Received: 20 April 2019 | Accepted: 29 April 2019
DOI: 10.1002/cav.1881

SPECIAL ISSUE PAPER

WILEY

Interaction with proactive and reactive agents in box manipulation tasks in virtual environments

Kuan-Yu Liu[1] | Matias Volonte[2] | Yu-Chun Hsu[1] | Sabarish V. Babu[2] | Sai-Keung Wong[1]

[1]College of Computer Science, National Chiao Tung University, Hsinchu, Taiwan

[2]School of Computing, Clemson University, Clemson, South Carolina

Correspondence
Sai-Keung Wong, College of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan.
Email: cswingo@cs.nctu.edu.tw

Funding information
The Ministry of Science and Technology (MOST), Taiwan (Republic of China), Grant/Award Number: MOST 106-2221-E-009-161-MY2

Abstract

This paper studies the user collaboration experience with proactive and reactive agents in transporting boxes in virtual environments. Two main characters, the avatar and the agent, are controlled by a user and a controller, respectively. The user and the agent communicate with each other by voice. The agent can be proactive or reactive. The user follows the instruction issued by the proactive agent, whereas the user instructs the reactive agent to perform actions. The goal is to transport boxes to goal positions with orientation constraints. We conducted a user study to analyze the behaviors of participants in several aspects, including task completion time, path length, control experience, and co-presence experience. We report our findings and make suggestions for future development.

KEYWORDS

box manipulation, proactive agent, reactive agent, user study, virtual reality, voice communication

1 | INTRODUCTION

Human–agent interaction and collaboration are ubiquitous in virtual or real environments. Various kinds of studies have been investigated in virtual environments, such as virtual peers,[1] proactive and adaptive selling agents,[2] emotional contagion,[3] multiple interactions,[4] and emotional agents.[5] This paper studies the experience of users in collaboration tasks with virtual agents. The agents can be proactive or reactive. Proactive agents play a main leading role in guiding the process to complete tasks, whereas reactive agents respond to actions taken by users accordingly. Proactive and reactive controllers have been studied extensively in robotics.[6–8] However, little attention is paid to investigate the user experience in interaction with proactive and reactive virtual agents in virtual environments. We believe that such studies are important to understand user behaviors when users interact with these kinds of agents. Thus, we would be able to further enhance the user experience in virtual environments.

To make the problem tractable, we focus on the collaboration tasks in which boxes are transported to destinations with orientation constraints. In robotics, there are several ways to transport passive objects, such as pushing,[9] caging,[10,11] and case-based manipulation.[12] In virtual environments, virtual agents are simulated to transport passive objects by pushing,[13] carrying, and pulling.[14] Tandem compliant behaviors between agents and users in virtual environment simulations have great potential in applications such as training users in collaborative tasks such as team work skills in sports, mechanical operation, and motor coordination. In this paper, we develop a desktop system that simulates the motion of objects. There are two main characters, which are an avatar and an agent, and they work together to push boxes. A user controls an avatar, whereas an agent controller controls the agent. The behaviors of the agent are computed based on the agent type, namely, proactive or reactive agent interactions. Our system is integrated with a voice recognition system and a speech system so that the user can use voice commands to interact with the agent in a natural manner. We conducted a user study to evaluate the user experience in collaboration tasks. Our hypothesis is that the user experiences for interacting with the two types of agents will be different. We report our findings and summarize the results.

2 | RELATED WORK

Our work is relevant to human–agent interaction, object transportation, and agent modeling.

Human–agent interaction. A large amount of studies report various approaches for human–agent interaction, including conversational agents,[15] emotional agents,[5] virtual peers,[1] and communication based on natural languages.[16] A virtual selling agent can be proactive and adaptive.[2] She initiates a dialogue to support a user and reacts appropriately based on the user's profile. Techniques for verbal and nonverbal human–robot interactive communication are reviewed in the work of Mavridis.[17] A theoretical framework that can be used in collaborative task settings is the joint intention model which states that entities should perform a collective action toward a common goal.[18] A project from the National Aeronautics and Space Administration examined the dyadic interaction between humans and robots in a shared task scenario, and the robot can ask for support by speech.[19]

Object transportation. Transportation techniques have been proposed for objects of different shapes, such as arbitrary two-dimensional convex shape,[20] long rod shape,[21] box shape,[22] and disc shape.[23] In computer animation, collaboration and cooperative behaviors are simulated, such as agents pushing boxes in crowd using collision probability fields,[13] collective transport of ants,[24] and ant colony.[25] To successfully transport a passive object, agents should apply appropriate forces or move in proper speed to maneuver the passive object.

Agents and controllers. In the works of Chen and Wong[13] and Wong et al.,[14] agents are controlled to transport passive objects. Motions of the agents are computed based on handcrafted rules. There are no interactions between users and agents. In robotics, approaches have been proposed to make controllers learn proactive or reactive behaviors.[6–8] The controllers are often built up based on probability models. In a learning phase, the controllers encode human control at the preprocessing stage. There is a motion predictor that evaluates the quality of the actions taken by robots. Subsequently, the controllers adapt the human motion so that the collaboration process can be carried out smoothly. Such techniques inspire us to develop our controllers for virtual agents.

