is their unidirectional communication, i.e., their methods only convey the robot's intentions to the human but do not support the communication the other way around. In comparison, ARROCH supports bidirectional communication, and has been applied to human-multi-robot, multi-room collaboration domains.

Bidirectional AR-based: Early research on AR-based human-robot interaction (HRI) has enabled a human operator to interactively plan, and optimize robot trajectories [23], [24]. The AR-based systems have also been used to improve the teleoperation of collocated robots [25]. Recently, researchers have studied how AR-based visualizations can help the shared control of robotic systems [26]. Most relevant to this paper is a system that supports a human user to visualize the robot's sensory information, and planned trajectories, while allowing the robot to ask questions through an AR interface [27]. In comparison to their work on single-robot domains, ARROCH supports human-multi-robot collaboration, where both multi-robot and human-robot teamwork is supported. More importantly, our robots are equipped with the task (re)planning capability, which enables the robots to respond to the human feedback by adjusting their task completion strategy leading to collaborative behaviors within human multi-robot teams. The key properties of a sample of existing methods and ARROCH are summarized in Table I.

TABLE I

A SUMMARY OF THE KEY PROPERTIES OF A SAMPLE OF EXISTING AR-BASED HUMAN-ROBOT COMMUNICATION METHODS.

	Beyond proximity	Bidir. comm.	Multiple robots	Task planning
[6], [22], [7]	✗	✗	✗	✗
[20], [21], [26], [27]	✗	✓	✗	✗
[25], [24]	✓	✓	✗	✗
ARROCH (ours)	✓	✓	✓	✓

III. ARROCH: Algorithm and System

In this section, we describe ARROCH (*AR for robots collaborating with a human*), an algorithm and system that utilizes AR technologies to enable bidirectional, multi-turn, beyond-proximity communication to enable collaborative behaviors within human-multi-robot teams. Fig. 1 shows our novel AR interface, which enables the human to visualize the robots' current states, and their intentions (planned actions), while at the same time supporting the human to give feedback to the robots. The robots can then use the feedback to adjust their plans as necessary. Next, we describe the ARROCH algorithm, and then its system implementation.

A. The ARROCH Algorithm

The input of Algorithm 1 includes, $\boldsymbol{\mathcal{S}}$, a set of initial states of N robots, where $s_i \in \boldsymbol{\mathcal{S}}$ is the initial state of the i^{th} robot; $\boldsymbol{\mathcal{G}}$, a set of goal states, where $g_i \in \boldsymbol{\mathcal{G}}$ is a goal state of the i^{th} robot.[1] A multi-robot task planner, $\mathcal{P}^t$, and a motion planner, $\mathcal{P}^m$ are also provided as input.

[1]Strictly speaking, tasks are defined as goal conditions [28]. Here we directly take goal states as input for simplification.

Algorithm 1 ARROCH

Input: $\boldsymbol{\mathcal{S}}$, $\boldsymbol{\mathcal{G}}$, $\mathcal{P}^t$, $\mathcal{P}^m$

1: Initialize empty lists: $\boldsymbol{\omega} \leftarrow \emptyset$; $\boldsymbol{\mathcal{I}} \leftarrow \emptyset$; $\boldsymbol{\mathcal{C}} \leftarrow \emptyset$
2: $\boldsymbol{P} = \mathcal{P}^t(\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{C}})$, where $p \in \boldsymbol{P}$ is a task plan (an action sequence) for one robot, and $|\boldsymbol{P}| = N$ ▷ Task planner $\mathcal{P}^t$ in Section III-C
3: **while** $\sum_{p \in \boldsymbol{P}} |p| > 0$ **do**
4: **for** $i \in [0, 1, \cdots, N\text{-}1]$ **do**
5: **if** p_i is **not** empty **then**
6: $a_i \leftarrow p_i$.front() ▷ Obtain current robot action
7: Obtain i^{th} robot's current configuration: $\boldsymbol{\omega}_i \leftarrow \theta(i)$
8: Update the i^{th} robot's current state s_i using $\boldsymbol{\omega}_i$
9: $\boldsymbol{\mathcal{I}}_i \leftarrow \mathcal{P}^m(\boldsymbol{\omega}_i, a_i)$
10: The i^{th} robot follows $\boldsymbol{\mathcal{I}}_i$ using a controller
11: **end if**
12: **end for**
13: $\lambda \leftarrow \mathcal{V}(\boldsymbol{\omega}, \boldsymbol{\mathcal{I}})$ ▷ Visualizer $\mathcal{V}$ defined in Section III-D
14: Collect feedback, H, from human, where the human gives feedback based on task completion status and λ
15: $\boldsymbol{\mathcal{C}} \leftarrow \mathcal{R}(H)$ ▷ Restrictor $\mathcal{R}$ defined in Section III-B
16: $\boldsymbol{P} \leftarrow \mathcal{P}^t(\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{C}})$ ▷ Update plans
17: **end while**

