GhostAR: A Time-space Editor for Embodied Authoring of Human-Robot Collaborative Task with Augmented Reality

Yuanzhi Cao; Tianyi Wang; Xun Qian, Pawan S. Rao, Manav Wadhawan, Ke Huo, Karthik Ramani
School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907 USA
[cao158, wang3259, qian85, rao81, mwadhawa, khou, ramani]@purdue.edu

Figure 1: GhostAR workflow. To author HRC tasks that achieve time-space coordination, (1) user first authors a human ghost by recording his body movement, (2) then using the ghost as a visual reference, (3) author collaborative robot actions. (4) When acting the task, our system’s collaborative model captures the body movement as input, maps it with the authored human motion, and outputs the corresponding collaborative robot motion.

ABSTRACT
We present GhostAR, a time-space editor for authoring and acting Human-Robot-Collaborative (HRC) tasks in-situ. Our system adopts an embodied authoring approach in Augmented Reality (AR), for spatially editing the actions and programming the robots through demonstrative role-playing. We propose a novel HRC workflow that externalizes user’s authoring as demonstrative and editable AR ghost, allowing for spatially situated visual referencing, realistic animated simulation, and collaborative action guidance. We develop a dynamic time warping (DTW) based collaboration model which takes the real-time captured motion as inputs, maps it to the previously authored human actions, and outputs the corresponding robot actions to achieve adaptive collaboration. We emphasize an in-situ authoring and rapid iterations of joint plans without an offline training process. Further, we demonstrate and evaluate the effectiveness of our workflow through HRC use cases and a three-session user study.

Author Keywords
Human-Robot collaboration; Human-Robot interaction; Time-space editing; Program-by-demonstration; Augmented reality; Embodied interaction; Embodied authoring.

CCS Concepts
• Information systems → Spatial-temporal systems; Multimedia content creation; • Human-centered computing → Interactive systems and tools;

INTRODUCTION
Robotics has been extensively used to automate a large number of particular and repetitive tasks with high accuracy and throughput in manufacturing environments. The tremendous economic and social impacts projected by robotics will be likely to expand in our future by infiltrating into broader fields in both commercial and consumer markets [37]. Unlike traditional manufacturing environments, these new segments, including medical, health care, and services, usually heavily involve human activities in the working environments. Thus, enabling robots to co-work with humans in collaborative tasks has become a significant pillar of the next generation robotics technology.

A typical human-robot-collaborative (HRC) task involves generating a joint intention, planning actions, and acting cooperatively [10]. In a human-centered task, the joint intention usually aligns with humans’ implicit or explicit expressions. Explicit communications such as speech and gestures have been widely studied for commanding robots [20, 63]. However, using these modalities may cause inefficiencies and ambiguities in spatially and temporally coordinated collaborations that require a comprehensive understanding of the contexts. On the other hand, embodied demonstrations from humans directly convey the intentions to the robots. More importantly, to avoid programming robots’ behaviors for the highly dynamic human-robot interactions, researchers propose programming by demonstrations (PhD) to generate task and action plans for the robots [17]. Further, to safely and robustly execute
the action plans in a coordinated manner, humans and robots need to communicate with their status, actions, and intentions timely [38]. To this end, we primarily endeavor to explore the design of an embodied authoring workflow to support real-time human motion inference, demonstrating example actions to robots and creating joint plans.

The advents of mobile computing foster the evolution of authoring workflows in an in-situ and ad-hoc fashion [42, 26]. However, existing workflows primarily target at pre-defined and rigorous tasks where robots operate in isolation and interact with the environment only. To enable novice user-friendly PhD in the authoring workflows, we need to support human motion capture and inference which traditionally involve a motion capture system. Since a body-suit [7] or an external-camera [3] based capture system requires heavy dependencies, demonstrations are often only captured off-line [8]. Moreover, for ad-hoc tasks, demonstrating with users’ bodies is preferable [17]. Recently, the emerging augmented/virtual reality (AR/VR) technologies, e.g., head-mounted AR/VR devices [1, 2], show a strong potential to enable embodied authoring [35]. Further, in HRC tasks, robot partners are desired to adapt to and coordinate with humans’ actions. Thus, to create a joint action plan, the counterpart motions of the robots can only be demonstrated with the humans’ part as contexts. In this work, we promote a critical advantage of using AR/VR authoring, namely externalizing the users’ body asynchronously [41, 69]. This way, the users can always view, manipulate, and edit their own recorded actions, and use them as contexts when demonstrating the counterpart motions for robots.

We promote an embodied authoring in AR for HRC tasks in this work because of the following reasons: (i) realistic visualization with contextual and spatial awareness, enabling creating, editing, and previewing the collaborative flow intuitively; (ii) easy programming with natural embodied interaction through real human demonstration via role-playing to establish time-space correspondence; (iii) supporting real-time motion inference, activity detection, and visual feedback on robots’ intents when conducting the HRC. We present GhostAR workflow which uses AR with body-tracking to enable visual, spatial, and embodied HRC tasking authoring, as illustrated in Figure 1. A typical authoring session starts when users role-play the human’s actions. We render the recordings as AR ghost. Users can freely observe, edit, and infer the actions and use it as a reference when role-playing the robot’s counterpart actions. Then, users designate correspondences between humans’ action plan and the demonstrated actions for robots. Further, GhostAR provides visual preview with AR simulation in-situ. When users act the HRC tasks, GhostAR continues to capture the user’s motion and use it to derive the robot’s motion plan. Also, users can refer to the next-step guidance and the robot’s intentions with AR visual feedback. In summary, we highlight our contribution as follows.

- A system workflow for authoring human–robot collaborative task through AR ghost as contextual references and role-playing with natural embodied interaction.

- A lead-assist collaboration model that achieves time-space correlation for the human-lead-robot-assist adaptive collaboration task based on dynamic time warping (DTW) algorithm.