3 | SYSTEM OVERVIEW

Figure 1 shows the main architecture of our system. Our system has three kinds of main objects: an avatar, an agent, and boxes. The boxes are passive objects. The avatar and the agent are characters. The avatar and the agent apply contact forces to push the boxes to satisfy goal conditions. At one time, there is only one task in which the two characters push one box to a predefined goal. If the box matches the position and orientation conditions, the task is finished. The process is repeated for the remaining boxes. Figure 2 shows the snapshots of the game play.

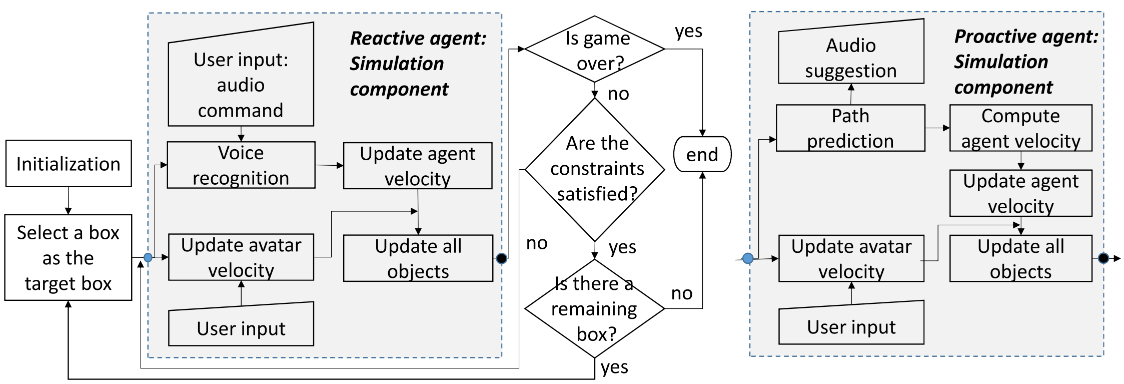


FIGURE 1 System architecture. The processes for the reactive and proactive agents are similar to each other. An avatar and an agent push a box to a goal. The communication is by voice

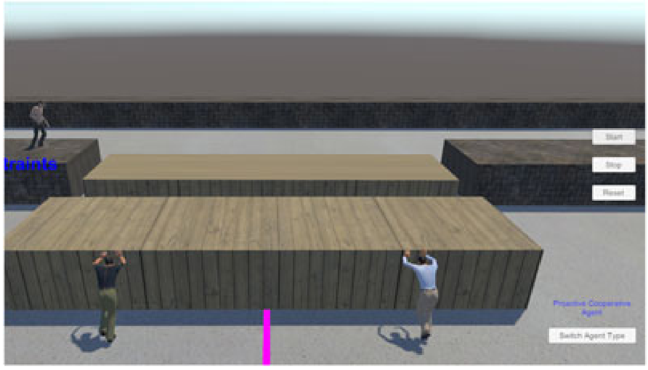


FIGURE 2 A game play snapshot. A (purple) polyline on the floor indicates the predefined path. The characters must push the box to its goal to build a bridge. In the Pressured condition, a zombie walks on the bridge formed by boxes

While working with a reactive agent, the user gives suggestions to the agent via voice commands. There are four basic voice commands, which are *speed up*, *keep pushing*, *slow down*, and *stop*. The user needs to evaluate the situation and then makes appropriate suggestions to the reactive agent. In an agent controller, a speech recognition system is employed to translate the voice input to a text. If the text matches with one of the commands, the agent controller updates the speed of the agent. A proactive agent performs actions based on a path predictor. The path predictor evaluates a set of pushing strategies. It evaluates the best combination of the avatar and the agent. Based on the desired speeds of the avatar and the agent, the controller of the proactive agent makes suggestions via the voice commands to the user.

4 | PRELIMINARY

In our virtual environment, the objects move on the x–z plane, and the y-axis is the up-direction. Each object i has its own local frame constituted by two axes, which are the x-axis (denoted as $\mathbf{x}_i$) and the z-axis (denoted as $\mathbf{z}_i$). For a box o, we use a face with a longer side for pushing. We assume that the normal of the face is negative to $\mathbf{z}_o$. A two-point manipulation method is flexible. To push a box, the avatar and the agent stand at the two sides of a face of the box, and their distances are the same to the center of mass of the box. While the box is being pushed, the relative positions of the avatar and the agent are fixed with respect to the box. The box can be pushed along a certain direction if the movement distances of the avatar and the agent are adjusted properly.

The two characters move to new positions to push a box. We can update the speed of a character i by

$$\Delta s_i^{t+1} = \eta_i^t \Delta t, \tag{1}$$

$$s_i^{t+1} = s_i^t + \Delta s_i^{t+1}, \tag{2}$$

where the superscript indicates time, η_i^t is the acceleration of the character, and Δt is the time step. The character's movement direction is along $\mathbf{z}_o$. Hence, the velocity and desired position of the character are computed by

$$\mathbf{v}_i^{t+1} = s_i^{t+1}\mathbf{z}_o, \tag{3}$$

$$\tilde{\mathbf{p}}_i^{t+1} = \mathbf{p}_i + \mathbf{v}_i^{t+1}\Delta t. \tag{4}$$

After the desired positions of the avatar and the agent are set, the physics-based simulator computes the simulation result for one step. Then, the new positions of all the objects are obtained. The process is repeated until the simulation is terminated. Friction parameters are set properly so that the boxes do not slip.

