2021 18th International Conference on Ubiquitous Robots (UR)
July 12-14, 2021. Gangneung-si, Gangwon-do, Korea

2021 18th International Conference on Ubiquitous Robots (UR) | 978-1-6654-3899-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/UR52253.2021.9494681

A Robot Capable of Proactive Assistance through Handovers for Sequential Tasks

Nayoung Oh, Junyong Park, Ji Ho Kwak and Sungho Jo

***Abstract*— For robots to coexist with humans in diverse situations, their ability to fluently interact with humans becomes important. One important aspect of interacting with humans is being able to understand what the humans are doing to provide appropriate forms of assistance. Previous works used information from hands and objects to understand the human behavior and its context. However, as environments, tasks, and interaction targets become more complex, it becomes difficult to design assistance rules that can cover the variety of situations with such simple reasoning methods. Therefore, we develop a robotic system that combines action recognition with an activity-level knowledge bank to assist a human performing a sequential activity. The system maps the detected action to objects related to the task using the knowledge bank and delivers the objects to the human through handover. To evaluate the performance of our system, we conduct comparative experiments with two other simple systems: command-initiated and random-trial. Through experiments on two cooking tasks, our system is compared to the two simple systems on the basis of human idle time and object idle time. Results show that our system leads to the shortest human idle time. The object idle time of our robot system is similar to the command-initiated system and much shorter than the random-trial system. We conclude that robots that understand human actions can more efficiently assist humans to accomplish their tasks.**

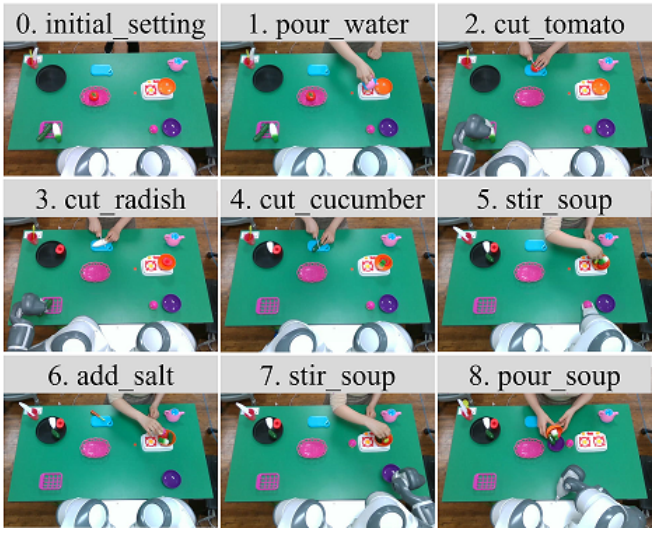

Fig. 1. Brief overview of the main robot system. The robot picks up and hands over objects based on the recognized action and the knowledge about the activity (in the photo 2, 3, 5, and 7)

I. INTRODUCTION

Robots are not just science fiction characters. For many decades, robots have mainly been used for repetitive tasks in static environments such as factories. In recent years, however, robots have been brought into various situations with humans. Humans have started to use robots in environments such as restaurants and cafes, where close interaction with people is required. In such situations, robots cannot only do what they are preassigned to do because human behavior is sometimes unpredictable and their needs continuously change. For example, for a chef in the kitchen, rather than a robot that always performs the same task of delivering kitchen tools, a robot that recognizes the chef's current needs and provides the appropriate assistance would be of greater help. In other words, robots need to understand both the environment and the human behavior to help humans seamlessly.

There are works in the HRI community that aim to design such smarter assistance robots. Some use object or hand detection algorithms in order to understand human activities when providing assistance [1], [2], [3]. Others have additionally utilized speech recognition techniques to provide the robot with a richer context [4]. However, as environments, tasks, and interaction targets become more complex, it becomes difficult to design rules that can cover the variety of situations with such simple detection methods. Instead, recognizing human actions on the whole and determining the human's need from them using general knowledge about the activity can resolve such difficulties. Brooks et al. use action recognition so that the robot can understand which action the human is performing [5]. Their work addresses discrete tasks where the robot is required to provide assistance only once. Many complex tasks such as cooking or furniture assembling, however, consist of multiple sequential subtasks. Thus, robots should be able to not only recognize multiple actions but also provide different types of assistance.