- An AR interface and interaction design for human–robot ghost creation and visualization, editing, and manipulation, previewing and simulation, and guidance throughout a successful collaborative action.

RELATED WORK

Human-Robot Collaboration Model

Many cognitive frameworks and computational architectures have been proposed for enabling and supporting teamwork between humans and robots [65]. One of the keywords in human–robot collaboration (HRC) is *adaptation*: a robot interacting with people needs to reason over its uncertainty over the human internal state, as well as over how this state may change, as humans adapt to the robot [48]. While some previous work took the approach of human adapting to robot [49], and human-robot mutual adaptation [52], the largest body of current HRC works have been focusing on a lead-assist collaboration type and empowering the robot to be an assistant and to adapt to human actions. Researchers have presented various mathematical models and formulations focusing on task allocation and communication via goal-oriented controller [61], improving human–robot coordination through cross-training [51], and efficient learning with human inference with joint-action demonstrations [50]. Other researchers emphasized on robot learning methods and frameworks and proposed interactive primitive. Along this thread, a series of studies demonstrated cooperative task learning with single [8] and multiple [22] primitives. Further, probabilistic movement model has been introduced to improve human–robot coordination [44] and action recognition [43]. Other alternative authoring and planning-based methods have been proposed to achieve similar goals. For example, Koppula et al. developed a Markov Decision Processes (MDP) based model for human–robot-collaboration tasks in contextually rich environment [39], while Szafrir et al. designed three collaborative interfaces to support human-flying robot collaboration [64]. These works primarily targeted at general mathematical solutions and learning methods for specific collaborative scenarios. However, it is still challenging to achieve applicable human-robot collaboration in real-world setups. Most of the HRC tasks were pre-defined and simplified versions of intended scenarios [65]. Also, many of these works require offline training with pre-capture data, which is not desired for on-site HRC.

On the other hand, our system complements the previous works by focusing on providing an in-situ HRC task authoring tool. We exploit the initiative of human users and enhance their capabilities with embodied interactions and AR interfaces. To better support a smooth workflow and rapid iteration of task plans, we adopt a real-time process for task authoring and collaboration acting without offline training. Taking advantages of the AR interface, we also provide active visual feedback with spatial and contextual reference so that human and robot are always aware of each other during the collaboration.
Robot Programming by Demonstration

Robot programming by demonstration (PbD), also referred to as imitation learning, has become a popular method for programming and training robots. PbD reduces search space complexity for learning, supports natural means of embodied user interaction, thus enables flexible and user-friendly robot programming and training [11]. A large body of works have been done in developing methods and algorithms for learning individual motions [21, 62, 54] and compound motions [47, 19], as well as incremental teaching methods [53, 58]. So far, PbD has shown great success in training individual robots to do specific tasks with offline data captures. When applying PbD into collaborative scenarios, additional reference is needed since the robot is no longer operating in isolation. Instead, robots need to coordinate with the human partner, whose uncertainty depends on human’s internal states upon actions. To achieve PbD for HRC tasks, previous works primarily relied on two people demonstrating the tasks where one of them plays the robot’s role. The human demonstration is captured with a motion tracking system offline and fed a computational model to generate robot policy at runtime [8, 22, 44, 66, 67]. The above approach is intuitive to practice and has been used in HRC task authoring, including object handover and joint manipulation. However, this PbD approach is limited to pre-determined and straightforward task authoring due to the lack of visual interface for sophisticated editing. Moreover, as the offline demonstrations usually happened in a controlled lab environment, the collaboration volume was constrained, e.g., most of the presented collaboration tasks were executed using a stationary robot arm.

GhostAR, on the other hand, exploits a visual interface and displays the captured human motion as ghost images in the AR scene. Using the AR ghost as time-space references, users can author the HRC tasks by manipulating a virtual avatar of the real robot collaborators. We emphasize instantiating PbD by supporting embodied authoring in our workflow. Our system allows for collaborative tasks authoring of robots with various types of configurations. Further, when users perform the collaborations with robots, we allow users to use the same self-contained AR interface for motion inference.

Human-Robot Interaction through Augmented Reality

An AR interface is spatially and contextually aware of the surrounding environment by its nature [12]. Thus, it serves as an ideal media to bridge the digital interface and physical reality. For example, it has been used for visual and spatial interactions with robots [29, 14, 36, 15] and smart devices [31, 32]. AR for human-robot interaction has been widely explored across industrial motion planning [23, 24, 18], mobile tele-operation [30, 40, 29, 36], sequential task planning [45, 42, 34, 26, 56], and multi-robot controlling [25], analyzing [28], and debugging [46]. Previous works primarily treated AR as a control interface for robots operating in isolation. While AR was explored to display robot’s intent for user visualization to achieve better collaboration [27, 68, 59, 55, 9, 16], it has not been proposed to empower the entire life-cycle of HRC, from task authoring to collaboration acting. To the best of our knowledge, GhostAR is the first system that achieves the incorporation of AR within a full HRC workflow, enabling naturally embodied authoring with context-aware visual programming.

The key to HRC task authoring is to provide a reference of the collaboration partner spatially and temporally during the authoring process, which in turn ensures correct time-space coordination when the HRC task is in action. By further exploring into human-human scenarios, we have found several exciting AR works that achieve augmented collaboration through interactively reconstructing the surrounding environment [41], spatially visualizing the collaboration partners [33], and demonstratively externalizing user’s body [69]. Informed and inspired by these recent works, we introduce a novel ghost visualization serving in a human-robot scenario for collaboration reference, authoring, and editing, as well as simulation and preview of authored joint action plans.