A goal is defined as $g = (o, \mathbf{p}_g, \mathbf{u}_g)$ for a box o, where $\mathbf{p}_g$ and $\mathbf{u}_g$ are a position and an orientation, respectively. To push box o to finish goal g, two conditions must be satisfied, and they are defined as follows.

1. Position condition: $\|\mathbf{p}_o^t - \mathbf{p}_g\| \leq \epsilon_p$
2. Orientation condition: $\theta(\mathbf{z}_o^t, \mathbf{u}_g) \leq \epsilon_\theta$

Here, $\mathbf{p}_o^t$ is the current position of o, $\mathbf{p}_g$ is the goal position, and $\theta(\mathbf{z}_o^t, \mathbf{u}_g)$ is the unsigned angle between the current local z-axis $\mathbf{z}_o^t$ and goal orientation $\mathbf{u}_g$. ϵ_p and ϵ_θ are thresholds.

5 | HUMAN–AGENT INTERACTION

This section presents our system components. We rely on gross motion and fine motion to push boxes.[26] To push a box to a goal, a path is generated (gross motion). Then, two characters adjust their speeds (fine motion) to push the box along the path. The speed of a character can be updated by Equation (2). Denote avatar as a and agent as b. The user can set η_a value to the avatar. We detail how η_b is determined for reactive and proactive agents.

For a reactive agent, the speed update depends on the voice suggestion of the user. The user makes a voice suggestion to the controller, and the controller computes the speed of agent b. Recall that $\Delta s_b = \eta_b \Delta t$. Δs_b is set as follows.

1. *Speedup*: $\Delta s_b = \text{rand}(0, s_{\max} - s_b^t)$
2. *Slow down*: $\Delta s_b = \text{rand}(-s_b^t, 0)$
3. *Keep pushing*: $\Delta s_b = 0$
4. *Stop*: $\Delta s_b = -s_b^t$

Here, rand() returns a random number between two input numbers. Based on Δs_b, η_b can be determined. Note that we can use a stratified sampling technique (e.g., bins) to obtain Δs_b to avoid getting a value close to the current speed.

For a proactive agent, the proactive controller predicts the trajectory of the box at real time and then makes a suggestion to the user. In the proactive controller, a path predictor samples some random combinations of speeds of the two characters. Then, it computes the best combination based on a cost function (the higher, the better). Due to a limited time budget, the entire computation is performed at real time. For a given a goal $(\mathbf{p}_g, \mathbf{u}_g)$, a path is computed (or designed by an animator). Denote the path as $Q = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n\}$, where $\mathbf{q}_1$ is the initial position of the box, and $\mathbf{q}_n = \mathbf{p}_g$. In the following, we assume that the box is pushed to a point $\mathbf{q}$. Denote $\theta(\mathbf{u}_1, \mathbf{u}_2)$ as the unsigned angle between two vectors $\mathbf{u}_1$ and $\mathbf{u}_2$. There are two cases to consider.

Case I: $\mathbf{q}$ is not the goal point. The cost C^{t+1} is computed as follows:

$$\beta^{t+1} = -\frac{\theta\left(\mathbf{z}_o^{t+1}, \mathbf{q} - \mathbf{p}^{t+1}\right) - \theta\left(\mathbf{z}_o^t, \mathbf{q} - \mathbf{p}^t\right)}{\gamma_{\max}}$$
$$\rho^{t+1} = -\frac{d(\mathbf{p}^{t+1}, \mathbf{q}) - d(\mathbf{p}^t, \mathbf{q})}{d_{\max}}$$
$$C^{t+1} = \omega^{t+1}\rho^{t+1} + (2 - \omega^{t+1})\beta^{t+1},$$

where

$$\omega^{t+1} = 1 - e^{-d(\mathbf{p}^{t+1}, \mathbf{p}_g)},$$
$$\gamma_{\max} = \frac{s_{\max}\Delta t}{d(a, b)}, \text{ and}$$
$$d_{\max} = s_{\max}\Delta t.$$

Here, $s_{\max}$ is the maximum walking speed of the characters, $d(a, b)$ is the distance between the characters, and $d(\mathbf{p}_1, \mathbf{p}_2)$ is the distance between two points $\mathbf{p}_1$ and $\mathbf{p}_2$. β^{t+1} is the orientation cost, and ρ^{t+1} is the distance cost. ω^{t+1} balances between the orientation cost and the distance cost.

Case II: $\mathbf{q}$ is the goal point $\mathbf{p}_g$. When the box is near the goal, adjusting the box orientation is more important than pushing the box closer to the goal position. Therefore, if $||\mathbf{p}^{t+1} - \mathbf{p}_g)|| \leq r$ and $\omega^{t+1} < \omega^{(t)}$, $\omega^{(t+1)}$ is updated. Otherwise, the previous value of ω^t is used as ω^{t+1}. Here, r is a threshold, and $r = 3$ (in meters). The angle term becomes

$$\beta^{t+1} = -\frac{\theta\left(\mathbf{z}_o^{t+1}, \mathbf{u}^g\right) - \theta\left(\mathbf{z}_o^t, \mathbf{u}^g\right)}{\gamma_{\max}}.$$

Algorithm 1 shows the psuedocode for computing the desired characters' speeds. The speeds of the avatar and the agent are inside intervals $\mathbf{I}_a$ and $\mathbf{I}_b$, respectively. For example, $\mathbf{I}_a = \mathbf{I}_b = [0, 2.5]$ (in meters per second). We adopt a stratified sampling strategy to sample the speeds of the two characters due to its lower standard deviation, comparing to simple random sampling. The interval is divided into bins, and a speed is uniformly sampled from each bin.