In this work, we develop a robot system that provides assistance to humans performing complex, sequential tasks. Our system consists of an action recognition module, an assistance selection module, and an execution module. The robot first recognizes the human's current action through the action recognition module. Then it performs the appropriate assistance in relation to the recognized action using general knowledge about the human activity. Without additional rules such as the specific order of actions the robot needs to

This work was supported in part by the National Research Foundation of Korea Grant funded by the Korean Government (MSIT) under Grant NRF-2016R1A5A1938472 and in part by the Technology Innovation Program funded by the Ministry of Trade, Industry & Energy (MI, Korea) under Grant 10070171.

The authors are with the School of Computing, Korea Advanced Institute of Science and Technology, Republic of Korea. {lightsalt, jyp0802, jim9611, shjo}@kaist.ac.kr

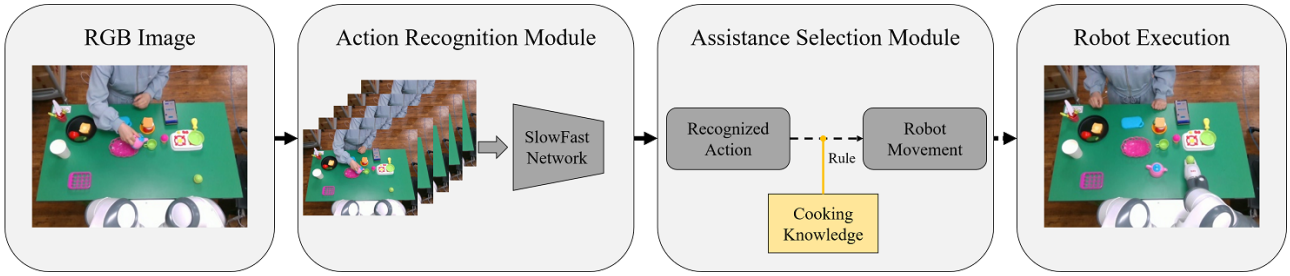

Fig. 2. Overview of the robot system

perform, the system uses only the activity goal and the information of each human action to select the appropriate actions. Actions recognized by our robot system and the executions of assistance in regard to the recognized actions are illustrated in Fig 1.

To evaluate how helpful our proposed system is, we compare our system with two other systems. The first is a command-initiated system in which the robot performs the actions that the human commands. The second is a random-trial system in which the robot consistently performs actions randomly. Through experiments with N=12 people on two daily life tasks, we compare our system to the other two systems in regard to the human idle time, i.e. time waiting for the robot [2], [6], and the object idle time, i.e. time for which objects are left unused. Furthermore, through a Likert scale survey on user satisfaction, we showed that users prefer our system especially in the aspect of interaction efficiency to the random-trial system.

The results show that our system greatly reduces the time that the human has to wait compared to the other two systems. The time that objects are left unused is also greatly shorter than the random-trial system, and similar to the command-initiated system which already yields optimal object idle times since the humans are waiting for the requested object. These results support our assumption that robots should be able to understand human actions in order to help humans more efficiently.

II. RELATED WORK

In the field of human robot collaboration, two ways existing research uses to make robots understand human actions are rule based methods and data driven methods. Rule based vision includes tracking of object positions [1], [2], hands [3], [7], [8], [9], the upper body [10], or the total body [4].

Baraglia et al. [2] focus on the problem of when to help by comparing three modes: human-initiated help, robot-initiated reactive help, and robot-initiated proactive help. To understand the current task, the robot recognizes the location objects through an RGBD camera. The research concludes that human-initiated mode is best for social interaction, and robot-initiated proactive mode achieves better team fluency. Casalino et al. [9] research on enhancing the efficiency of collaboration in respect to the robot idle time through predicting human actions based on hand position. Other research using hand tracking focuses on how a human and a robot collaborate to use common objects while avoiding collisions by the robot asking the human whether it can pick up the object [8]. Park et al. [10] use the position of the upper body to deal with the collision issue itself as a research topic to build a better motion planner for collaboration with humans. Liu et al. [4] use human tracking data and speech data as interaction histories which are input data for the robot to learn how to help.