DESIGN GOALS

We have derived the following Design Goals (DG) from the design rationale of our approach. The motivation for DGs has been extensively discussed in the RELATED WORK. An essential requisite for HRC is the adaption between the two parties of the collaboration: the human and robot. We chose the lead-assist type of HRC due to the scope of this work, hence adapting the robot to the human (DG1). Program-by-Demonstration (PbD) has been considered as one of the easiest ways of programming robot behavior through natural body movement with a shallow learning curve (DG2). AR can supplement PbD with a digital interface, that is in-situ and spatially situated. This enables us to create an authoring interface with contextual awareness (DG3) and rich digital visualization (DG4). The in-situ nature also promotes fast iteration with real-time feedback (DG5). Later we will describe how these DGs guide the design of our system.

DG1: Adapting robot behavior to human. Author human-lead-robot-assist typed collaborative tasks that are initiated by a human, where the robot always act adaptively to the human partner’s actions.

DG2: Programming with natural interaction. Lower the barrier for users to effectively program complex HRC tasks, with natural body movement and intuitive interactions.

DG3: Authoring with contextual awareness. Provide spatial and contextual awareness that is important for Human-Robot task authoring. Both parties need to be aware of each other’s position and status, as well as the surrounding environment.

DG4: Visualizing with realistic simulation. Give active and accurate visual feedback about what the user has authored, to ensure efficiency and correctness of the authoring through realistic simulations.

DG5: Iterating with real-time feedback. Enable a real-time process and rapid iterations from collaborative task authoring to action, with no need for offline programming and testing.

GHOSTAR

Human-Robot Collaboration Model (lead-assist type)

It is important to first define the meaning of collaboration in our work as it touches a wide range of aspects, even just for
tasks between humans and robots. In GhostAR, we essentially present a robot programming tool that controls robots’ actions based on its human partner’s body movement. In other words, the robot collaborates with the human in the sense that it must act adaptively according to its human partner (guided by DG1). To achieve this, we present a collaboration model that is dynamically generated based on the user’s authoring and is able to output robot action corresponding to the input human motion.

In a human-lead-robot-assist HRC task, we achieve motion coordination by defining user’s action segments first. Our system allows users to record their body movement as a Human Motion Clip (a sequence of Motion Frames with different timestamps) and to use it to create HRC tasks. Note that the authored human motion could consist of several meaningful movements, and the user can put them into Groups to author HRC tasks correspondingly. For example, the human character in Figure 2 records the following motion: he/she walks, stops, and waves his/her hand, then walks for some distance and waves again. The HRC task the user wishes to author is to make the robot come over when he first waves, follow him and shoot videos for him as he walks, and then leaves when he/she waves hand again. To achieve this, the user needs to put the two hand waving and a walking into three Groups and author the robot to behave as come over, follow and shoot videos, and leave correspondingly in these three Groups. For each Group of human motion, our system provides two types of collaborative tasks for the user to author. They are Synchronize and Trigger tasks.

- A Synchronize task authors a robot action to take place at the same pace of the reference human group. In this type of HRC task, robot and human will perform their own task, but at the same speed or progress, i.e., if the human moves faster, the robot will move faster to keep up, and vice versa. This applies to HRC tasks such as joint object manipulation, motion following for lighting or camera shooting, and coordinated movements like hand-shaking, etc.

- A Trigger task authors a robot action to take place after the human group. In this type, the robot starts executing its authored task right after the human has completed the reference group, i.e., human snaps his finger, and the robot starts sweeping the floor. This applies to HRC scenarios such as sequential joint assembly, and gesture signaling, etc.

As for the example in Figure 2, the user will author the come over and leave robot action as Trigger tasks for the two hand wave Groups, and author the follow and shoot video as a Synchronize task for the walk Group.

So far, the user has been preparing the collaboration by creating the HRC TaskSequence. As shown in Figure 3, the HRC TaskSequence is a list of Groups that represents the authored task in an accessible and manageable manner. Note that adjacent ungrouped human Motion Frames will be automatically grouped as Empty Groups. The HRC TaskSequence together with the Motion Mapping module, form the collaboration model of GhostAR. When the HRC action is started, the user needs to repeat his authored motion in the sequential order. Meanwhile, our system will activate the first Group and start the motion mapping between the real-time captured human motion and the grouped Human Motion Clip. When the mapping progress indicates the current Group is completed, our system activates the next one and repeats this process until all Groups in the HRC TaskSequence are completed. For a Synchronize task, the system calculates the progress and output robot behavior at the corresponding timestamp. For Trigger task (which is generally shorter), the system focuses on recognizing the completion of the human movement and then issues commencement instructions for the authored robot actions. Note that Empty Group will be treated the same way as a Synchronize Group, for proper progress monitoring and activation of the next Group.

**Embodied Authoring with Augmented Reality**

Our system’s interaction workflow is implemented as a state machine, where a HRC task is authored with the following five modes: Human Authoring Mode, Robot Authoring Mode, Observation Mode, Preview Mode, and Action Mode. At the beginning of a new task authoring session, a user is first asked to choose the robot collaborator(s). Note that in the case of simultaneously collaborating with multiple robots, each robot will share the same Human Motion Clip but has its own HRC TaskSequence. After initialization, the user will be promoted to the Human Authoring Mode to create the first Human Motion Clip. After finishing the creation, the current tasks are displayed as AR ghost for visualization and manipulation in the Observation Mode. The user uses the cursor to perform a Grouping operation, and authors robot tasks for the selected
Figure 4: GhostAR system interface. (1) The user first acts out the human part of an HRC task in Human Authoring Mode via embodied movement. The human motion is captured and represented as AR ghosts for editing in Observation Mode. (2) Using the human ghosts as the time-space contextual reference, the user then authors collaborative robot action in the Robot Authoring Mode. (3) During the Action Mode, the user plays out HRC task while following the AR ghost to repeat his previously authored human motion.

