# TiltWalker: Operating a Telepresence Robot with One-Hand by Tilt Controls on a Smartphone

GHAZAL ZAND, Inclusive Interaction Lab, University of California, Merced, United States
YUAN REN, Inclusive Interaction Lab, University of California, Merced, United States
AHMED SABBIR ARIF, Inclusive Interaction Lab, University of California, Merced, United States

Mobile clients for telepresence robots are cluttered with interactive elements that either leave a little room for the camera feeds or occlude them. Many do not provide meaningful feedback on the robot's state and most require the use of both hands. These make maneuvering telepresence robots difficult with mobile devices. TiltWalker enables controlling a telepresence robot with one hand using tilt gestures with a smartphone. In a series of studies, we first justify the use of a Web platform, determine how far and fast users can tilt without compromising the comfort and the legibility of the display content, and identify a velocity-based function well-suited for control-display mapping. We refine TiltWalker based on the findings of these studies, then compare it with a default method in the final study. Results revealed that TiltWalker is significantly faster and more accurate than the default method. Besides, participants preferred TiltWalker's interaction methods and graphical feedback significantly more than those of the default method.

CCS Concepts: • **Computer systems organization** → **Robotic control**; **External interfaces for robotics**; • **Human-centered computing** → *Gestural input*.

Additional Key Words and Phrases: robotics, control, maneuver, remote, out-of-sight, mobile interaction, tilting, tilt-based, telepresence

572

**ACM Reference Format:**
Ghazal Zand, Yuan Ren, and Ahmed Sabbir Arif. 2022. TiltWalker: Operating a Telepresence Robot with One-Hand by Tilt Controls on a Smartphone. *Proc. ACM Hum.-Comput. Interact.* 6, ISS, Article 572 (December 2022), 26 pages. https://doi.org/10.1145/3567725

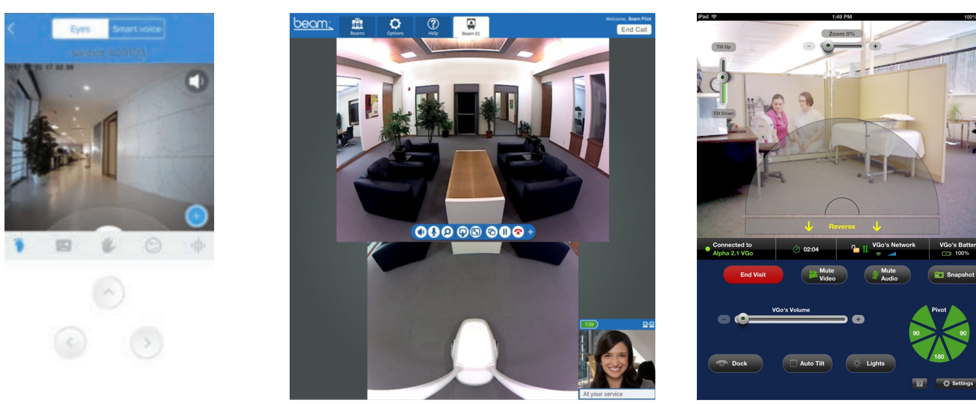
## 1 INTRODUCTION

Telepresence robots incorporate videotelephony equipment onto robots that users can operate and steer from remote locations. Its purpose is to enable users to visit places and be with others from the comfort and safety of one's home. Today, many people around the world use telepresence robots as part of their everyday routines [68]. Telepresence robots are also becoming increasingly popular among special user groups [46, 81] and within certain application domains such as education [23, 74] and office environments [39, 48], healthcare [38, 73], independent living for the elderly [9, 62, 78], live events [53, 67] and shopping [79]. The recent global pandemic has also seen an increase in telepresence robots in these contexts since users could visit places, attend events, and communicate with healthcare professionals and their loved ones without risking anyone's well-being [22, 31, 43, 49]. However, the technology has a long way to go before it would be effective and usable enough for the consumers to adapt it in their daily lives.

