Workshop on Robotic Co-workers 4.0: Human Safety and Comfort in Human-Robot Interactive Social Environments
2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain

Virtual Barriers in Augmented Reality for Safe Human-Robot Collaboration in Manufacturing

Wesley P. Chan[1], Adnan Karim[1], Camilo Perez Quintero[1], H.F. Machiel Van der Loos[1], and Elizabeth Croft[2]

***Abstract*— We present a system for safe human-robot collaboration through the use of virtual barriers created in augmented reality (AR). In any human-robot cooperative scenario, safety is foremost important and must be guaranteed. Using a Microsoft HoloLens AR headset, our system provides two mechanisms for ensuring safety - 1) a person barrier that encapsulates and follows the user to protect the user from collision with the robot. 2) virtual barriers that can be created and placed arbitrarily by the user to protect surrounding objects or regions from the robot. We demonstrate the usefulness of our system through two case studies representing tasks in collaborative manufacturing, showing how our system achieves seamless operation through two different response behaviours to barrier collisions.**

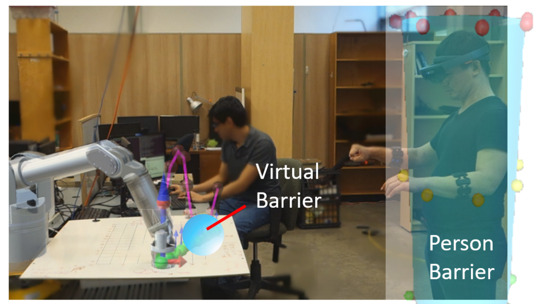

Fig. 1. Our system creates a "person barrier" around the user to protect the user by preventing the robot from colliding with the person. The user can also create "virtual barriers" at runtime and place them at arbitrary locations to protect objects and regions in the surrounding.

I. INTRODUCTION

Safety of workers in any manufacturing process is paramount. To ensure safety, most current manufacturing processes involving the use of robots utilize safety cages to isolate the robot completely from the workers during operation, and employ teach pendants for programming the robots, limiting the controls to slow joint/end effector position commands only. This programming paradigm is sufficient for repetitive processes that do not require frequent re-programming of robots. However, with the emergence of lower-cost interactive robots in the recent decade, the industry has begun to utilize robots in the manufacturing of products of smaller batch size and higher variability by utilizing human-robot collaborative teams.

With this new approach of manufacturing comes new requirements for easier ways of re-programming the robot, more intuitive methods of communicating and collaborating with the robot, and better measures for ensuring the safety of workers sharing the same workspace as the robots. Aside from the already proposed kinesthetic teaching and compliant control, the emerging technology of augmented reality (AR) provides an alternative promising method for achieving intuitive and safe human-robot cooperation.

II. RELATED WORK

With AR devices such as the Microsoft HoloLens [1], and Magic Leap [2] becoming increasingly available, researchers have explored the use of AR for various tasks including assembly [3], maintenance [4], and training [5], and achieved positive results. AR technology has shown potential for improving human-robot interactions [6] and robot programming [7]. Zaeh et al. [8] proposed a robot programming interface in which trajectories and target coordinate are projected in AR onto the robot's environment and can be manipulated interactively. Chong et al. [9] introduced a method for planning collision-free paths in AR environment. They proposed using a scalable virtual robot to offer flexibility and adaptability to different environments when an in-situ robot programming approach is desired. Green et al. [10] proposed an AR teleoperating system for mobile robots and compared it with traditional systems relying on camera feedback. They showed that their AR system yields better task completion accuracy and situational awareness.

Recently, we created a multimodal system using AR for programming robot trajectories with virtual fixtures [11]. Our study showed that our AR interface yielded shorter teaching time and lower physical workload, while maintaining comparable task performance, compared to using kinesthetic teaching and a gamepad-based interface. Building upon our previous work, in this paper, we focus on using AR for ensuring safety of human-robot collaborative tasks. We present an AR-based system for ensuring safe collaboration by allowing the user to create virtual barriers around regions of interests. These virtual barriers then protect the surrounded regions by preventing the robot from penetrating the barriers and moving into the protected regions. In the following, Section III provides a description of our system. Section IV describes two case studies demonstrating the features our system. Section V gives a discussion, and Section VI provides a conclusion and future work.