ARROCH starts by initializing: an empty list of robot configurations (in *configuration space*, or C-Space), $\boldsymbol{\omega}$, where each configuration is in the simple form of $\langle x, y, \theta \rangle$ for our mobile robots, an empty list, $\boldsymbol{\mathcal{I}}$, to store the intended trajectories of the N robots in C-Space ($N = |\boldsymbol{\mathcal{S}}|$), and an empty list to store the activated constraints, $\boldsymbol{\mathcal{C}}$.

In Line 2, ARROCH generates a set of N plans, $\boldsymbol{P}$, using multi-robot task **Planner** $\mathcal{P}^t$. Entering the while-loop (Lines 3-17), ARROCH runs until $\sum_{p \in \boldsymbol{P}} |p| > 0$ is false, meaning that all robots have an empty plan, i.e., all tasks have been completed. In each iteration, ARROCH enters a for-loop for updating the current configuration and the intended trajectories of the robots yet to reach their goal states (Line 4-12). The θ function in Line 7 returns the configuration of the i^{th} robot, which is stored at $\boldsymbol{\omega}_i$. In Line 9, $\mathcal{P}^m$ generates the intended trajectory of the i^{th} robot, $\boldsymbol{\mathcal{I}}_i$, to implement action a_i. After the for-loop, the set of robot configurations and robots' intended trajectories, are passed to **Visualizer** (Section III-D), represented by the $\mathcal{V}$ function, which renders a single frame, λ, on the AR interface (Line 13).

The AR interface of ARROCH allows the human to give feedback on (the AR-based visualization of) the robots' plans. ARROCH obtains the human feedback, H (Line 14), and then passes it on to **Restrictor** (Line 15). Restrictor generates a set of activated constraints in logical form, $\boldsymbol{\mathcal{C}}$, that can be processed by Planner in Line 16 (details in Sections III-B and III-C).

Remarks: Leveraging a novel AR interface, ARROCH enables bidirectional, multi-turn, beyond-proximity communication toward collaborative behaviors within human-multi-robot teams. ARROCH is particularly useful in domains with poor visibility, e.g., multi-room indoor environments. The multi-robot task replanning capability enables the robots to collaborate with each other, while responding to human feedback by adjusting their task completion strategies.

Fig. 2 shows an overview of ARROCH. In the following subsections, we describe how **Restrictor** converts human feedback into symbolic constraints, how **Planner** incorporates the constraints to plan for the robots, and how the robots' current and planned actions are visualized in **Visualizer** – all within our ARROCH system.

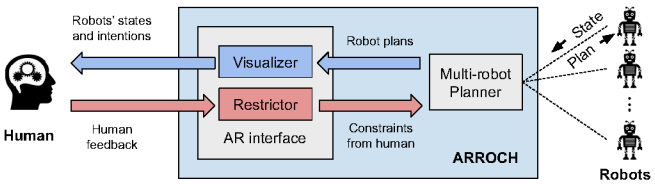


Fig. 2. Key components of our ARROCH algorithm and system: *Visualizer* and *Restrictor* for visualizing robot's intention (for people) and collecting human feedback (for robots) respectively, and *Planner* for computing one action sequence for each robot.