Authors' addresses: Ghazal Zand, Inclusive Interaction Lab, University of California, Merced, 5200 N. Lake Road, Merced, California, United States, 95343, gzand@ucmerced.edu; Yuan Ren, Inclusive Interaction Lab, University of California, Merced, 5200 N. Lake Road, Merced, California, United States, 95343, yren5@ucmerced.edu; Ahmed Sabbir Arif, Inclusive Interaction Lab, University of California, Merced, 5200 N. Lake Road, Merced, California, United States, 95343, asarif@ucmerced.edu.


2573-0142/2022/12-ART572
https://doi.org/10.1145/3567725

One factor that limits telepresence robots' global acceptance is the difficulties in maneuvering the robots. Prior research revealed that users find variating and maintaining the speed, turning, and reversing the robot tedious and heavy in cognitive load, resulting in the robot to back into walls, running into obstacles, and moving too slowly to avoid collisions [4, 31, 57, 60]. These works identified unintuitive interaction methods and the lack of relevant information in the client applications as one of the main reasons for this, thus recommended making interaction with the robots as natural as possible and provide users with wider and uncluttered views of the cameras and relevant feedback and indicators conveying the speed, direction, and degrees of rotation turned for making informed maneuvering decisions [31, 36, 37, 57, 60]. Besides, most client applications are designed and optimized for desktop platforms that confine users to a desk, limiting their mobility and the ability to use telepresence robots when travelling or on-the-move. Their mobile counterparts are cluttered with interactive elements like sliders and buttons, which not only reduct and occlude the camera views but also makes it difficult to use the applications with one hand. They also require repeated actions for controlling the robots, such as tapping on a button repeatedly until the robots reach an intended speed. In this paper, we attempt to address these limitations with TiltWalker, a mobile client that enables controlling a telepresence robot with one hand by using tilt gestures with a smartphone.




(a) Sanbot Elf mobile application (b) Beam Pro tablet application (c) VGo/Veena tablet application

Fig. 1. Screenshots of three mobile applications for controlling telepresence robots.

## 2 RELATED WORK

Almost all commercial telepresence robots provide both desktop and mobile applications (mostly aimed at tablets) to remotely control the robots. However, a comprehensive review of mobile applications for the most popular telepresence robots revealed that most mobile applications do not support all control features supported by their desktop counterparts. Table 1 presents the findings of the survey, which also revealed that the most mobile applications are cluttered with interactive elements, like buttons, sliders, and virtual trackpads, leaving very little room for the two commonly used camera views for navigation: a downward-facing camera view and the robot's head view. Using too many interactive elements also makes it difficult for users to learn all their functions and the interface confusing. Some visual elements, e.g., labels and alerts, also occlude the camera views, making navigating the robot difficult, especially when surrounded by obstacles, e.g., in a busy room. Fig. 1 presents screenshots of one mobile and two tablet interfaces for controlling three popular telepresence robots.

Table 1. Supported controls and corresponding actions available in default mobile applications for popular telepresence robots (in alphabetical order). The "×" symbol indicates unavailable feature. Tap-hold actions are performed on the display, while press-hold actions are performed on specific buttons.

| Robot | Robot Move | Robot Turn | Speed | Camera Pan | Camera Zoom |
|---|---|---|---|---|---|
| Amy A2[1] | Tap-drag | Body control | × | × | × |
| Ava[2] | Tap-drag | Press-hold | × | × | × |
| Beam Pro[3] | Tap-drag | Tap-drag | × | × | Button & Slider |
| Double 3[4] | Press-hold | × | × | Press-hold | × |
| Ohmni[5] | Press-hold | Slider | × | Press-hold | × |
| PadBot P2[6] | Tap-drag | Tap-drag | × | × | Button & slider |
| Sanbot Elf[7] | Tap-hold | Movement control | × | × | × |
| Temi[8] | Tap-hold | Tap-hold | × | × | × |
| VGo/Veena[9] | Tap-drag | Tap-drag | Tap-drag | Tap-drag | Slider |