III. SYSTEM

Our system consists of two main components - a Microsoft HoloLens AR headset, and a Barrett Technologies Whole

[1]Collaborative Advanced Robotics and Intelligent Systems (CARIS) Laboratory, Mechanical Engineering, University of British Columbia.
[2]Monash University

Workshop on Robotic Co-workers 4.0: Human Safety and Comfort in Human-Robot Interactive Social Environments
2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, Madrid, Spain

Arm Manipulator (WAM) (Figure 2). The HoloLens is a the first commercially available tether free AR head-mounted display [1]. It allows the user to see virtual objects rendered over the real world. The WAM is a 7 DoF arm that we use as our test robot. Our system provides the following modules.

A. HoloLens

Virtual Robot Model A geometrically accurate model of the robot is created and collocated at the same place as the real robot. This virtual model serves as a representation of the real robot, and is used for previewing the robot's trajectory. The joint angles of the real robot is constantly sent to the virtual model, so that the virtual model always reflects the pose of the real robot. Trajectories planned using the virtual model can also be sent to the real robot for execution.

Person Barrier A cylindrical virtual barrier is created around the person. As the user wearing the HoloLens moves around, the motion tracking system of HoloLens tracks the user's position in 3D space. This position is constantly send to the person barrier such that it always follows the user. Since the user is always inside the cylindrical geometry, the user is not able to see the person barrier as HoloLens does not render the inside surface of virtual objects. Therefore, small spherical visual markers are added to the person barrier to allow the user to "see" the barrier around him/her. The person barrier is used to protect the user from the robot.

Virtual Barriers Our system allows the user to create new virtual barriers and place them at arbitrary locations during runtime. The HoloLens tracks user gaze by tracking the device orientation. A ray is traced out in the direction of the user's gaze, and a red ring is rendered at the location where the ray hits the surrounding objects or the work surface in front of the robot. When the user says the command "barrier", a new barrier is created and placed at the gaze location. These virtual barriers are used to protect surrounding objects or areas specified by the user.

Collision Detector The collision detector module computes any collision between the robot model and the person barrier or virtual barriers. When a collision is detected, the flag "isCollision" is set to true and sent to the *trajectory modifier* of the robot. Else, the flag is set to false. The *trajectory modifier* then modifies the robot's trajectory as described in the subsection below.

B. WAM

Virtual Fixture Trajectory Generator We use the trajectory generator described in [12]. The trajectory generator allows the user to create a path (a virtual fixture) through specifying a set of way points. The robot's end effector is allowed to move along the tangential direction of the path freely, but virtual springs and dampers keep the end effector on the path. Let $\boldsymbol{x}$ be the end effector location, and $\boldsymbol{x}_d$ be the closest point on the trajectory. A restoring force F_s is applied in the direction of $\hat{\boldsymbol{s}} = \hat{\boldsymbol{n}} \times \hat{\boldsymbol{t}}$, where $\hat{\boldsymbol{n}}$ is the normal

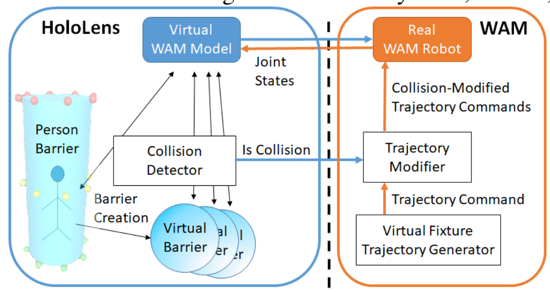

Fig. 2. Our system consists of the two main components - the HoloLens and the WAM robot. The system provides several modules working together including virtual robot model, person barrier, virtual barriers, collision detector, virtual fixture trajectory generator, and trajectory modifier.

unit vector and $\hat{\boldsymbol{t}}$ is the tangential unit vector of the path at point $\boldsymbol{x}_d$. F_s is calculated as

$$F_s = K_{p_p}\left((\boldsymbol{x} - \boldsymbol{x}_d) \cdot \hat{\boldsymbol{s}}\right) + K_{d_p}(\dot{\boldsymbol{x}} \cdot \hat{\boldsymbol{s}}) \quad (1)$$

where K_p and K_d are the proportional and derivative gains. The user can apply a force F_t in the tangential direction $\hat{\boldsymbol{t}}$ to drive the robot along the trajectory, and a force F_n in the downward normal direction $-\hat{\boldsymbol{n}}$ to exert a force onto the work surface. Thus, the overall force applied to the end effector is

$$\boldsymbol{F} = F_s \cdot \hat{\boldsymbol{s}} + F_t \cdot \hat{\boldsymbol{t}} - F_n \cdot \hat{\boldsymbol{n}} \quad (2)$$

Trajectory Modifier Trajectory commands from the trajectory generator is sent to the trajectory modifier, which listens to the "isCollision" flag. If the flag is false, the trajectory commands are passed onto the real robot for execution as-is. If the flag is true, indicating the robot is in collision with a virtual barrier, then the trajectory command is modified depending on the selected collision response behaviour. We implement two collision response behaviours.