B. Restrictor for Constraint Generation

ARROCH realizes human-multi-robot collaboration by adding human-specified symbolic constraints into the multi-robot task planner. Within the task planning context, a **Constraint** is in form of modal-logic expressions, which should be true for the state-trajectory produced during the execution of a plan [29]. However, naive people have difficulties in directly encoding constraints in logical forms. ARROCH leverages the *graphical user interface* (GUI) of AR to take human feedback, with which Restrictor generates logical constraints. We use Answer Set Programming (ASP) for encoding constraints [30], [31]. A constraint in ASP is technically a STRIPS-style "headless" rule:

```
:- B.
```

where `B` is a conjunction of literals. It conveys the intuition of a constraint: satisfying `B` results in a contradiction. There is rich literature on ASP-based logic programming that provides more technical details [32], [33].

We predefine a constraint library F, which is presented to people through the AR-based GUI. The human can select constraints $H \subseteq F$ to give feedback on the robots' plans (Line 14 in Algorithm 1). The form of human feedback can be flexible and task-dependent. For instance, in tasks that involve navigation, the human can forbid the robots from entering area `A` in step `I` using the following constraint,

```
:- in(A,I), area(A), step(I).
```

and in collaborative assembly tasks, the human can reserve a tool for a period of time. ARROCH uses Restrictor $\mathcal{R}$ to convert H into a set of constraints, $\boldsymbol{\mathcal{C}}$, which is the logical form of human feedback. Next, we describe how $\boldsymbol{\mathcal{C}}$ (from human) is incorporated into multi-robot task planning.

C. Multi-Robot Task Planner

In this subsection, we focus on the multi-robot task planner, $\mathcal{P}^t$ (referred to as **Planner**), that computes one task plan for each robot from the team of robots, while also considering the human feedback (Lines 2 and 16 in Algorithm 1). The input of $\mathcal{P}^t$ include, a set of robot initial states, $\boldsymbol{\mathcal{S}}$; a set of robot goal states, $\boldsymbol{\mathcal{G}}$; and a set of constrained resources, $\boldsymbol{\mathcal{C}}$.

Jointly planning for multiple agents to compute the optimal solution is NP-hard [34]. We adapt an *iterative interdependent planning* (IIDP) approach to compute joint plans for multiple robots [35]. IIDP starts with independently computing one plan for each robot toward completing their non-transferable tasks. After that, in each iteration, IIDP computes an optimal plan for a robot under the condition of other robots' current plans (i.e., this planning process depends on existing plans). While IIDP does not guarantee global optimality, it produces desirable trade-offs between plan quality and computational efficiency.

In the implementation of ARROCH, we use ASP to encode the action knowledge (for each robot) that includes the description of five actions: `approach`, `opendoor`, `gothrough`, `load`, and `unload`. For instance,

```
open(D,I+1) :- opendoor(D,I),
               door(D), step(I).
```

states that executing action `opendoor(D,I)` causes door `D` to be open at the next step. The following states that a robot cannot execute the `opendoor(D)` action, if it is not facing door `D` at step `I`. Such constraints are provided by Restrictor, as described in Section III-B.

```
:- opendoor(D,I), not facing(D,I).
```

The output of $\mathcal{P}^t$ is $\boldsymbol{P}$, a set of task plans (one for each robot). For instance, $p_i \in \boldsymbol{P}$ can be in the form of:

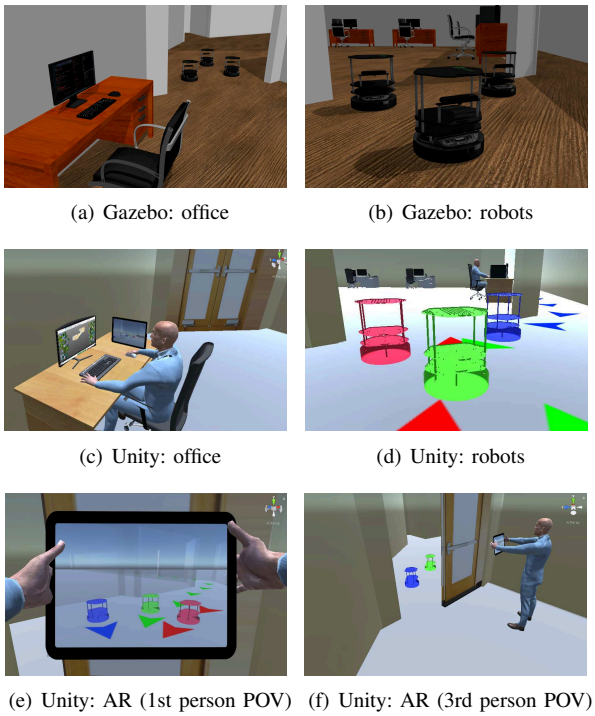
```
load(O,0). approach(D,1). opendoor(D,2).
gothrough(D,3). unload(O,4).
```