Algorithm 1 Path predictor

input: avatar a, agent b, box o, and path Q
Output: speed pair (s_a, s_b)
Reset the best score
$\mathbf{I}_a \leftarrow$ get a's intervals of speed
$\mathbf{I}_b \leftarrow$ get b's intervals of speed
for each interval I_1 in $\mathbf{I}_a$ **do**
  for each interval I_2 in $\mathbf{I}_b$ **do**
    $s_1 \leftarrow$ randomSample(I_1)
    $s_2 \leftarrow$ randomSample(I_2)
    Store states of all objects
    PerformPhysicsSimulation(s_1, s_2)
    ComputePredictionScore(o, Q)
    if the new score is better **then**
      $(s_a, s_b) \leftarrow (s_1, s_2)$
      Update the best score
    end if
    Restore states of all objects
  end for
end for
Return (s_a, s_b)

Voice suggestion. Algorithm 1 computes the speed pair (s_a, s_b) for the characters. The controller updates the speed of the proactive agent based on $\Delta s_b = s_b - s_b^t$. For the suggestion, we compute $\Delta s_a = s_a - s_a^t$, which is mapped to a voice command in the following priority order.

1. *Stop*: if $s_a \leq \epsilon_{s_1}$
2. *Speedup*: if $\Delta s_a \geq \epsilon_{s_2}$
3. *Slow down*: if $\Delta s_a \leq -\epsilon_{s_2}$
4. *Keep pushing*: otherwise

The user listens to the voice command and then controls the avatar properly to move. ϵ_{s_1} and ϵ_{s_2} are user-defined parameters. We set $\epsilon_{s_1} = 0.2$ and $\epsilon_{s_2} = 0.5$.

6 | USER STUDY DESIGN

In order to test the user experience with this proactive and reactive agent simulation, we developed a $2 \times 2 \times 3$ within-subjects repeated-measures experiment. There are two constraint conditions. In the No-Pressured condition, boxes are moved to predefined goal positions without a time constraint. In the Pressured condition, boxes are moved to form a bridge for an elevated walkway along which a zombie is walking over in a time-constrained situation. In the Pressured condition, the zombie is walking at constant velocity along the walkway path, and the boxes need to be placed as soon as possible to bridge gaps in the zombie's path such that it can cross the entire path successfully. By successfully placing the boxes in the predefined areas, the zombie continues its travel along the path. If the boxes are not placed properly, the zombie falls and returns to the previous waypoint or nearest starting point. Thus, this attempt is counted as a failed attempt.

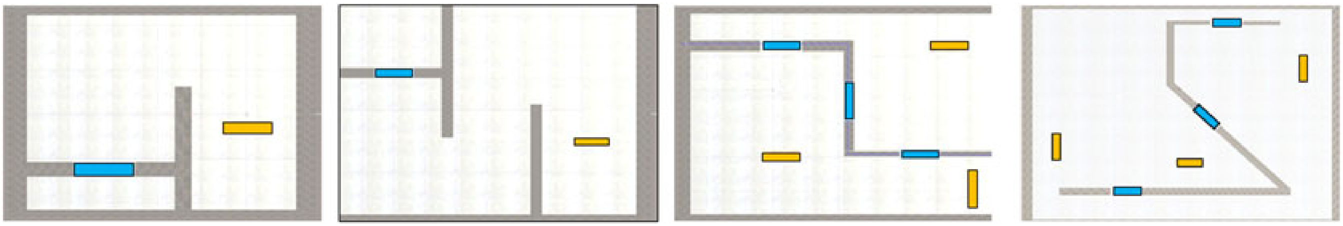


FIGURE 3 Some game levels in the user study. The rectangles are boxes. Their goals are the gaps (shown as a box) between fixed boxes. All the boxes have the same dimensions. Note that the maps have been resized

There are three conditions of difficulty, namely, Easy, Medium, and Hard. In the easy level, a single box is needed to be pushed. The medium level entails moving a box through a longer path through obstacles. The hard level involves moving a series of three boxes to get around obstacles. The decision of creating No-Pressured and Pressured conditions was based on studies reporting that pressured game scenarios influence the sense of flow and engagement for participants.[27] Figure 3 shows some game levels.

7 | USER STUDY PROCEDURE

First, each participant listened to a brief explanation regarding the experiment. After consent was obtained, the participant was introduced to a training session where an experimenter showed the functioning of the game in the reactive and proactive roles. After training, the participant was introduced to the first session of the experiment. The participant played the proactive or reactive role in a random manner. Users played a total of six levels under No-Pressured and Pressured conditions. In each condition, there were two Easy, two Medium, and two Hard modes. After the completion of the levels, participants filled out the Psychometric Evaluation of the Post-Study System Usability,[28] Game Experience,[29] Networked Minds,[30] and an overall evaluation questionnaire. Next, the participants refreshed for 10 min before the second session of the experiment began. In the second session, the participant played different sets of six levels (two Easy, two Medium, and two Hard) in another role (e.g., playing the reactive role if the proactive role was played in the first session). Finally, the participant filled out the same questionnaires as the first session and were finally debriefed.