**Group** in the **Robot Authoring Mode**. Our system adopts robot authoring via manipulating a virtual AR robot, as opposed to a real robot. This approach enables easy programming of a mobile robot with spatial movement and object interaction while using the human ghost as a time-space reference. The authored robot behavior will then be displayed as robot ghosts, together with human ghosts. The human-robot ghosts form an expressive and editable HRC. In the **Observation Mode**, user can choose to enter **Preview Mode** to visualize the entire HRC task simulation with AR ghost animation. Once the user is satisfied with the authored task, he/she can act out the authored HRC tasks by entering the **Action Mode**. The system utilizes the dynamically generated collaboration model to derive the corresponding robot behaviors based on the user’s real-time motions.

**Human Authoring Mode.** The **Human Motion Clip** is the baseline of the HRC task authoring. It contains the human motion that the robot will collaborate with, as well as the movement that the user needs to repeat during the **Action Mode**. Guided by DG2, the authoring of the **Human Motion Clip** is achieved through natural embodied movement, where the system records the user’s body motion by tracking the position and orientation of the AR headset and two hand-held controllers. Then the **Human Motion Clip** will be represented by segmented ghost avatars and displayed in the user’s AR view, as illustrated in Figure 4-(1). The ghost avatar also plays the authored human movement repeatedly as an animation in real-time scale for review. To extend the **Human Motion Clip**, the user first needs to trigger the last pose in the recorded clip and then act new human motion, which will automatically be tailed to the end of the current **Human Motion Clip**.

**Robot Authoring Mode.** Once the **Human Motion Clip** is created, user can pick a segment from it and generate a **Group**, then author a **Synchronize** or **Trigger** robot task for it. For each selected robot collaborator, there exists a virtual robot avatar in GhostAR that mimics the behavior of the real robot. User can control the virtual robot, with the hand-held controllers and physical movements, to facilitate the robot motion authoring. Guided by DG3, we establish the time-space correlation between the robot and the human by utilizing the human ghost as the contextual reference. For a **Synchronize** task, the time-length of the robot clip is equal to that of the human group. As the user is authoring robot and progressing, the human ghost with the same timestamp will be displayed as AR reference to assist the user, as illustrated in Figure 4-(3). The user can pause/resume and walk around anytime during the authoring process in order to observe and operate the robot avatar from the optimal perspective. In terms of a **Trigger** task, the user authors robot actions independently which will be placed **after** the **Trigger Group**. Once robot authoring is finished, the authored HRC task will be animating repeatedly, with both human and robot ghosts, to visualize and preview the task before the user decides to accept or redo (guided by DG4).

**Ghost Visualization and Manipulation.** Our system provides in-situ authoring experience by exploiting the advantage of AR interfaces, thus promotes rapid iteration without offline preparation (guided by DG5). In the **Action Mode**, the authored tasks are displayed as AR ghosts for the user to preview and manipulate. The ungrouped raw human ghosts are displayed as transparent segmented snapshots while the grouped ghosts are displayed with Start/End **Motion Frames** with a uniquely assigned color and a floating 3D icon indicating its collaboration type, as illustrated in Figure 4-(2). Using the interactive cursor, user can edit the **Human Motion
Clip and perform operations such as Grouping, unGrouping, ‘trimming’, etc. If the cursor is pointing at any unGrouped raw human ghost, the pointed ghost Human Frame will be highlighted. Otherwise, if the cursor is inside a Group, the Human-Robot task of that group will animate repeatedly until the cursor is moved outside. Note that the user can also enter the Preview Mode and visualize the entire task as a continuous simulated AR ghost animation.

**Action Mode.** The Action Mode is where the user carries out the collaboration tasks. In this Mode, the system captures the real-time movement of the user and maps it with the recorded Human Motion Clip, then issues corresponding instructions to drive the robot and perform the collaborative task. To help the user repeat his authored motion and alleviate the mental burden of memorization, the system provides numerous AR guidance to assist the user. As illustrated in Figure 4-(4), our system not only projects a dotted trail for the user to follow, but also plays the next-to-act Group’s animation to refresh the user’s memory. Therefore, the user only needs to focus on the current task, and the system is guiding him/her step-by-step. Besides, our system also provides numeric progress information for the user to keep track of him/herself as well as the robot’s working status.

**Motion Mapping using Dynamic Time Warping**

We describe how our system achieves motion mapping for both Synchronize and Trigger tasks. Essentially, in order to recognize the user’s status, we rely on positions of the user’s head and both hands which are provided by our AR interface. We then introduce DTW to infer the user’s activities using the nine degree-of-freedom (DOF) inputs. At the time $t_i$, the user’s state is represented by a $\mathbb{R}^9$ vector:

$$v_{ti} = [x_{h_i}, y_{h_i}, z_{h_i}, x_{l_i}, y_{l_i}, z_{l_i}, x_{r_i}, y_{r_i}, z_{r_i}]^T$$

In this manner, each Human Motion Clip derives a $\mathbb{R}^9$ curve as: $L_{record} = [v_{t1}, v_{t2}, v_{t3}, \ldots, v_{tn}]$. And we denote the human motion in Group $G_i$ as $l_{Gi}$ which is a continuous segment within $L_{record}$.

To reduce the DOF of the inputs and to keep the most relevant information from the raw gesture data $l_{Gi}$, we apply principal component analysis (PCA) [13] to project this $\mathbb{R}^9$ curve onto a $\mathbb{R}^2$ plane. A projected curve $f_{Gi}$ and a projection matrix $P_{Gi}$ are derived as well as in Algorithm 1. For each activated Group $G_i$, the real time data $v_{tnow}$ is projected by $P_{Gi}$ and then compared with the $f_{Gi}$ to acquire the corresponding progress in $G_i$.