Data-driven vision includes action recognition for the robot to understand current status of humans. Brooks et al. [5] use action recognition to decide how to help without human command. They measure proactive assistance against command-driven assistance and find out that users prefer proactive assistance. Some research uses action recognition even for deciding when to help [11].

The most relevant research is done by Braglia et al. and Brooks et al. Baraglia et al. compare three modes to decide when to help; however, the system estimates human actions based on object positions. We assume that this estimation works well since the tasks are moving objects. On the other hand, our experiment sets cooking as a task which requires understanding of more complex actions; thus, we use action recognition and link the detected actions with general knowledge about activities instead of using a rule-based vision system. Our research is similar to the research of Brooks et al. in terms of using action recognition. However, they concentrate on assembly tasks and connect one task to one robot assistance. Because daily life tasks consist of diverse subtasks, we make a task contain various types of actions.

III. METHOD

A. Overview of Our Main System

The overview of our main system is as in Fig 2. The system is composed of three modules: action recognition, assistance selection, and execution. The action recognition module takes in a sequence of RGB images from the camera and performs action recognition. The outputs of this module are possibilities for each action, which are then delivered to the decision making process. Through simple rules based on human actions, the next module decides whether or not the robot has to help. If the decision is to help, it determines the specific action of the robot. Finally, the robot moves based on

predetermined paths to pick up and place objects. Detailed explanations of each part are as follows.

Action Recognition We use SlowFast network to detect human actions from videos [12], [13]. Specifically, we sample 32 frames with a sampling rate of 2 and set the ResNet depth, alpha, and beta inverse to 50, 4, and 8, respectively. For training, the Kinetics model provided by Facebook Research is used as a pre-trained model.

The training data is collected by using the command-initiated system in order to collect data both with and without a moving robot. The videos used for training are categorized into two groups. In one, a human controls the robot using the app to send commands. In the other, another person who is not working controls the robot for proactive assistance. The camera view includes both the task space with objects and the upper body of the human. Videos taken as 640 X 480 size are resized to 256 X 341 size and used for training. The total length of the video data collected is about 30 minutes.

The labels of the videos describe human actions such as cutting a tomato or stirring soup. For higher accuracy, the duration of action does not contain pre- or post-actions such as picking a tomato or placing cut pieces onto a plate.

For real time action detection, we use a frame buffer so that half of the frames are identical to the previous video. The network outputs consist of probabilities of each action and only the most probable action is considered as the predicted action. If the probability of the predicted action is more than 72%, the action is passed on to the assistance selection module; if not, the predicted action is treated as 'None'.

Assistance Selection Humans are capable of helping others even when they do not know exactly what task the others are doing. When designing assistance robots that help humans in similar ways, it is difficult and often inefficient to give the complete information of all tasks due to the diversity of possible task executions. Activities that consist of sequential tasks, such as cooking or assembling furniture, can be divided into ordered stages that are made up of multiple related tasks. For example, the tasks that make up a cooking activity can be divided into the stages 'preparing ingredients', 'combining food', 'tasting and adjusting', and so on. In such cases, knowing which tasks and objects are related to which stage may be sufficient to assist the activity.

In our system, we provide the robot with a cooking knowledge bank that contains the necessary information about cooking activities in order to assist the activity. Specifically, the cooking knowledge bank maps each human action to a specific stage of the activity, and contains a list of objects that are related to each stage. For example, for the stage of 'preparing ingredients', the knowledge bank maps the ingredients 'bread', 'tomato', and 'cucumber' as ingredients for a sandwich.

Using the cooking knowledge bank, the robot can provide an appropriate assistance to the human performing a specific stage of the activity. The robot first detects which stage the human is at through the action recognition module. The detected action is then mapped to a specific stage by the knowledge bank. Then, knowing the possible objects that it can handover to the human, the robot chooses an object that belongs to that particular stage. When multiple objects are feasible to the stage, the robot chooses one according to a predefined order. In order to provide the object to the human as soon as possible, the robot acts immediately once a possible object is proposed.