1) **Stop on collision**: The trajectory modifier stops the robot trajectory by setting $F_t = F_n = 0$.
2) **Divert on collision**: A force vector pointing away from the barrier is applied to the end effector to push it around the barrier. In our current case we use a flat, leveled work surface, and we set the diverting force to F_n plus a small constant F_d in the direction of the surface normal $\hat{\boldsymbol{n}}$. Hence, the modified force on the end effector $\boldsymbol{F}_{mod}$ is equal to

$$\boldsymbol{F}_{mod} = \boldsymbol{F} + (F_n + F_d) \cdot \hat{\boldsymbol{n}} = F_s \cdot \hat{\boldsymbol{s}} + F_t \cdot \hat{\boldsymbol{t}} + F_d \cdot \hat{\boldsymbol{n}} \quad (3)$$

Using our system, a user working in a collaborative manufacturing setting is protected by the person barrier when he/she moves around in the same workspace as the robot. The user can also place additional virtual barriers at arbitrary locations to protect tools and workpieces in the surrounding. To demonstrate the capabilities of our system, we perform two case studies as described below.

IV. CASE STUDY

We discuss two example scenarios where the use of virtual barriers play a vital role in achieving safe human-robot

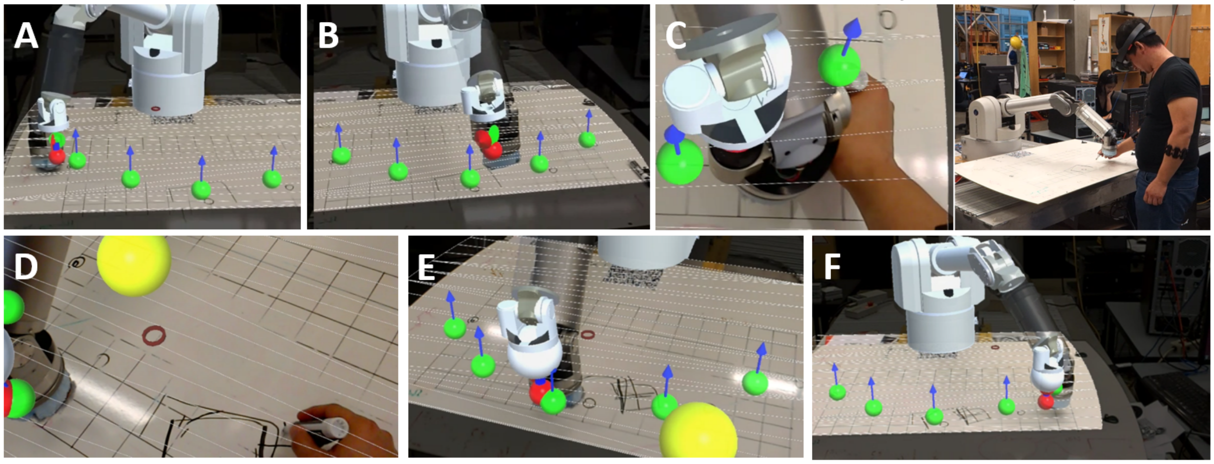

Fig. 3. User demonstrating the person barrier's functionality. A - user specifies a trajectory by setting key points (green spheres). B - Robot moves along user-defined trajectory. C - Without the person barrier, the robot collides with the worker coming into its workspace. D - With the person barrier (yellow spherical visual marker shows how far the barrier extends), the robot stops when a collision with the virtual geometry is detected. E - Once the person backs away from the robot and the person barrier is no longer colliding with the robot, the robot resumes its trajectory. F - Robot completes the trajectory by moving all the way to the right, and thus, completing its task.

collaboration in manufacturing settings [1]