8 | EXPERIMENTS AND RESULTS

We implemented our methods using Unity (version 2018.2.1f1). The system configuration was Intel Core CPU: Intel i7-8700 K @ 3.70GHz, with 16GB RAM and NVIDIA GTX 1080. The simulation time step was set to 1/60 s. The frame rate was 60. The maximum walking speed of characters was 2.5 ms^{-1}. We recruited 20 participants, including 10 males and 10 females, between 20 and 30 years old.

8.1 | Quantitative results

During the experiments, we collected the items, including (IA) total time used, (IB) movement distance, (IC) number of restart times, (ID) position errors, and (IE) orientation errors for boxes. Tables 1 and 2 show the results for participants under conditions No-Pressured and Pressured, respectively. Participants took much longer time to finish tasks with reactive agents (reactive tasks) than those with proactive agents (proactive tasks). The orientation errors were higher in reactive tasks than in proactive tasks. The results indicated that participants encountered difficulty in working with a reactive agent. Sometimes, the speech recognition system failed to detect the suggestions by participants. In addition, the amount of the change in speed of the agent was not suitable. In most cases, the numbers of restart times were higher in reactive tasks than in proactive tasks. When a participant worked with a proactive agent, the participant could observe the states of the environment and made a proper decision. There were reference paths on the floor for the participants

TABLE 1 The results for participants under the No-Pressured condition. The numbers are denoted as mean (standard deviation). Numbers are bold for $p < 0.05$ (paired-samples t test) in comparison of the equality of means between proactive and reactive tasks

No-Pressured condition									
Proactive	All			Male			Female		
Difficulty	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
Time Used	**29**	**88**	**81**	**32**	**65**	**81**	27	110	**82**
(sec)	(15)	(55)	(10)	(18)	(32)	(11)	(10)	(64)	(10)
Movement Length	**31**	103	**110**	**31**	82	113	31	125	**107**
(m)	(11)	(55)	(9)	(11)	(28)	(12)	(12)	(66)	(3)
Restart #Times	**0.15**	0.75	**0.05**	**0.20**	0.20	0.10	0.10	1.30	**0 .00**
(number)	(0.36)	(1.14)	(0.22)	(0.40)	(0.40)	(0.30)	(0.30)	(1.35)	(0.00)
Position Error	0.41	0.41	0.45	0.49	0.40	0.45	0.34	0.42	0.44
(m)	(0.46)	(0.29)	(0.19)	(0.57)	(0.27)	(0.23)	(0.28)	(0.31)	(0.15)
Orientation Error	1.12	**0.66**	**0.79**	0.80	**0.52**	**0.67**	1.44	**0.80**	**0.91**
(degree)	(1.33)	(0.96)	(0.76)	(1.34)	(0.70)	(0.50)	(1.25)	(1.15)	(0.93)
Reactive	All			Male			Female		
Difficulty	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
Time Used	**77**	**134**	**171**	**91**	**103**	**122**	64	165	**220**
(sec)	(72)	(91)	(83)	(73)	(38)	(40)	(67)	(115)	(85)
Movement Length	**47**	104	**140**	**59**	87	116	34	121	**165**
(m)	(32)	(60)	(55)	(40)	(31)	(25)	(13)	(76)	(66)
Restart #Times	**0.75**	0.75	**0.55**	**1.20**	0.60	0.30	0.30	0.90	**0.80**
(number)	(1.14)	(1.00)	(0.87)	(1.40)	(0.66)	(0.90)	(0.46)	(1.22)	(0.75)
Position Error	0.36	0.42	0.40	0.41	0.42	0.35	0.31	0.42	0.44
(m)	(0.45)	(0.27)	(0.21)	(0.60)	(0.29)	(0.11)	(0.18)	(0.25)	(0.27)
Orientation Error	1.79	**2.57**	**2.08**	1.72	**2.65**	**1.69**	1.86	**2.49**	**2.48**
(degree)	(1.74)	(1.46)	(1.07)	(1.80)	(1.59)	(0.70)	(1.67)	(1.30)	(1.22)

to follow. The movement distances and position errors in both kinds of tasks were quite similar. This may be due to that there were reference paths and that the goal position of a box was at a gap between fixed boxes. Thus, the box could be positioned properly.

8.2 | Game play experience questionnaire

We found a significant difference with regard to perceived game flow experience. Participants perceived the game flow experience in the reactive condition ($M = 1.74, SD = 0.78$) to be significantly higher than in the proactive condition ($M = 1.53, SD = 0.70$), Wilcoxon's rank sum $Z = -2.0, p = 0.045$. We did not find any significant difference between the proactive and reactive agent conditions in any of the other dimensions of the game experience questionnaire or in any of the other questionnaires, such as Co-Presence ($M = 3.7$ in Proactive and $M = 3.4$ in Reactive, $p = 0.13$), System Usability Scale, and Overall System Assessment subjective quantitative responses.

8.3 | Overall preference questionnaire

In response to the question of which session was their favorite, 13 participants favored the reactive condition, whereas seven preferred the proactive condition. Some of the comments provided by the participants suggested that the reactive agent was "*easy to use*," "*easy to complete*," "*easy to control by the user*," "*more challenging*," "*more interesting to control the agent*," and "*easier to reach the goal*." On the other hand, participants in the proactive condition mentioned that this scenario was "*easier to play*," "*easier to reach the goal*," "*easier to finish the task*," and they can just "*let the agent move the box and just react*." One participant said that "*the first (proactive agent) condition was useful, but the second (reactive agent) is more interesting to play with*."