Fig. 3. A photo of the robot and the task space

B. Basic Platform Info

ABB yumi robot which has two arms with 7 axes each is used for the system. RAPID server is installed on ABB yumi and used to plan paths and control joints. The type and destination of movement are sent by Yumipy python interface.

Realsense D435i is used to collect RGB images for action recognition. This camera is also used for the random-trial system to detect whether the human uses the object. The camera is fixed on ABB yumi to film the human and the task space together.

In front of ABB yumi, a table is set with kitchen toys. Since the maximum weight that an arm of yumi can lift is 0.5kg [14], we use toys instead of real kitchen tools. Fig 3 shows the setup of our robot for the experiments.

C. Task Representation

We choose cooking as our target task since it is a daily activity that contains various actions. Furthermore, a person can naturally ask for help when they are cooking, in situations such as when they need to hold a hot pot or need to keep stirring soup. Thus we set two cooking tasks: Task A for preparing dinner and Task B for preparing breakfast. During each task, the robot helps the human four times. The robot picks up an object and places it where the human can easily reach. For instance, while the human is stirring soup, the robot picks up a bowl and places it next to the gas stove. Then, the human can hold the hot pot with two hands and safely pour the soup into the bowl.

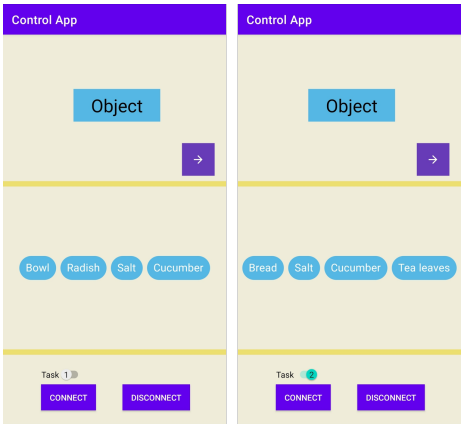

Fig. 4. The control app design, the left side is for task A and the right side is for task B

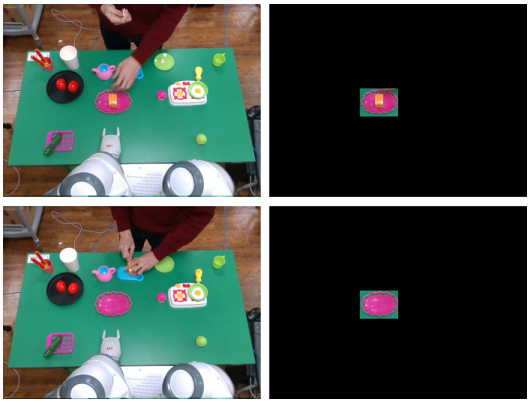

Fig. 5. Right top: Right after the robot places the bread on the tray, Left top: Masked area of the tray right after placement, Right bottom: After waiting for 5 seconds, Left bottom: Masked area of the tray after 5 seconds

D. Implementation of Comparison Systems

To evaluate how well our system performs, we develop two other systems for comparison. For the first comparison system, we used a command-initiated system similar to the command-driven assistant in [5]. In this system, the human decides when and how the robot will help. To make the interaction simple and robust an Android app, as shown in Fig 4, is used to choose which object the robot will give. The chosen object's label is sent through socket transport to a computer that is connected to ABB yumi. Then, the robot moves to pick up the requested object following the same predetermined path as our main system.

The other comparison system is random-trial system where we assume that the robot does not understand anything about the humans' activities. The system chooses a random object to give from the list of unused objects using the python random module. Then, the robot picks up the object, places it, and takes a photo of the task space. After waiting for 5 seconds, the system takes another photo. With a mask to consider only the space nearby the placed object, histograms of h channels in each hsv photo are calculated, and are compared using Bhattacharyya distance. Examples of photos used in this system are shown in Fig 5. If the similarity is high, we assume that the human does not pick up the object, then the system takes the object again to place it where it was. If not, the robot goes to its home position. Then, this object is removed from the unused object list. After waiting for 2 seconds, the system repeats this process until the unused object list becomes empty. The robot movement is the same as others except the robot places the object, waits for the human, and moves depending on human action.