**Algorithm 1 Calculate Projected Curve and Projection Matrix**

1. procedure PCA_PROJECTION($l_{Gi}[1 \ldots n]$)
2.   $\overline{L_{Gi}} \leftarrow (\Sigma_{Gi})/(9 \times n)$
3.   $V \leftarrow (l_{Gi} - \overline{L_{Gi}})(l_{Gi} - \overline{L_{Gi}})^T$
4.   Let $v_1$ and $v_2$ be two eigen vectors associated with the largest eigen values of $V$
5.   output $P_{Gi} \leftarrow [v_1, v_2]^T$
6.   output $f_{Gi} \leftarrow P_{Gi}l_{Gi}$

**Trigger Task Detection.** Assume that an activated Group $G_i$ is a trigger Group and we want to determine whether the user has finished performing the human motion $l_{Gi}$. We first collect the motion that the user has just performed: $l_{realtime} = [v_{tnow} - n+1, \ldots, v_{tnow} - 1, v_{tnow}]$ where $n$ is the length of $l_{Gi}$. Then, we get the projected curve $f_{realtime} = P_{Gi}l_{realtime}$ and compare it with $f_{Gi}$. This method is close to a conventional human action recognition problem [60]. We use Dynamic Time Warping (DTW) algorithm [57] to calculate the similarity. DTW is an algorithm to find the alignment between two time series data. Given two time series $s = [s_1, s_2, \ldots, s_n]$ and $t = [t_1, t_2, \ldots, t_m]$ with length $n$ and $m$, a distance matrix $D$ is calculated using Algorithm 2. Each element $D[i, j]$ in the distance matrix $D$ is the distance between $s[1 : i]$ and $t[1 : j]$ with best alignment. And we define $D[n, m]$ as DTW distance between $s$ and $t$, note as $<s, t>$. In our specific case, if $<f_{realtime}, f_{Gi}>$ reaches a local minimum and this minimum value is smaller than $\varepsilon$, we assume that this minimum value is the global value and report to the system that $G_i$ is triggered by the user. To adapt this threshold for different $f_{Gi}$ with various lengths, we set $\varepsilon = a + n$ where $a$ is a fixed coefficient.

**Synchronize Task Progress Estimation.**

If an activated Group $G_i$ is a Synchronize task, we need the user’s progress ($0\% \sim 100\%$) in order to temporally coordinate the robots’ motions. We propose to compare the real time data $l_{realtime} = [v_{tnow} - n+1, \ldots, v_{tnow}, v_{tnow}]$ with the sub-sequences of $l_{Gi}$: $l_{Gi}[1], l_{Gi}[1:2], \ldots, l_{Gi}[1:n]$, where $t_{stau}$ is the time when $G_i$ is activated. And we derive the user’s progress as $n’/n$ if the sub-sequences $l_{Gi}[1:n’]$ approximates $l_{realtime}$ the most. In other words, we first project $l_{realtime}$ to $f_{realtime}$ using $P_{Gi}$ and calculate the DTW distances between $f_{realtime}$ and the sub-sequences of $f_{Gi}$: $f_{Gi}[1], f_{Gi}[1:2], \ldots, f_{Gi}[1:n]$, noted as $d_1, d_2, \ldots, d_n$. And find $n’ = \arg \min_{1 \leq i \leq n}(d_i)$. However, we note that the scale of $d_i$ is influenced by the length of the sub-sequence $f_{Gi}[1:i]$. To eliminate this influence, a modified DTW distance $d_i’ = d_i/\sqrt{i}$ is introduced. Then we determine a sub-sequence $f_{Gi}[1:n’]$ that is best aligned with $f_{realtime}$ while $n’$ is given by $n’ = \arg \min_{1 \leq i \leq n}(d_i’)$, and
Algorithm 3 Progress Estimation Using DTW

1: \( d_{old} \leftarrow \text{array}[0 \ldots n], d_{new} \leftarrow \text{array}[0 \ldots n] \)
2: \( i \leftarrow 0, n \quad \text{do} \quad d_{old}[i, 0] \leftarrow \infty \)
3: \( i \leftarrow 0, n \quad \text{do} \quad d_{new}[i, 0] \leftarrow 0 \)
4: \( \text{while Synchronized Task } S_i \text{ has started do} \)
5: \( \text{if } \nu_{\text{now}} \text{ is updated then} \)
6: \( f_{\text{now}} \leftarrow P_S \nu_{\text{now}} \)
7: \( \text{for } i \leftarrow 1, n \quad \text{do} \)
8: \( d_{new}[i] \leftarrow |f_S[i] - f_{\text{now}}| + \min(d_{new}[i - 1], d_{old}[i - 1]) \)
9: \( n^* \leftarrow \arg\min_{1 \leq i \leq n} (d_{new}[i] / \sqrt{i}) \)
10: \( d_{old}[i] \leftarrow d_{new} \)
11: \( \text{output } \text{progress} \leftarrow n^*/n \)

IMPLEMENTATION
System Setup and Development

We build our see-through AR platform by attaching a stereocamera (ZED Dual 4MP Camera (720p)) in front of a VR headset (Oculus Rift). Four external Oculus IR-LED Sensors track the human body motion with an active working area of 5mx5m. Two Oculus Touch Controllers enable interactions used in the system. The major part of GhostAR software system is developed with Unity3D engine and Robot Operating System (ROS)[5], including the AR interface and embodied interaction, motion recording, and DTW calculation, etc. The authored Human Motion Clip and robot clips are recorded at the rate of 90Hz. It is worth to note that this prototyped AR platform still relies on external tracking and tethered computer, which limits the interaction volume. However, with the newly developed mobile AR/VR technologies, e.g., Hololens [1] and Oculus Quest [2], we believe that implementing GhostAR with stand-alone devices would not involve much effort.