In response to the question of which type of scenario they would like to play again, 15 participants preferred to play the reactive condition again, whereas five participants preferred to play the proactive condition again, $X^2(1, N = 20) = 5.0$,

TABLE 2 The results for participants under the Pressured condition. The numbers are denoted as mean (standard deviation). Numbers are bold for $p < 0.05$ (paired-samples t test) in comparison of the equality of means between proactive and reactive tasks

Pressured condition									
Proactive	All			Male			Female		
Difficulty	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
Time Used	28	89	**148**	**22**	99	146	34	**79**	**149**
(sec)	**(17)**	(63)	(78)	(9)	(63)	(97)	(21)	(61)	(53)
Movement Length	37	115	188	30	128	186	44	101	191
(m)	(20)	(68)	(68)	(9)	(74)	(86)	(24)	(59)	(43)
Restart #Times	**0.35**	0.80	1.25	0.10	0.80	1.10	0.60	0.80	1.40
(number)	(0.65)	(1.69)	(1.48)	(0.30)	(1.17)	(1.58)	(0.80)	(2.09)	(1.36)
Position Error	0.23	0.38	0.48	0.24	0.45	0.47	0.22	0.32	**0.48**
(m)	(0.12)	(0.25)	(0.19)	(0.13)	(0.33)	(0.17)	(0.10)	(0.10)	(0.20)
Orientation Error	**0.73**	**1.33**	0.87	0.47	2.09	0.71	0.99	**0.58**	1.02
(degree)	(1.09)	(1.55)	(0.92)	(0.78)	(1.83)	(0.86)	(1.28)	(0.57)	(0.95)
Reactive	All			Male			Female		
Difficulty	Easy	Medium	Hard	Easy	Medium	Hard	Easy	Medium	Hard
Time Used	73	122	**235**	**54**	84	206	92	**160**	**261**
(sec)	(105)	(75)	(101)	(46)	(62)	(67)	(138)	(68)	(118)
Movement Length	**51**	103	178	48	80	177	55	127	179
(m)	(37)	(45)	(53)	(32)	(30)	(42)	(41)	(46)	(62)
Restart #Times	**0.90**	0.80	1.00	0.70	0.40	0.89	1.10	1.20	1.10
(number)	(1.22)	(1.03)	(1.12)	(1.01)	(0.66)	(0.88)	(1.38)	(1.17)	(1.30)
Position Error	0.20	0.26	0.93	0.19	0.27	1.59	0.20	0.25	**0.34**
(m)	(0.08)	(0.10)	(2.43)	(0.08)	(0.09)	(3.40)	(0.07)	(0.10)	(0.13)
Orientation Error	**1.58**	**2.30**	3.24	1.37	2.22	5.04	1.79	**2.38**	1.63
(degree)	(1.75)	(1.66)	(6.62)	(1.92)	(1.57)	(9.27)	(1.54)	(1.75)	(0.69)

$p = 0.025$. Some of the participants who preferred the proactive condition mentioned that they had a sense of "*achievement when cooperating with the agent*," "*wanted to challenge the computer's commands*," "*the agent could move the box according to the situation and they just had to follow*," and "*it was easier to use*." On the other hand, some of the participants that preferred the reactive condition said these: "*easy to control the situation by themselves*," "*easy to complete and achieve*," "*they can control everything*," "*task was more fun*," "*more challenging*," "*better as a game*," and "*fun to speak to agent*."

In response to the question of which type of condition they would recommend to others, 15 participants suggested that they would recommend the reactive condition, whereas five participants suggested that they would recommend the proactive condition, $X^2(1, N = 20) = 5.0, p = 0.025$. Some of the participants who suggested that they would recommend the proactive agent said that it was because "*their partner made great commands*,“ ”*wanted to try being controlled by the computer*," "*easier to play and simple to use*," and "*easier to adapt*." On the other hand, participants who suggested that they would recommend the reactive condition said that it was because it was "*more challenging*," "*more interactive*," "*provided higher degrees of freedom*," "*preferred to control the situation by themselves*," "*can learn to give commands to others*," "*learn to control and collaborate with others*," and it was "*more fun*."

Discussion. The physics fidelity relies on the simulation quality. We enable factors including friction, drag, force, inertia, and gravity. However, a user does not feel the pressure between his hands and the box while he pushes the box in the virtual environment. In a real case, a user can feel the speed change of the box, and thus, a better decision can be made. We assume that the relative positions of the characters are kept. However, in a real case, their relative positions may be changed, and there can be environmental uncertainty. This affects the prediction accuracy of the path predictor for a proactive agent in real environments.

We enabled collision response between the boxes and obstacles. There were small gaps between the obstacles and the desired positions of the boxes. Thus, in both conditions, the performances of the participants were similar to each other in position and orientation errors for pushing the boxes. We would like to investigate further to explore how to ensure performance errors that can be reduced under a nonconstrained condition for collision response.