suggesting the i^{th} robot to pick up object `O`, approach door `D`, enter a room through `D`, and drop off the object. Next, we describe how ARROCH handles the visualization of robots' current states and their planned actions through an AR interface.

D. AR Visualizer

ARROCH uses an AR interface to bridge the communication gap in human-multi-robot teams, where **Visualizer** $\mathcal{V}$ in Line 13 of Algorithm 1 plays a key role. The input of $\mathcal{V}$ includes the robots' current configurations in C-Space, $\boldsymbol{\omega}$, and their intended motion trajectories, $\boldsymbol{\mathcal{I}}$. The output of $\mathcal{V}$ are spatially visualizable 3D objects which are augmented over the real world. To accurately overlay virtual objects over the real world, AR devices need to localize itself in 3D environments. ARROCH assumes the availability of a set of landmark objects with known 3D poses (position and orientation). Using visual localization, ARROCH computes the pose of the AR device, and then accordingly augments visual information over the mobile device. We use a tablet for AR, though ARROCH is compatible with other mobile devices, e.g., head-mounted displays. Fig. 1 (b) presents an example of the illustrative visualization.

In this section, we describe the ARROCH algorithm and system, the first contribution of this paper. Next, we delineate a novel simulation platform (our **second contribution** of this paper) for prototyping and evaluating AR-based human-multi-robot collaboration algorithms.



(a) Gazebo: office (b) Gazebo: robots (c) Unity: office (d) Unity: robots (e) Unity: AR (1st person POV) (f) Unity: AR (3rd person POV)

Fig. 3. (a)-(b), Gazebo environment for simulating multi-robot behaviors; (c)-(d), Unity environment for simulating human behaviors and her AR-based interactions with the robots; and (e)-(f), Simulated AR interface.

IV. SUGAR2: Simulation with Unity and Gazebo for Augmented Reality and Robots

Our simulation platform has been **open-sourced** under the name of **SUGAR2**.[2] SUGAR2 is built on Gazebo [10] for simulating robot behaviors, and Unity [9] for simulating human behaviors and AR-based interactions. Fig. 3 shows the Gazebo and Unity environments, including a simulated AR interface. Gazebo does not support the simulation of AR-based interactions, and Unity is weak in simulating robot behaviors, which motivated the development of SUGAR2.

V. Experiments

We have conducted experiments in simulation and using real robots. We aim to evaluate **two hypotheses**: I) ARROCH improves the overall efficiency in human-multi-robot team task completion, determined by the slowest agent of a team, in comparison to AR-based methods that do not support bidirectional human-robot communication; and II) ARROCH produces better experience for non-professional users in human-multi-robot collaboration tasks in comparison to non-AR methods. **Hypothesis-I** was evaluated in simulation and using real robots, whereas **Hypothesis-II** was only evaluated based on questionnaires from real human-robot teams.

In both simulation and real-world environments, a human-multi-robot team works on **collaborative delivery tasks**. The robots help people move objects from different places into a storage room, while the human helps the robots open a door that the robots cannot open by themselves. At the same

[2]https://github.com/kchanda2/SUGAR2

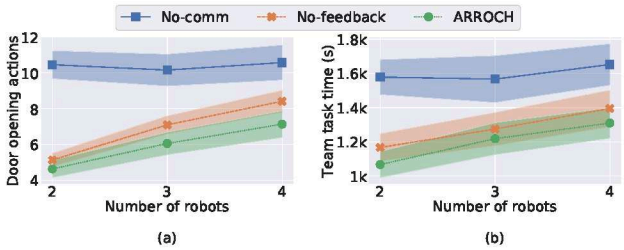


Fig. 4. With different numbers of robots, ARROCH enables the human-robot team to: (a) reduce the human's workload, measured by the number of door-opening actions, and (b) improve the task-completion efficiency, at the same time. The total delivery workload (the number of objects to be delivered) is proportionally increased when the number of robots increases.