IV. EXPERIMENTS

Our study was completed with 12 participants. Participants conducted task A and task B with different systems. For instance, one participant performed task A with our main system (action recognition system) and task B with the first comparison system (command-initiated system). Therefore, 4 data for each specific task and system were collected. We required the participants to treat the kitchen toys as real ingredients and cooking equipment and to spend 5∼10 seconds on each action. However, we did not use a timer to avoid explicitly setting the timing of robot assistance. Then, we requested them to read a recipe in which the objects that will be given by the robot were written in red.

After both task A and B were completed, we surveyed the participants on their user satisfaction of each system. The survey was conducted on a Likert scale of 1 to 7, 7 showing highest agreement. The 7 sentences that we used are as follows:

- The robot could accurately perceive the situation.
- The robot accomplished the right task at the right time.
- I will use the robot again.
- The robot was helpful.
- The actions of the robot were distracting. (reverse scale)
- The robot was easy enough to use.
- The robot and I collaborated efficiently together.

We recorded the experiment procedure and measured two objective factors. The first factor is human idle time as in [2]. Human idle time is the time gap between the time a human finishes the action prior to robot assistance and the time they reach out for the object delivered by the robot. The second factor is object idle time. This is the gap between the time the robot places an object in its designated space and the time the human grabs the object. For the second comparison system (random-trial), object idle time is summed up for each object. The timeline figure Fig 6 shows how the human idle time and object idle time are measured.

V. RESULTS

A. Objective Metrics

The duration of all four actions was each measured and analyzed using one-way ANOVA. We used the alpha value as $\alpha = 0.5$; thus, $F_{crit} = 3.2093$ for task A and $F_{crit} = 3.2145$ for task B. For meaningful comparison, the results of three subtasks (one from task A and two from task B) that were incorrectly performed by the participant were excluded from our results. The results of the analysis are as Fig 7 to Fig 10.

Human Idle Time

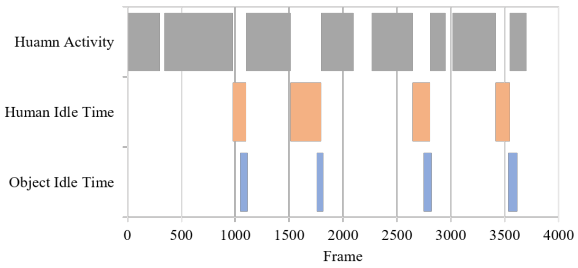

Fig. 6. Timeline example including human activities, human idle time, and object idle time

(a) Task A: $F = 5.1607$, $p < 0.05$
(b) Task B: $F = 10.3588$, $p < 0.05$

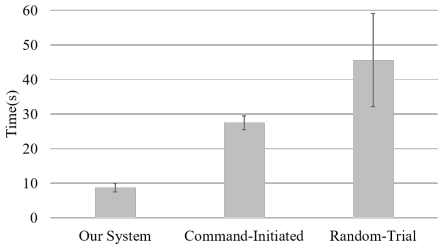

Fig. 7. Human idle time graph for task A

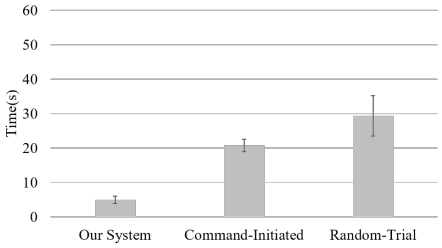

Fig. 8. Human idle time graph for task B

With our system, participants did not have to wait long for both task A and B. As shown in Fig 7 and Fig 8, the human idle time of our system was the shortest among the three systems, with participants waiting for around 5 seconds in task A and 10 seconds in task B on average for each subtask. This is significantly lower the human idle times of the command-initiated system and the random-trial system which are all greater than 20 seconds.