Robot Simulation and Prototyping

We have prototyped several robots, including three physical robots (GripperBot, CamBot, ArmBot) and a virtual robot drone, for use case demonstration and studying the effectiveness of robot authoring user interaction. The CamBot is an omni-mobile robot with a camera mounted. The ArmBot is a fixed 6-DOF robot arm (Arduino TinkerKit Braccio). The GripperBot is an omni-mobile robot with the 6-DOF robot arm sitting on top of it. As is illustrated in Figure 5, the mobile robot base is powered by 3 DC motors (locally controlled by Arduino) driven by omni wheels that are capable of moving towards any direction while rotating. The robot is equipped with an NVIDIA Jetson TX1 Development Kit running ROS as the robot’s central controller and with a SICK TiM 561 2D LIDAR for SLAM navigation. The robot is powered by four LiPo batteries (11.1V, 5000mAh for each battery). During the Robot Authoring Mode, in order to deliver realistic virtual robot simulation that closely resembles the dynamics and physical behavior of the real robot, we adopt ROS-Gazebo [4] as back-end robot simulator, the workflow is illustrated in Figure 5. In detail, the controller inputs are sent to ROS-Gazebo using ROS#-Unity protocol[6] via WiFi communication. ROS-Gazebo then simulates the motion of the robot under dynamic and physical constraints (maximum torque, speed, acceleration, etc). Meanwhile, it simultaneously pushes the real-time robot status back to Unity3D where the virtual robot is then rendered accordingly in the user’s AR view. In this way, users can experience realistic robot manipulation and visualization with virtual robot avatars. Within the Action Mode, our collaboration model derives the corresponding robot behavior into ROS-Gazebo, which then instructs the physical robot to act accordingly.

USE CASE SCENARIOS

Figure 6 illustrates four use case scenarios of GhostAR. Figure 6-(1) demonstrates our primary use case, involving the human user simultaneously collaborating with two robots for both Synchronize and Trigger tasks. In this use case, the human walks towards the table with a red object in his hand to be put onto the table in the designated area. His body motion of ‘bending over and place the object’ is authored as a Trigger for the robot arm to grab the red object and place it into the basket. Meanwhile, the human motion when he is walking towards the table is authored as a Synchronize task for the CamBot to follow and videotape the whole process, in order to get the best shooting angle. Figure 6-(2) demonstrates a joint assembly task with the ArmBot where the user provides the bottom part of the assembly, and the ArmBot grabs the top part and assembles them. The task is authored as a Trigger action and can be performed repeatedly. Figure 6-(3) demonstrates a scenario where a drone is providing spotlight for the user while he/she walks towards the couch, sits down, and puts the round object into the container. The entire HRC action is authored as one Synchronize task. Figure 6-(4) demonstrates a Synchronize hand-shaking scenario where the robot reaches out its gripper at the same pace as the human reaches out his/her hand, e.g., it pauses if the human pauses, and proceeds when the human proceeds.

USER STUDY

To evaluate our collaboration model accuracy, robot authoring interactivity, and overall usability of our system, we invited 12 users (11 male, age ranging from 19 to 31) to our three-session preliminary user study, with 10 of them from engineering background and the other 2 from management background. Eight
users had VR experience while five had AR experience. None of the users had prior experiences with our system. The study was conducted in a 5mx5m area using only virtual robots (the GripperBot and the Drone) for safety concerns. The study for each user cost about 2 hours cumulatively and each user was paid 20 dollars for compensation. The entire process was video recorded for post-study analysis. Each user was given a 15 min tutorial about the background of the project before proceeding to the task in session 1. After each session, each user was given a survey to answer objective Likert-type questions. Each Likert-type item is graded by users from 1 to 5, on the usefulness of the feature and the level of agreement. After all the sessions, a conversation-style interview was conducted to acquire subjective feedback and a standard System Usability Scale (SUS) questionnaire was also given to each user. (P = participant)

Session 1: Human Authoring and Motion Mapping
One of the core features of GhostAR is to recognize the user’s body gestures and map it with the previous authoring to output the corresponding robot behavior. This is achieved by our in-situ generated collaboration model using DTW based algorithm. The first session of the study is designed to evaluate this with novice users.

Procedure. Users were asked to perform a continuous motion in the Human Authoring Mode that included six regular gestures (Figure 7-(1)): stand up from a chair ($G_1$), wave hand ($G_2$), pick up a virtual item ($G_3$), walk to another place and put down the virtual item ($G_4$), bow and reach out to the handles of a chair ($G_5$), push the chair a short distance and stand up straight ($G_6$). The whole motion series took approximately 30 seconds. The users then forwarded into the Observation Mode and put each of the above gesture into a Trigger Group $T_i$ ($i = 1, \ldots, 6$). Also, the object-moving motion between $G_3$ and $G_4$, and the chair pushing motion between $G_5$ and $G_6$ are Grouped as two Synchronize tasks $S_i$ ($i = 1, 2$), respectively. Each user repeated the above process 4 times and all data set were recorded for a cross validation: using 1 set of data as authoring and 1 set as acting, to acquire large amount of evaluation results. For each Trigger task $T_i$, we collected the detection time from the collaboration model, $t_{T_i}$. For each Synchronize task $S_i$, we collect the estimated progress at time $t$ by the collaboration model instead, noted as $P_{est}^S(t)$. The end time, $t_{T_i}$ of each Trigger gesture $G_i$, as well as the start time $t_{S_i}^{\text{start}}$ and end time $t_{S_i}^{\text{end}}$ of each Synchronize task $S_i$ were manually labeled as ground truth.

Evaluation of Trigger task detection accuracy. Figure 8-(top) shows an example of the DTW distance values of a user (P4) in the Action Mode. All 12 users authored 846 valid Trigger tasks in total (6 gestures × 12 comparisons × 12 users),
840 of which were successfully detected (99.3%). For the 840 detected Trigger tasks, we calculated the error of detection time |t_G − t_T| and display its distribution in Figure 8-(Bottom). The average of the detection error is 414.8 ms (SD, 1052.2 ms) and 803 Trigger tasks (95.6%) were detected within 1 second before or after the user had completed that gesture. To better illustrate the Trigger detection accuracy that is associated with different gestures, we calculate the 80% medians, which reveals the time within which most (> 80%) of the triggers were detected: G1: 375 ms, G2: 358 ms, G3: 857 ms, G4: 685 ms, G5: 642 ms, G6: 517 ms. We observed that the accuracy of detecting the pick-up (G1) and put-down (G4) gesture was lower than that of the stand-up (G3) and wave-hand (G2). This is because of the motion involved in G3 and G4 has less amplitude, with only one hand moving in the relatively smaller distance, resulting in lower detection accuracy.