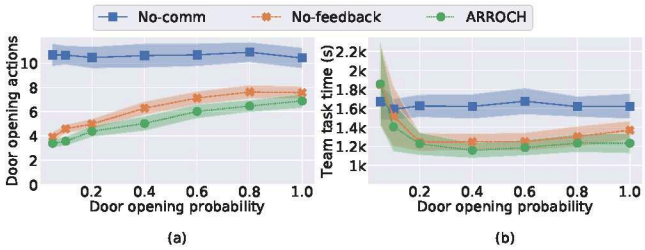


Fig. 5. With different door-opening probabilities (the lower it is, the lazier the human is), ARROCH enables the team to: (a) reduce the number of door-opening actions, and (b) improve the task-completion efficiency, as long as the human is reasonably cooperative (i.e., willing to help the robots in ≥ 0.1 probability).

time, the human has her own dummy task (solving Jigsaw puzzles). Tasks are non-transferable within the human-multi-robot team. The tasks and experiment settings are shared by simulation and real-world experiments.

A. *Simulation Experiments using SUGAR2*

We use the SUGAR2 environment (Section IV) to simulate human and robot behaviors, and their AR-based interactions. ARROCH has been compared with **two baselines**:

- **No-feedback:** It is the same as ARROCH, except that Lines 14-15 in Algorithm 1 are deactivated. As a result, the human can see the robots' current states and planned actions, but cannot provide feedback to the robots.
- **No-comm:** It is the same as No-feedback, except that Line 13 is deactivated. As a result, AR-based communication is completely disabled.

Three types of human behaviors are simulated in SUGAR2. a_0^H, *Work on her own task (simplified as sitting still in Unity)*; a_1^H, *Help robots open the door*; and $a_2^H(N)$, *Indicate unavailability in the next N minutes.*

When the AR-based visualization is activated (ARROCH and No-feedback), the human's door-opening behavior is independently triggered by each waiting robot, meaning that the human is more likely to open the door when more robots are waiting for the human to open the door. For No-comm, in every minute, there is a probability (0.6 in our case) that the human goes to open the door. To realistically simulate human behaviors, we added noise (Gaussian) into the human's task completion time. Before the human completing her task, we randomly sample time intervals during which the human will be completely focusing on her own task. At the beginning of such intervals, there is a 0.9 probability that the human takes action $a_2^H(N)$, i.e., indicating this "no interruption" time length to the robots in ARROCH.

Different Number of Robots: Fig. 4 presents the results in simulation experiments, where we varied the number of robots in the human-multi-robot team. Each robot's workload is fixed. In our case, each robot needs to deliver three objects. Fig. 4 (a) shows that ARROCH produced the lowest number of door-opening actions which directly reduces the human's workload. Fig. 4 (b) shows that ARROCH produced the best performance in the human-multi-robot team's task-completion time, which is determined by the slowest agent. The results support that Hypothesis-I is valid in teams of different sizes.