Object Idle Time

(a) Task A: $F = 3.3029$, $p < 0.05$
(b) Task B: $F = 11.1239$, $p < 0.05$

The object idle time of our system is similar to that of the command-initiated system as shown in Fig 9 and Fig 10. For both systems, objects were left unused for around 5 seconds on average for each subtask. This is lower than one third of the object idle time of the random-trial system.

B. Subjective Metrics

We used an average score of each system from the survey answers of each user as the user satisfaction score. We used

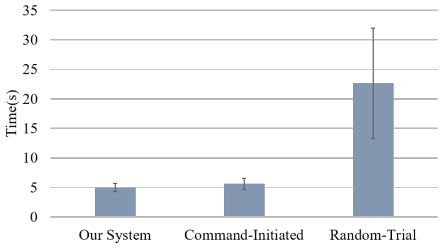

Fig. 9. Object idle time graph for task A

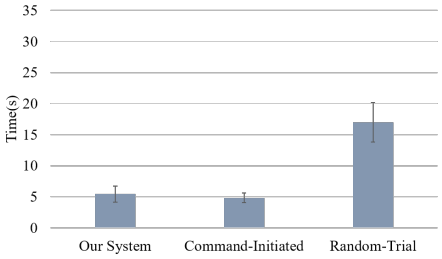

Fig. 10. Object idle time graph for task B

the alpha value as $\alpha = 0.5$; thus, $F_{crit} = 3.4668$. Then, the user satisfaction score was analyzed through A repeated-measure Analysis of variance (one-way ANOVA). After that, since users did not experience all three systems, one-tailed paired T-test ($\alpha = 0.5$) was used to compare experiences of a user who collaborated with the robot in the same systems based on the ANOVA result.

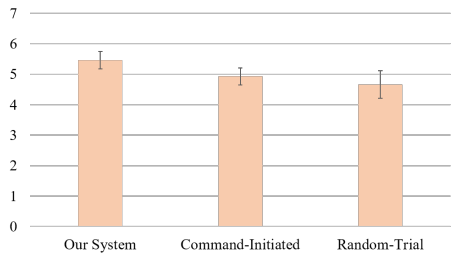

Fig. 11. User satisfaction score

The user satisfaction score was the highest for our system as shown in Fig 11. However, the p-value was higher than 0.05; $F = 1.3901$ indicating that the average scores of the three systems are not significantly different. This can be for a number of reasons. Due to the small number of participants to our experiments, the amount of data was not sufficient. Furthermore, users rated the system with their own standards causing the variance to become large. Nevertheless, the one-tailed paired T-test of the question "The robot and I collaborated efficiently together" for our system and random-trial system showed a meaningful difference as in Fig 12 : $p < 0.05$. Thus, users preferred our system to the random-trial system especially in the aspect of collaboration efficiency.

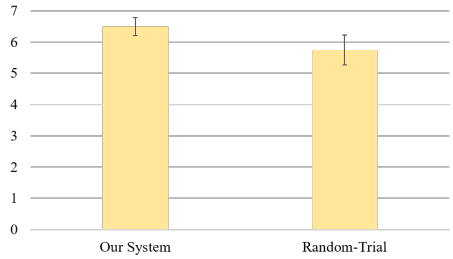

Fig. 12. User satisfaction score for collaboration efficiency: "The robot and I collaborated efficiently together"

VI. DISCUSSION

In the aspect of the human idle time, our system did not require participants to click buttons to send commands or wait for the right object to be chosen during random trials, reducing unnecessary time. Furthermore, since the robot started moving while the participants were conducting other tasks, the participants did not have to wait long until the robot finished the action. Therefore, the human idle time of our system was the shortest of three systems.

Moreover, although the robot proactively assists a human in our system, the object idle time was similar to the command-initiated system and much smaller than the random-trial system. With the command-initiated system, a human issues an order and keeps focusing on the object that the robot hands over, which is why the object idle time is optimal. Because the object idle time of our system was similar to the 'optimal' command-initiated system, it is possible to conclude that our system efficiently assisted humans.

However, because we used kitchen toys, some people finished their actions too quickly. In another case, a participant even kept cooking an egg until the robot in the random-trial system handed over salt. Thus, using real tools will be better for collecting data related to execution time of subtasks.