9 | CONCLUSIONS AND FUTURE WORK

Our work models and studies the user experience with proactive and reactive agents in box manipulation tasks in virtual environments. The user and the agents communicate with each other via voice. We conducted a within-subjects user study to evaluate various aspects of the user experiences. We found that the participants with proactive and reactive agents exhibit different performance, perception, and preference results. In general, participants took longer time, a larger number of speed changes, and a greater number of attempts with reactive agents than with proactive agents. However, despite the difficulties with efficiency, participants preferred to work with reactive agents again and would recommend others to interact with reactive agents, as they were perceived as greatly interactive, educational, enjoyable, and cooperative than proactive agents. In the future, we would like to improve both kinds of controllers to finish complex tasks such as those requiring diverse motor skills in immersive simulations. Deep learning techniques can be adopted to enhance the function of controllers for proactive and reactive agents so that the agents can handle diverse interaction situations with uncertainty.

ACKNOWLEDGEMENTS

We would like to thank all the participants. This research was supported by The Ministry of Science and Technology (MOST), Taiwan (Republic of China) under Grant MOST 106-2221-E-009-161-MY2.

ORCID

Sai-Keung Wong https://orcid.org/0000-0002-4248-0052

REFERENCES

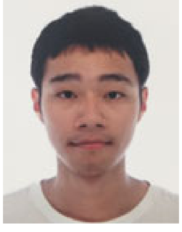
1. Babu SV, Grechkin TY, Chihak B, et al. An immersive virtual peer for studying social influences on child cyclists' road-crossing behavior. IEEE Trans Vis Comput Graph. 2011;17(1):14–25.
2. Delecroix F, Morge M, Routier JC. A virtual selling agent which is proactive and adaptive. In: Demazeau Y, Müller JP, Rodríguez JMC, Pérez JB, editors. Advances on practical applications of agents and multi-agent systems. Berlin, Germany: Springer-Verlag Berlin Heidelberg; 2012. p. 57–66.
3. Wu Y, Babu SV, Armstrong R, et al. Effects of virtual human animation on emotion contagion in simulated inter-personal experiences. IEEE Trans Vis Comput Graph. 2014;20(4):626–635.
4. Barange M, Saunier J, Pauchet A. Multiparty interactions for coordination in a mixed human-agent teamwork. In: Beskow J, Peters C, Castellano G, O'Sullivan C, Leite I, Kopp S, editors. Intelligent virtual agents. Cham, Switzerland: Springer International Publishing; 2017. p. 29–42.
5. Huang PH, Wong SK. Emotional virtual crowd on task completion in virtual markets. Comput Animat Virtual Worlds. 29(3–4):e1818. https://doi.org/10.1002/cav.1818.
6. Rozo L, Bruno D, Calinon S, Caldwell DG. Learning optimal controllers in human-robot cooperative transportation tasks with position and force constraints. Paper presented at: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2015 Sept 28–Oct 2; Hamburg, Germany. Piscataway, NJ: IEEE; 2015. p. 1024–1030.
7. Sheng W, Thobbi A, Gu Y. An integrated framework for human-robot collaborative manipulation. IEEE Trans Cybern. 2015;45(10):2030–2041.
8. Rozo L, Silvério J, Calinon S, Caldwell DG. Learning controllers for reactive and proactive behaviors in human-robot collaboration. Front Robotics AI. 2016;3:30.
9. Mataric MJ, Nilsson M, Simsarin KT. Cooperative multi-robot box-pushing. Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots, vol. 3; 1995 Aug 5–9; Pittsburgh, PA. Piscataway, NJ: IEEE; 1995. p. 556–561.
10. Fink J, Hsieh MA, Kumar V. Multi-robot manipulation via caging in environments with obstacles. Paper presented at: IEEE International Conference on Robotics and Automation; 2008 May 19–23; Pasadena, CA. Piscataway, NJ: IEEE; 2008. p. 1471–1476.
11. Wan W, Shi B, Wang Z, Fukui R. Multirobot object transport via robust caging. IEEE Trans Syst Man Cybern: Syst. 2017. Early Access.
12. Meriçli T, Veloso M, Akın HL. A case-based approach to mobile push-manipulation. J Intell Robotic Syst. 2015;80(1):189–203.
13. Chen H, Wong SK. Transporting objects by multiagent cooperation in crowd simulation. Comput Animat Virtual Worlds. 2018;29(3–4):e1826. https://doi.org/10.1002/cav.1826.
14. Wong SK, Chou YH, Yang HY. A framework for simulating agent-based cooperative tasks in crowd simulation. Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games; 2018 May 15–18; Montreal, Canada. New York, NY: ACM; 2018.
15. Jan D, Roque A, Leuski A, Morie J, Traum D. A virtual tour guide for virtual worlds. Paper presented at: International Conference on Intelligent Virtual Agents; 2009 Sep 14–16; Amsterdam, The Netherlands. Berlin, Germany: Springer; 2009. p. 372–378.