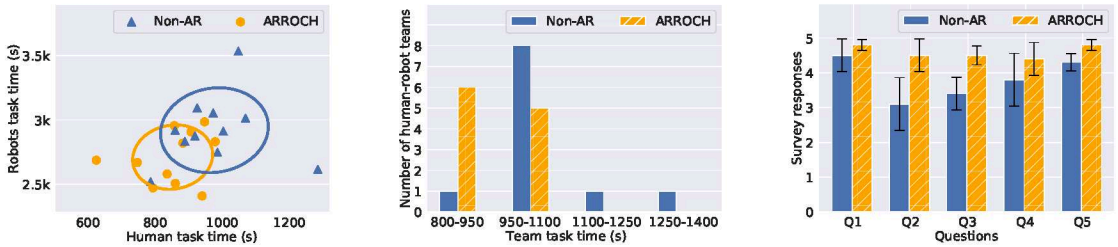
Human Laziness: Fig. 5 shows the results of an experiment, where we evaluated ARROCH's performance under different human behaviors. We varied the human's laziness level that is inversely proportional to the probability that the human opens the door after she sees (through AR) a robot waiting outside. From the results, we see that ARROCH significantly reduced the number of door-opening actions (Left subfigure), while producing the best performance in task completion efficiency (Right subfigure), except for situations where the "door opening probability" is very low, meaning that the human is very lazy. This observation makes sense, because ARROCH only *encourages* lazy people to help robots (so does No-feedback) because the door opening behavior is triggered based on the number of visualized robots. On the other hand, "No-comm" forces people to open the door after every minute. The results support that Hypothesis-I is valid as long as the human is reasonably cooperative.

B. *Real-world Experiments*

We have conducted experiments with human participants collaborating with multiple robots in the real world. The tasks to the human-robot team are the same as those in simulation: robots work on delivery tasks, and the human plays Jigsaw puzzles while helping robots through door-opening actions. The puzzles at the same difficulty level were randomly generated in each trial to avoid the participants' learning behaviors. We selected the Jigsaw game due to its similarity to assembly tasks in the manufacturing industry. The **real-world setup** is shown in Fig. 7(a). We compared ARROCH to a standard **non-AR baseline**, where the human uses a Rviz-based interface to visualize the robots' current states and planned actions, as shown in Fig. 7(b).[3]

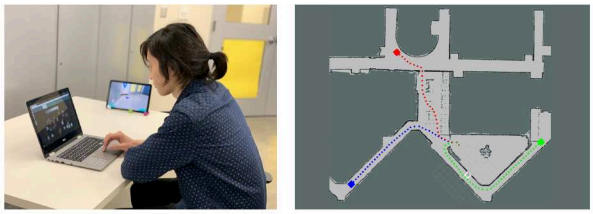
Participants: Eleven participants of ages 20-30, all students from Binghamton University (BU), volunteered to participate in the experiment, including four females and seven males. Each participant conducted two trials using both the Rviz-based baseline, and ARROCH – randomly ordered. None of the participants had any robotics background. There was no training performed on ARROCH or the baseline prior

[3]http://wiki.ros.org/rviz



(a) Individual completion time (b) Team completion time (histogram) (c) Questionnaire scores

Fig. 6. (a) ARROCH performed better than a traditional non-AR baseline (Rviz-based) in individual task completion time; (b) ARROCH performed better, c.f., non-AR, in team task completion time; and (c) ARROCH produces a better user experience based on results collected from user questionnaires.



(a) Human participant (b) Non-AR baseline

Fig. 7. (a) A human participant playing the Jigsaw game alongside our AR interface running on a tablet, while helping the robots open the yellow door, where simulation environment shown in Fig. 3(c) was constructed accordingly; and (b) The Rviz-based interface (non-AR baseline) showing the three robots' current locations and planned trajectories.

to the trials. The experiments were approved by the BU Institutional Review Board (IRB).

Task Completion Time: Fig. 6(a) shows *individual task completion times* produced by ARROCH and the non-AR baseline. The x and y axes correspond to the human's task completion time, and the total of the individual robots' task completion times. The two ellipses show the 2D standard deviations. Results support Hypothesis-I that states ARROCH improves the human-robot team's task completion efficiency. To analyze the statistical significance, we sum up the team agents' individual completion times (both human and robots), and found that ARROCH performed **significantly better** than the non-AR baseline, with $0.01 < p\text{-value} < 0.05$.

We also looked into the *team task completion time*, which is determined by the time of the slowest agent (human or robot). Fig. 6(b) shows the histogram, where the observation is consistent with that of individual task completion time. For instance, in most trials of ARROCH, the teams used $\leq$ 950 seconds, whereas in most trials of non-AR the teams could not complete the tasks within the same time frame. We observed that the participants, who did not have a robotics background, frequently experienced difficulties in understanding the visualization provided by the Rviz-based interface, including the map and robot trajectories. More importantly, ARROCH allows the participants to focus on their own tasks by indicating their unavailability, which is particularly useful in the early phase, whereas Rviz-based communication is unidirectional.