[1]Amy A2, http://www.amyrobotics.com/indexen, [2]Ava, https://www.avarobotics.com, [3]Beam Pro, https://www.suitabletech.com, [4]Double 3, https://www.doublerobotics.com, [5]Ohmni, https://ohmnilabs.com, [6]PadBot P2, https://www.padbot.com, [7]Sanbot Elf, http://en.sanbot.com, [8]Temi, https://www.robotemi.com, [9]VGo/Veena, http://www.vgocom.com

There are also some academic solutions. Dong et al. [19] used a video stitching algorithm to combine two video frames, one from a wide-angle camera for looking forward and another from fish-eye camera to view the ground, to generate a large view video frame. An evaluation revealed that the stitched live video improves task efficiency, accuracy, and remote operators' feelings of presence. Mosiello et al. [47] proposed a feature that overlays visual indicators over the video feed transmitted by the robot to help users understand distance, depth, and dimensions. Batool et al. [5] developed an Internet of Things (IoT) system with many sensors and actuators for family members to monitor the condition of an elderly relative residing in an aged-care center. Mosiello et al. [47], Rodríguez Lera et al. [65] showed that using augmented reality enhances novice users' experience in remotely driving telepresence robots. Rodríguez Lera et al. [65] attached augmented reality labels on walls and doors, mapped those to direction arrows, then merged with a robot's real video to help users sending control command for to the robot for navigation. None of these works, however, focused on reducing visual clutter or are aimed at a specific device or context.

## 2.1 Controlling Telepresence Robots

Researchers have explored various techniques and technologies to control telepresence robots. Most of these methods, however, are stationary, optimized for specific devices, require extramural devices and sensors, or the use of both hands. Bazzano et al. [6], for instance, investigated the efficiency of manual and semi-autonomous control of a custom-built robot [45] with keyboard and point-and-click video navigation in an office scenario. Kiselev et al. [33] argued that using semi-autonomous navigation enables users to focus on the tasks at hand instead of devoting all their time and attention to operating the robot. Hence, they added features like autonomous mapping and localization, autonomous navigation to the desired point, and automatic docking to a charging station to an application for controlling a Giraff telepresence robot. Kratz and Rabelo Ferriera [35] used keyboard arrow keys to remotely drive a robot. Mosiello et al. [47] enabled controlling a robot and command it to the specific location by using any standard pointing devices. Tonin et al. [71] enabled users with disabilities to remotely drive a telepresence robot in an unfamiliar environment by mapping brain signals left and right-turn commands. Baker et al. [3] proposed a target selection method that uses a 6DoF tracked Vive controllers with an HTC Vive HMD, where users point at a target position and the robot autonomously navigates towards it. Zalud [80] enabled the operator to change the point-of-view of a robot's camera based on where she is looking. She then guided the robot through the environment using a two-hand joystick. Tee et al. [70] developed an application for a custom-built telepresence robot that controls the robot's orientation towards desired targets by using an audio-visual gesture-based attention recognition system. Ainasoja et al. [1] developed a touch-based, a tilt-based, and a touch-tilt hybrid method for controlling telepresence robots with a tablet computer that required the use of both hands. In an incomplete evaluation (prematurely stopped the study due to network failures), the tilt-based approach yielded the fastest task completion time. Based on the existing work in the area and observations in our own work, we created the following list of operations for controlling telepresence robots (Table 2). The list does not include all factors that can facilitate or improve conversations with humans, such as adjusting audio volume, since they are outside the context of this work.

Table 2. Telepresence robot control operation requirements with respective usage scenarios.

| Operation | Scenario |
|---|---|
| R1) Rotate around a point or move backward | Having a conversation; changing direction |
| R2) Adjust the camera | Having a conversation; changing view |
| R3) Speed up on-the-go | The path is clear |
| R4) Slow down on-the-go | Approaching the target or an obstacle |
| R5) Maintain constant speed | Following a person; strolling |