A. *Person Barrier*

Example one looks into the case where a human worker shares the same workspace as the robot. This would obviously be a safety issue as a robot can seriously injure a worker if it were to collide with the worker. Our goal is to make use of our virtual barrier system to protect the worker from collisions with the robot. To achieve this, we create a transparent cylindrical virtual object in AR (the "person barrier"), which follows the user (or the HoloLens). Our setup involves a flat work surface located between the user and the robot (Figure 1). Figure 3 shows the user's view through the HoloLens during the demonstration. In this example, the user first defines a trajectory for the WAM to follow using the AR-based programming system we previously created [11] (Figure 3A). The user sets the trajectory way points (greens spheres with blue normal arrows) using gaze and speech. A trajectory is then create using these way points and the robot is constraint to moving back and forth along the trajectory (Figure 3B). When the user comes into the trajectory of the robot to work in the same workspace, without the person barrier, the robot moves into the user and hits the user (Figure 3C). However, with the person barrier enabled, the robot stops when it comes into contact with the worker's virtual cylindrical barrier (Figure 3D). Once the user has done his work and moves away from the workspace, the robot automatically resumes its trajectory moving to the right (Figure 3E). Finally, the robot completes its task by finishing the trajectory, moving all the way to the right (Figure 3F).

B. *Virtual Barrier*

Example two looks into the case where the user would like to protect a certain object (or many objects) in the work space of the robot. For example, a worker may want to protect tools and workpieces that are in the robot's workspace. Similar to the first example, our goal is to use our virtual barrier system, but this time to protect the object of interest (Figure 4). Without the virtual barrier system, if the object of interest falls on the desired trajectory of the WAM (Figure 4A), it will hit the object (Figure 4B). Unless the user is paying attention and takes action to stop the robot, it will continue in its trajectory, pushing into the object, potentially damaging the object or itself (Figure 4C). To allow the user to project objects or regions in the surrounding, our system enables the user to set virtual barriers through gaze and speech. The user simply looks at a location on the work surface. The work surface geometry is given to the HoloLens, and the intersection of the user's gaze with the work surface is computed and visualized by rendering a red ring at that location (Figure 4D). By saying the voice command "barrier", a virtual barrier is created at the gaze location (Figure 4E). Once the virtual barrier is placed, when the robot moving in its trajectory comes into contact with the barrier, it is diverted away from the object by the barrier (Figure 4F), and it moves around the barrier (Figure 4G). Once it has avoided and gone around the object, the robot then continues on its desired task trajectory (Figure 4H).

V. DISCUSSION

Worker safety is paramount and must be guaranteed if we are to enable human-robot collaborative teams. The person barrier in our system addresses this requirement and ensures user safety when he/she comes close to the robot. As our first case study demonstrated, the person barrier automatically follows the user, and thus, protecting the user wherever the user goes, by pausing the robot's movement when the user is near the robot. When the user moves away, the robot automatically resumes. This seamless transition of play-

[1]Demo video at https://youtu.be/yrgeevkAmRw

pause-play enables a smooth, efficient work flow, and at the same time, the user to move freely in the shared workspace.

In a dynamic work environment, such as a factory assembly line, where tools and workpieces may be frequently moved around, the ability to place virtual barriers at arbitrary locations allows the user to protect specific regions and objects. Compared to traditional approaches where the whole workspace needs to be modeled and provided to the program to compute collision-free paths, our system's ability to create virtual barriers at runtime, as demonstrated in our second case study, negates the need to remodel the environment every time the workspace configuration changes.

Currently, collision detection is performed in the HoloLens' onboard computer, and if a collision is detected, the trajectory modifier in the WAM's control computer is then notified through the "isCollision" flag (shown in Figure 2). This utilizes the collision detection functionality provided by the HoloLens's programming library. However, since the HoloLens communicates with the WAM's control computer wirelessly, it results in a communication delay, and thus, affects the smoothness of the robot's trajectory. One potential method to address this would be to move the collision detection module onto the WAM's controller computer.

VI. CONCLUSION AND FUTURE WORK

In this paper, we presented a virtual barrier system to protect workers who are using the HoloLens and objects that workers want to protect. We validated our virtual barrier system functionality through two use cases driven by collaborative manufacturing scenarios:

1) Robot arm automatically pauses when the user in the shared workspace comes into the robot's trajectory.
2) Robot arm automatically avoids and goes around objects protected by virtual barriers set by the user.

We are currently working on further developing our system in areas including allowing the user to create other geometrical virtual barriers and more complex virtual barriers, expanding our system to support and protect multiple concurrent users, as well as allowing users to dynamically adjust virtual barrier position and size after initial creation.