Questionnaires: At the end of each trial, participants were asked to fill out a survey form indicating their qualitative opinion over the following items. The response choices were: 1 (Strongly disagree), 2 (Somewhat disagree), 3 (Neutral), 4 (Somewhat agree), and 5 (Strongly agree). The questions include: Q1, *The tasks were easy to understand*; Q2, *It was easy to keep track of robot status*; Q3, *I could focus on my task with minimal distraction from robot*; Q4, *The task was not mentally demanding (e.g., remembering, deciding, and thinking)*; and Q5, *I enjoyed working with the robot and would like to use such a system in the future.* Among the questions, Q1 is a verification question to evaluate if the participants understood the tasks, and is not directly relevant to the evaluation of our hypotheses.

Fig. 6(c) shows the average scores from the questionnaires. Results show that ARROCH produced higher scores on Questions Q2-Q5, where we observed **significant improvements** in scores in Q2, Q3, and Q5 with the p-values < 0.001. The significant improvements support Hypothesis-II on user experience: ARROCH helps keep track of the robot status, is less distracting, and is more user-friendly. The improvement in Q4 was not significant, and one possible reason is that making quantitative comparisons over the "mentally demanding" level can be difficult for the participants.

VI. Conclusions

Leveraging augmented reality (AR) technologies, we introduce *AR for robots collaborating with a human* (ARROCH), a novel algorithm and system that enables bidirectional, multi-turn, beyond-proximity communication to facilitate collaborative behaviors within human-multi-robot teams. ARROCH enables the human to visualize the robots' current states and their intentions (planned actions), while supporting feedback to the robots. The human feedback is then taken by the robots toward human-robot collaboration. Experiments with human participants showed that ARROCH performed better than a traditional non-AR approach, while simulation experiments highlighted the importance of ARROCH's bidirectional communication mechanism.

Acknowledgment

A portion of work has taken place in the Autonomous Intelligent Robotics (AIR) Group at SUNY Binghamton. AIR research is supported in part by grants from the National Science Foundation (NRI-1925044), Ford Motor Company (URP Awards 2019 and 2020), OPPO (Faculty Research Award 2020), and SUNY Research Foundation.

REFERENCES

[1] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," *AI magazine*, vol. 29, no. 1, p. 9, 2008.

[2] S. H. Ivanov, C. Webster, and K. Berezina, "Adoption of robots and service automation by tourism and hospitality companies," *Revista Turismo & Desenvolvimento*, vol. 27, no. 28, pp. 1501–1517, 2017.

[3] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent advances in augmented reality," Navel Research Lab, Tech. Rep., 2001.

[4] S. A. Green, M. Billinghurst, X. Chen, and J. G. Chase, "Augmented reality for human-robot collaboration," in *Human Robot Interaction*, 2007.

[5] 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.

[6] R. T. Chadalavada, H. Andreasson, R. Krug, and A. J. Lilienthal, "That's on my mind! robot to human intention communication through on-board projection on shared floor space," in *2015 European Conference on Mobile Robots (ECMR)*. IEEE, 2015, pp. 1–6.

[7] R. K. Ganesan, Y. K. Rathore, H. M. Ross, and H. B. Amor, "Better teaming through visual cues: how projecting imagery in a workspace can improve human-robot collaboration," *IEEE Robotics & Automation Magazine*, vol. 25, no. 2, pp. 59–71, 2018.

[8] M. B. Luebbers, C. Brooks, M. J. Kim, D. Szafir, and B. Hayes, "Augmented reality interface for constrained learning from demonstration," in *Proceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI)*, 2018.

[9] 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.

[10] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[11] J. Y. Chai, L. She, R. Fang, S. Ottarson, C. Littley, C. Liu, and K. Hanson, "Collaborative effort towards common ground in situated human-robot dialogue," in *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*. ACM, 2014, pp. 33–40.