Furthermore, in this system, we used a simple rule-based algorithm that uses information from a knowledge bank to decide the appropriate assistance. However, such rule-based algorithms may not be able to model the diversity of human activities. Therefore, in future study, designing a more complex model that considers not only the current human action but also the context of the environment could help the robot become more robust to various situations.

Finally, one participant who used the command-initiated system issued a command before he started the action previous to the one that requires robot assistance. After a few interactions with the robot, he seemed to realize that the robot requires some time to pick up and put down an object. This kind of proactive assistance determined by humans can serve as a good reference to develop a better system.

VII. CONCLUSIONS

In this paper, we propose a robotic system that provides assistance to humans performing sequential tasks through handovers of related objects. The system first detects the human's action and then finds related objects using the knowledge bank of the activity. Through the experiment comparing our system to the command-initiated system and the random-trial system, we show that our system can effectively help humans by reducing time for which the human should wait. Furthermore, through a user satisfaction survey, we show that users preferred our system particularly to the random-trial system. We, however, use a simple rule-based algorithm with the knowledge bank to decide the appropriate assistance, where the robot delivers any object that belongs to the particular stage. A more intricate algorithm may be required to be robust against anomalies and unexpected situations. Furthermore, we require the mapping between human actions and the activity stages, which could be replaced by automated techniques such as action segmentation. Through these advances, robots will be able to better understand human activities and provide assistance in various situations robustly.

REFERENCES

[1] T. Liu, J. Wang, and M. Q.-H. Meng, "Human robot cooperation based on human intention inference," in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*. IEEE, 2014, pp. 350–355.

[2] J. Baraglia, M. Cakmak, Y. Nagai, R. P. Rao, and M. Asada, "Efficient human-robot collaboration: when should a robot take initiative?" *The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 563–579, 2017.

[3] T. Hazbar, "Task planning and execution for human robot team performing a shared task in a shared workspace," 2019.

[4] P. Liu, D. F. Glas, T. Kanda, and H. Ishiguro, "Learning proactive behavior for interactive social robots," *Autonomous Robots*, vol. 42, no. 5, pp. 1067–1085, 2018.

[5] C. Brooks, M. Atreya, and D. Szafir, "Proactive robot assistants for freeform collaborative tasks through multimodal recognition of generic subtasks," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 8567–8573.

[6] G. Hoffman, "Evaluating fluency in human–robot collaboration," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 3, pp. 209–218, 2019.

[7] D. Kim, B. B. Kang, K. B. Kim, H. Choi, J. Ha, K.-J. Cho, and S. Jo, "Eyes are faster than hands: A soft wearable robot learns user intention from the egocentric view," *Sci Robot*, vol. 4, no. 26, p. eaav2949, 2019.

[8] B. A. Anima, J. Blankenburg, M. Zagainova, M. T. Chowdhury, D. Feil-Seifer, M. Nicolescu, M. Nicolescu *et al.*, "Collaborative human-robot hierarchical task execution with an activation spreading architecture," in *International Conference on Social Robotics*. Springer, 2019, pp. 301–310.

[9] A. Casalino, A. M. Zanchettin, L. Piroddi, and P. Rocco, "Optimal scheduling of human-robot collaborative assembly operations with time petri nets," *IEEE Transactions on Automation Science and Engineering*, 2019.

[10] J. S. Park, C. Park, and D. Manocha, "Intention-aware motion planning using learning based human motion prediction." in *Robotics: Science and Systems*, 2017.

[11] A. Reneau and J. R. Wilson, "Supporting user autonomy with multimodal fusion to detect when a user needs assistance from a social robot," *arXiv preprint arXiv:2012.04078*, 2020.

[12] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6202–6211.

[13] H. Fan, Y. Li, B. Xiong, W.-Y. Lo, and C. Feichtenhofer, "Pyslowfast," https://github.com/facebookresearch/slowfast, 2020.

[14] Technical data irb 14000 yumi. [Online]. Available: https://new.abb.com/products/robotics/collaborative-robots/irb-14000-yumi/irb-14000-yumi-data