Evaluation of Synchronize task progress estimation. We used the timestamp values t to characterize a user’s progress in the Synchronize task. The actual progress is defined as \( P_{\text{act}}(t) = \frac{(t - t_{\text{start}})}{(t_{\text{end}} - t_{\text{start}})} \) \((t_{\text{start}} < t < t_{\text{end}})\). Figure 9-(Left) shows an example of the \( P_{\text{act}}(t) \) curve. For each Synchronize task \( S_i \), we uniformly selected 100 data points from the \( P_{\text{act}}(t) - P_{\text{act}}(t) \) curve \( (P_{\text{act}}(t) = 1\% , 2\% , \ldots , 100\%) \) and calculate the estimation error \( |P_{\text{act}}(t) - P_{\text{act}}(t)| \). All 12 users contributed 14400 data points (2 Synchronize tasks × 100 data points × 6 comparison × 12 users) in total. The distributions of the estimation errors are shown in Figure 9-(Right). The average of the progress estimation error is 7.31% (SD, 7.61%). The 80% medians are 12.24% (object moving) and 11.73% (chair pushing), which implies that in most of the time (> 80%), the robot will not surpass or fall behind a user for more than 1 second considering the fact that the Synchronized tasks last between 4 to 7 seconds long. Based on our observation during the user study, we suspect that the error may come from the minor inconsistency (e.g. irregular pause) of the user’s behavior during some of the motions.

Session 2: Robot Authoring Interactivity

Another highlighted feature of GhostAR is to author spatially and temporally synchronized robot motion with the human reference. In this session, we tested the robot interactivity and system interface towards authoring a Synchronize HRC task.

Procedure. A user first defined the human motion ghost by traveling through two routes: a straight-line and a circular path within the 5mx5m arena. Then we asked the user to author two virtual robots to travel alone with the human ghost while trying to coincide with the footprint (for the GripperBot) and the head position (for the Drone) of the human ghost, as illustrated in Figure 7-(2). The authoring data was recorded for accuracy analysis, and each user repeated the process twice.

Result and Discussion. In general, users were able to understand the robot authoring interaction quickly, and all users successfully authored the described task. Many users frequently use the “pause/resume” feature to adjust themselves for better observing and maneuvering perspective during the authoring. The histogram in Figure 10-(Top) shows the distributions of the robot authoring errors. The average of authoring errors are 13.9 cm (SD, 9.2 cm) for the GripperBot moving along straight line, 16.0 cm (SD, 8.2 cm) for the Drone moving along straight line, 16.1 cm (SD, 8.7 cm) for the GripperBot moving along a circle and 17.5 cm (SD, 9.3 cm) for the drone moving along a circle. Since the users used the robot’s body as reference, and the GripperBot and the Drone both have a radius of 25 cm, we consider that the human ghost and the robot are aligned if the captured distance is shorter than 25 cm. Based on these criteria, we calculated an alignment rate which is defined by the percentage of errors which are smaller than 25 cm. The values of alignment rate are 89.57% (the GripperBot following a line), 86.87% (the Drone following a line), 84.29% (the GripperBot followed a circle) and 81.46% (the Drone following a circle). This result indicates that most of the time (> 80%), the users were able to author the robot to be precisely aligned with the human ghost for this Synchronize task.

By observing the study and analyzing the results, we find that keeping the error below 10 cm was generally a hard task for regular users, especially for the Drone which has one added DoF than the GripperBot. We believe this is mainly because the users were not familiar with the kinetic mechanism of the robots. Restricted by the physical principals, the robots had large inertia and could not strictly follow the users' authoring behaviors as assumed. So that many users tended to overshoot while controlling the robots. Additionally, the Drone is always swinging due to its aerodynamics properties (simulated by ROS-Gazebo), which makes it even harder for maneuvering. Besides, the circular route evidently produced more error than the straight-line, which we assume is caused by the lack of next-position reference and users could not anticipate the time when the Ghost made a turn. We also compare the novice users with an experienced user who had practiced the authoring process five times. And display their average error in Figure 10-(Bottom). The result shows that the experienced user achieved much better accuracy result than the novice users. This indicates that the proposed robot interaction can be
Session 4B: Human-Robot Interaction

easily mastered with a few rounds of practice, and therefore better Synchronize performance can be achieved.

Session 3: System Usability Evaluation

Here we evaluated the overall usability of our system by asking users to author an HRC task, then act out the collaboration.

Procedure. The users were asked to complete a joint assembly task with the GripperBot, during which the user and the robot each picked up one part and met in the middle to put them together. The HRC task consists of a Synchronize action and two Trigger actions. As illustrated in Figure 7-(3), the collaboration scenario is described as following: the users picked up his green part, Triggering the robot to pick up the red part; then they traveled towards the middle workstation at a Synchronize pace; when met, the users put down their parts first, Triggering the robot to place its red object and complete the assembly.

Result and Discussion. All participants were able to successfully act out the collaboration task with our system issuing the correct robot behavior according to the authoring. The average task authoring time for task completion is 2 min 16 s.

The system feature related Likert-type results collected from the 3-session study are shown in Figure 11. After the tutorial, participants were generally confident to author the HRC task and agreed on the smoothness of our system workflow (Q9: avg = 4.25, sd = 0.62). “It’s fast and easy to plan a task, just role-plays your action and use the ghost reference to play the robot part. (P2)” The timely authoring process and rapid iteration were appreciated by the users. “I like how fast it is from planning the task to acting it out, encourages me to try more. (P4)” We believe the feedbacks indicate that our system enables real-time and in-situ authoring, meeting our DG5. Users are also impressed with the motion mapping accuracy and robustness of our system during the Action Mode. “I thought my acting was not that consistent with multiple pauses, but surprisingly your system recognized it and issues the correct robot behaviors. (P3)” This comment indicates that we have achieved robot collaborative adaption in terms of coping with human partner’s uncertainty (DG1).