[12] J. Thomason, S. Zhang, R. J. Mooney, and P. Stone, "Learning to interpret natural language commands through human-robot dialog," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[13] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox, "Learning to parse natural language commands to a robot control system," in *Experimental Robotics*. Springer, 2013, pp. 403–415.

[14] S. Amiri, S. Bajracharya, C. Goktolga, J. Thomason, and S. Zhang, "Augmenting knowledge through statistical, goal-oriented human-robot dialog," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[15] S. Waldherr, R. Romero, and S. Thrun, "A gesture based interface for human-robot interaction," *Autonomous Robots*, vol. 9, no. 2, pp. 151–173, Sep 2000.

[22] A. Watanabe, T. Ikeda, Y. Morales, K. Shinozawa, T. Miyashita, and N. Hagita, "Communicating robotic navigational intentions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[16] K. Nickel and R. Stiefelhagen, "Visual recognition of pointing gestures for human–robot interaction," *Image and vision computing*, vol. 25, no. 12, pp. 1875–1884, 2007.

[17] H.-D. Yang, A.-Y. Park, and S.-W. Lee, "Gesture spotting and recognition for human–robot interaction," *IEEE Transactions on robotics*, vol. 23, no. 2, pp. 256–270, 2007.

[18] R. T. Azuma, "A survey of augmented reality," *Presence: Teleoperators & Virtual Environments*, vol. 6, no. 4, pp. 355–385, 1997.

[19] T. Williams, D. Szafir, T. Chakraborti, and E. Phillips, "Virtual, augmented, and mixed reality for human-robot interaction (vam-hri)," in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2019, pp. 671–672.

[20] J. Park and G. J. Kim, "Robots with projectors: An alternative to anthropomorphic hri," in *Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction*, 2009.

[21] G. Reinhart, W. Vogl, and I. Kresse, "A projection-based user interface for industrial robots," in *IEEE Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems*, 2007.

[23] P. Milgram, S. Zhai, D. Drascic, and J. Grodski, "Applications of augmented reality for human-robot communication," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1993.

[24] S. A. Green, J. G. Chase, X. Chen, and M. Billinghurst, "Evaluating the augmented reality human-robot collaboration system," *International journal of intelligent systems technologies and applications*, vol. 8, no. 1-4, pp. 130–143, 2010.

[25] H. Hedayati, M. Walker, and D. Szafir, "Improving collocated robot teleoperation with augmented reality," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2018.

[26] C. Brooks and D. Szafir, "Visualization of intended assistance for acceptance of shared control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[27] F. Muhammad, A. Hassan, A. Cleaver, and J. Sinapov, "Creating a shared reality with robots," in *Proceedings of the 14th ACM/IEEE International Conference on Human-Robot Interaction*, 2019.

[28] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: theory and practice*. Elsevier, 2004.

[29] A. E. Gerevini, P. Haslum, D. Long, A. Saetti, and Y. Dimopoulos, "Deterministic planning in the fifth international planning competition: Pddl3 and experimental evaluation of the planners," *Artificial Intelligence*, vol. 173, no. 5-6, pp. 619–668, 2009.

[30] V. Lifschitz, "What is answer set programming?." in *AAAI*, vol. 8, 2008, pp. 1594–1597.

[31] S. Zhang, F. Yang, P. Khandelwal, and P. Stone, "Mobile robot planning using action language bc with an abstraction hierarchy," in *International Conference on Logic Programming and Nonmonotonic Reasoning*. Springer, 2015, pp. 502–516.

[32] C. Baral, *Knowledge representation, reasoning and declarative problem solving*. Cambridge university press, 2003.

[33] M. Gelfond and Y. Kahl, *Knowledge representation, reasoning, and the design of intelligent agents: The answer-set programming approach*. Cambridge University Press, 2014.

[34] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, "Conflict-based search for optimal multi-agent pathfinding," *Artificial Intelligence*, vol. 219, pp. 40–66, 2015.

[35] Y. Jiang, H. Yedidsion, S. Zhang, G. Sharon, and P. Stone, "Multi-robot planning with conflicts and synergies," *Autonomous Robots*, 2019.