REFERENCES

[1] B. C. Kress and W. J. Cummings, "11-1: Invited paper: Towards the ultimate mixed reality experience: Hololens display architecture choices," in *SID Symposium Digest of Technical Papers*, vol. 48, no. 1. Wiley Online Library, 2017, pp. 127–131.
[2] "Magic leap," 2018, https://www.magicleap.com/.
[3] X. Wang, S. Ong, and A. Nee, "A comprehensive survey of augmented reality assembly research," *Advances in Manufacturing*, vol. 4, no. 1, pp. 1–22, 2016.
[4] T. Engelke, J. Keil, P. Rojtberg, F. Wientapper, M. Schmitt, and U. Bockholt, "Content first: a concept for industrial augmented reality maintenance applications using mobile devices," in *Proceedings of the 6th ACM Multimedia Systems Conference*. ACM, 2015, pp. 105–111.
[5] S. Webel, U. Bockholt, T. Engelke, N. Gavish, M. Olbrich, and C. Preusche, "An augmented reality training platform for assembly and maintenance skills," *Robotics and Autonomous Systems*, vol. 61, no. 4, pp. 398–403, 2013.
[6] S. A. Green, M. Billinghurst, X. Chen, and J. G. Chase, "Human-robot collaboration: A literature review and augmented reality approach in design," *International journal of advanced robotic systems*, vol. 5, no. 1, p. 1, 2008.
[7] Z. Pan, J. Polden, N. Larkin, S. Van Duin, and J. Norrish, "Recent progress on programming methods for industrial robots," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 87–94, 2012.
[8] M. F. Zaeh and W. Vogl, "Interactive laser-projection for programming industrial robots," in *Mixed and Augmented Reality, 2006. ISMAR 2006. IEEE/ACM International Symposium on*. IEEE, 2006, pp. 125–128.
[9] J. W. S. Chong, S. Ong, A. Y. Nee, and K. Youcef-Youmi, "Robot programming using augmented reality: An interactive method for planning collision-free paths," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 3, pp. 689–701, 2009.
[10] 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, 2009.
[11] C. P. Quintero, S. Li, M. K. Pan, W. P. Chan, H. M. V. der Loos, and E. Croft, "Robot programming through augmented trajectories in augmented reality," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018.
[12] C. P. Quintero, M. Dehghan, O. Ramirez, M. H. Ang, and M. Jagersand, "Flexible virtual fixture interface for path specification in telemanipulation," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5363–5368.

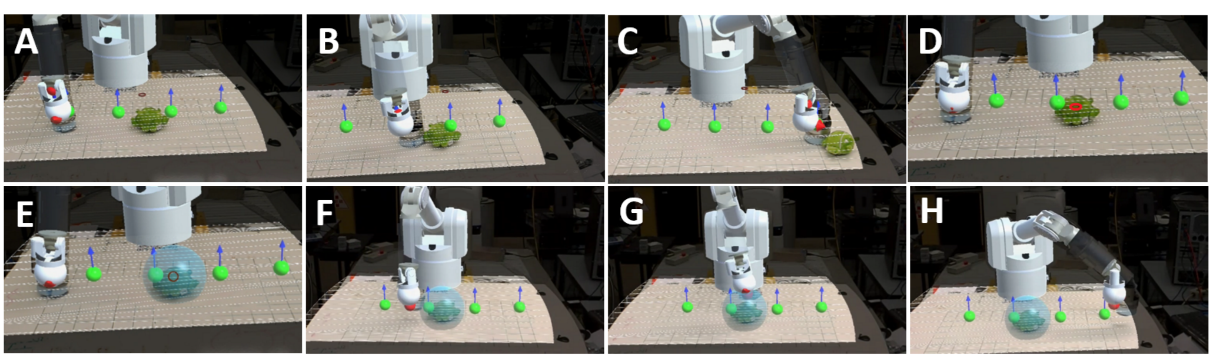

Fig. 4. User demonstrating the virtual barrier's functionality. A - The object is in the robot's planned trajectory. B - Without the virtual barrier system, the robot hits the object. C - The robot continues to move in its original trajectory, pushing the object away. D - The user specifies the region of interest by gazing at that location. A red ring is rendered to indicate the users gaze point. E - User sets a spherical virtual barrier around the object through the speech command "barrier". F - When the robot comes in contact with the virtual barrier, it is deflected away, thus, protecting the object. G - Robot moves around the spherical virtual barrier, which is protecting the object. H - Once the robot moves away from the barrier, it continues along its original path.