16. Wagoner AR, Matson ET. A robust human-robot communication system using natural language for harms. Procedia Comput Sci. 2015;56:119–126. Part of special issue: The 10th International Conference on Future Networks and Communications (FNC 2015)/The 12th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2015) Affiliated Workshops.
17. Mavridis N. A review of verbal and non-verbal human-robot interactive communication. Robotics Auton Syst. 2015;63:22–35.
18. Jennings NR. Specification and implementation of a belief-desire-joint-intention architecture for collaborative problem solving. Int J Intell Coop Inf Syst. 1993;2(03):289–318.
19. Hoffman G, Breazeal C. Collaboration in human-robot teams. Paper presented at: AIAA 1st Intelligent Systems Technical Conference; 2004 Sep 20–22; Chicago, IL. Reston, VA: AIAA; 2004. p. 6434.
20. Li Q, Payandeh S. Manipulation of convex objects via two-agent point-contact push. Int J Robotics Res. 2007;26(4):377–403.
21. Ohashi K, Takahashi R, Takimoto M, Kambayashi Y. Cooperative transportation of a rod using mobile agents. Paper presented at: IEEE International Conference on Robotics and Biomimetics (ROBIO 2014); 2014 Dec 5–10; Bali, Indonesia. Piscataway, NJ: IEEE; 2014. p. 1019–1026.
22. Parra-González EF, Ramírez-Torres G, Toscano-Pulido G. Motion planning for cooperative multi-robot box-pushing problem. Paper presented at: Ibero-American Conference on Artificial Intelligence; 2008 Oct 14–17; Lisbon, Portugal. Berlin, Germany: Springer; 2008.
23. Nieuwenhuisen D, van der Stappen AF, Overmars MH. Pushing a disk using compliance. IEEE Trans Robotics. 2007;23(3):431–442.
24. Guo S, Wang M, Notman G, Chang J, Zhang J, Liao M. Simulating collective transport of virtual ants. Comput Animat Virtual Worlds. 2017;28(3–4):e1779. https://doi.org/10.1002/cav.1779.
25. Xiang W, Ren J, Wang K, Deng Z, Jin X. Biologically inspired ant colony simulation. Comput Animat Virtual Worlds. e1867. https://doi.org/10.1002/cav.1867.
26. Yamashita A, Fukuchi M, Ota J, Arai T, Asama H. Motion planning for cooperative transportation of a large object by multiple mobile robots in a 3D environment. Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings, vol. 4; 2000 April 24–28; San Francisco, CA. Piscataway, NJ: IEEE; 2000. p. 3144–3151.
27. Yildirim IG. Time pressure as video game design element and basic need satisfaction. PhD [thesis]. Ankara, Turkey: Middle East Technical University; 2015.
28. Lewis JR. Psychometric evaluation of the post-study system usability questionnaire: the PSSUQ. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 36; 1992 Oct 12–16; Atlanta, GA. Los Angeles, CA: SAGE Publications; 1992. p. 1259–1260.
29. IJsselsteijn WA, De Kort YAW, Poels K. The game experience questionnaire. Eindh Tech Univ Eindh. 2013.
30. Harms C, Biocca F. Internal consistency and reliability of the networked minds measure of social presence. Paper presented at: 7th Annual International Workshop: Presence 2004; 2004 Oct 13–15; Valencia, Spain. Valencia, Spain: Universidad Politecnica de Valencia; 2004.

AUTHOR BIOGRAPHIES

Kuan-Yu Liu is a master student in the Institute of Multimedia Engineering at National Chiao Tung University, Hsinchu, Taiwan. His research interests include computer graphics, virtual reality, and human-agent interaction.

Matias Volonte is a PhD Student in Human Centered Computing in the School of Computing at Clemson University, Clemson, South Carolina. His research interests include human and virtual human interaction and computer animation.



Yu-Chun Hsu is a master student in the Institute of Multimedia Engineering at National Chiao Tung University, Hsinchu, Taiwan. His research interests include VR experience, eye gaze simulation and crowd simulation.

Sabarish Babu is an Associate Professor in the Division of Human Centered Computing in the School of Computing at Clemson University in the USA. He received his BS (2000), MS (2002) and PhD (2007) degrees from the University of North Carolina at Charlotte, and completed a post-doctoral fellowship in the Department of Computer Science at the University of Iowa prior to joining Clemson University in 2010. His research interests are in the areas of virtual environments, virtual humans, applied perception, educational virtual reality, and 3D human computer interaction. He has authored or co-authored over 75 peer-reviewed publications in premiere venues in the research field. He was the General Chair of the IEEE International Conference on Virtual Reality (IEEE VR) 2016. He also served as a Program Co-Chair for IEEE VR 2017. He and his students have received Best Paper Awards in the IEEE International Conference on Virtual Reality, IEEE International Conference on 3D User Interaction, ACM Symposium on Applied Perception, and the IEEE International Conference on Healthcare Informatics. His research has been sponsored by the US National Science Foundation, US Department of Labor, St. Francis and Medline Medical Foundations.

Sai-Keung Wong received the M.Phil. and Ph.D. degrees in Computer Science from the Hong Kong University of Science and Technology in 1999 and 2005, respectively. Since August 2014, he has been an Associate Professor with the Department of Computer Science and the Institute of Multimedia Engineering, National Chiao Tung University, Hsinchu, Taiwan. His research interests include computer graphics, virtual reality, and human-computer interaction. Dr. Wong received the Distinguished Mentor Award in 2011 and the Excellent Teaching Award in 2014.

How to cite this article: Liu K-Y, Volonte M, Hsu Y-C, Babu SV, Wong S-K. Interaction with proactive and reactive agents in box manipulation tasks in virtual environments. *Comput Anim Virtual Worlds*. 2019;30:e1881. https://doi.org/10.1002/cav.1881