The embodied authoring and interaction method (referred to as ‘role-playing’) is receptive to our participants, for both human ghost authoring (Q1: avg = 4.17, sd = 0.94) and robot avatar control (Q7: avg = 4.08, sd = 0.79). “Moving a virtual robot in AR space was much easier than I thought. (P4)” These comments have reflected positively to our DG2. The visualization accuracy of the ghost in terms of time-space reference is high according to (Q3: avg = 4.5, sd = 0.67). Further, the realistic robot simulation used for robot avatar interaction and visualization is also generally appreciated (Q6: avg = 3.83, sd = 0.94). “That drone was kind of difficult to control. But I think the interaction method you provide is super realistic. The robot didn’t move to where you were pointing to, it moved slowly to the target like a real robot. And for the drone, it was swinging and tilting when moving. (P7)” We believe these comments confirm the necessity of adopting a professional robotics engine (ROS-Gazebo) with realistic simulation to enhance the experience, meeting our DG4.

DISCUSSION AND FUTURE WORK

While users all appreciated the usefulness of AR ghost in terms of contextual visualization and task simulation, they have almost unanimously raised one interestingly conflicting problem. 6 out of 12 users have mentioned in one way or another that, the AR ghosts can occasionally become distracting and obtrusive. “There are too many ghosts in front of me when I am trying to see and act. (P10)” This feedback emerges after the users get familiar with the system and they start feeling not needing the AR guidance all the time. This finding brings out an important question when designing such systems: how shall we balance between demonstrative ghost reference and clear authoring view, and provide both for the user? While this may be a research question for the future endeavor, we have some initial thoughts. A quick fix could be giving the user the ability to toggle all the AR ghost manually. While this may be a research question for the future endeavor, we have some initial thoughts. A quick fix could be giving the user the ability to toggle all the AR ghost manually. However, if the user only wants to hide some of the ghost images, the added interaction could increase the cognitive load of the user. Another potential solution involves intelligently detecting the user’s intention and only display the most relevant and needed ghost. For example, during the Action Mode, the ghost appears only when the user is about to go off-track. In this work, we prototyped our system with see-through HMD AR and achieved body externalization with IR-based tracking.

Survey responses were positive about the AR ghost to display the authored task in a spatially situated manner (Q4: avg = 4.5, sd = 0.52) with intuitive visual representation (Q5: avg = 4, sd = 0.85). The ghost images are welcomed as a time-space context for authoring collaborative robot task (Q8: avg = 4.33, sd = 0.78), as well as a visual guidance during the Action Mode for successful collaboration execution (Q11: avg = 4.08, sd = 1.24). “It’s very interesting like Sci-Fi, when I’m able to see what I have done with ghosts. (P3)” The most popular feature of our system is the animation preview for the newly authored ghost (Q2: avg = 4.25, sd = 0.62) and the entire HRC task before action (Q10: avg = 4.58, sd = 0.51). “The ghost animation is definitely my favorite part of the system, I can see so many potential applications for this technique. (P12)” We believe the feedback matches our goal of providing contextual aware authoring experience (DG3). The standard SUS survey result for the entire study is 80 with a standard deviation of 6.75, indicating high usability of the system.
device. The current hardware setup provides only 3-joints tracking (head and two hands), and we utilized only the position value, resulting in a 9-dimensional input data for our collaboration model. Note that this setup is largely limited by the currently available hardware platform, and is likely to change. For example, future AR-based body tracking technique is expected to have multiple-joints and provides more realistic humanoid ghost. Furthermore, with additional sensory input embedded, such as tactile force feedback, we can achieve force-sensitive collaborative authoring with our system, such as joint object carrying.

Although the GhostAR system can detect the user’s motion status with fair accuracy, the DTW algorithm we are currently using largely relies on user’s consistency in order to achieve satisfying performance. As a result, the user in Action Mode is constrained to the previously authored motions and has very limited flexibility. To tackle this problem in the future, our initial guess could be utilizing the state-of-the-art human action recognition approaches, such as probabilistic methods and deep neural networks, to capture the critical features in the user’s motion. Thus granting more freedom to the user and enabling for intuitive authoring and acting behavior while maintaining collaborative accuracy.

It is worth emphasizing that GhostAR is an HRC task authoring and acting platform designed as a complimenting workflow for the more advanced human-robot-collaborative learning frameworks, as discussed in the Related Work section. Our system can be applied to many other HRC models specializing in different applications, to achieve a higher level of collaborative intelligence while empowering users with real-time, spatially situated visual task authoring capability.

CONCLUSION
We have presented GhostAR, a human-robot-collaborative task authoring system featuring role-playing embodied interaction and contextually situated visual editing. In this paper, we have demonstrated how an AR interface can be synergistically integrated with embodied authoring to create elevated HRC experience. We have proposed essential guidelines for HRC authoring system design, highlighting 1) robust motion adaptation, 2) natural embodied interaction, 3) contextual authoring reference, 4) realistic visual simulation, and 5) fluid real-time iteration. Our three-session system evaluation received positive results, indicating that the proposed system has reached the design goals, while also unveiling the potential directions for future endeavors. GhostAR has created a brand new perspective to solve the balancing problem between sophisticated functionality and intuitive interaction in an adaptive collaboration context, thus offering future inspirations to the HCI and HRI community.

ACKNOWLEDGEMENT
This work was partially supported by the NSF under grants FW-HTF 1839971, IIS (NR) 1637961 and IIP (PFI:BIC) 1632154. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agency.

REFERENCES


Session 4B: Human-Robot Interaction