## 2.2 Phone Tilting Gestures

Many have investigated the possibility of using phone tilting gestures to extend the interaction space of mobile devices. Crossan and Murray-Smith [16] studied the variability in tilting a mobile device to up, down, left, and right directions. They found out upward motions have a higher variability than downwards motions. Pinsenschaum and Neff [58] studied tilting depths (angles) towards up, down, left, and right. They found out up and left tilts are relatively deeper than down and right tilts. Rahman et al. [61] investigated the levels of control possible with various phone tilting gestures. They reported that users can control comfortably at least 16 levels when rotating the device along the $z$-axis. They also found out that using a quadratic mapping function for discretization of tilt space significantly improves user performance across all tilt axes. They, however, used a feature phone in their investigation that has different holding position and posture than smartphones. Constantin and MacKenzie [15] compared velocity and position-controlled mapping in a mobile maze game where players can move a virtual ball through the maze by tilting the device. Results revealed that movements with position-controlled are significantly faster than velocity-controlled movements, which supports the findings of a previous study [69]. Baglioni et al. [2] studied faster tilting gestures (JerkTilts) in the context of choice selection. They found out that JerkTilts' recognition rates in an eight-choice selection task are as high as with thumb slides on the touchscreen. They also demonstrated the effectiveness of the method in controlling a music player, text editing, and switching between differet application windows.

Some have proposed novel applications of phone tilting gestures. In an early work, Rekimoto [63] enabled menu item selection by pressing a button and tilting the device to various directions. He also enabled map navigation by pressing a button and tilting the device to respective directions. Harrison et al. [27], Ni et al. [54] built on this work to combine tilting with other phone interactions,

Table 3. TiltWalker interactions and the corresponding actions. Note that the robot used in the studies does not support panning.

| | Robot Control (Navigation Mode) | | Camera Control (Neck Mode) | |
|---|---|---|---|---|
| | **Interaction** | **Action** | **Interaction** | **Action** |
| **Direction & Angle** | *Tap-hold & tilt up* | Move forward | *Double-Tap-hold & tilt up* | Tilt up |
| | *Tap-hold & tilt down* | Move backward | *Double-Tap-hold & tilt down* | Tilt down |
| | *Tap-hold & tilt left* | Turn left | *Double-Tap-hold & tilt left* | Pan left |
| | *Tap-hold & tilt right* | Turn right | *Double-Tap-hold & tilt right* | Pan right |
| **Speed** | *Quick gesture* | Increase speed | *Quick gesture* | Increase speed |
| | *Regular gesture* | Maintain speed | *Regular gesture* | Maintain speed |
| | *Slow gesture* | Decrease speed | *Slow gesture* | Decrease speed |

like squeezing and pinching, while Oakley and Park [55] and Cho et al. [12] extended the work to marking menu and photo scrolling, respectively. In a follow-up study, Liu et al. [40] compared six tilt-based scrolling techniques on a mobile phone, where tilt-based scrolling techniques yielded better performance and caused less fatigue compared to touch in one-handed interaction. van Tonder and Wesson [72] compared tilt-based map navigation with keypad interaction. Results revealed that keypad interaction is more efficient in precise selection but tilt interaction results in a greater perceived controllability, efficiency, and ease of use for navigation tasks.

Hinckley et al. [30] augmented a mobile device with a 2-axis linear accelerometer to automatically switch between portrait and landscape modes based on the phone's orientation. Eslambolchilar and Murray-Smith [21] developed a automatic zooming and scaling method that enabled users to zoom in and out by tilting the device while scrolling. Wigdor and Balakrishnan [77] enabled users to disambiguate between the letters on the keys of a 12-key keypad by tilting the device in one of four directions. Castellucci et al. [8] developed a non-touch tilt-based text entry technique with which users select the keys by controlling a cursor over the keyboard by tilting the device in various directions. Dunlop et al. [20] designed a novel layered text entry method that fades between a full-screen keyboard and a full-screen display of text of the message being typed when tilted. Liu [42] enabled manipulating text property on mobile devices by performing directional tilt gestures. Oakley et al. [56] used up and down tilting motion of a mobile device to control scroll position in a application. Chang et al. [10] studied usage patterns of tilting larger mobile devices toward the thumb, and based on the findings, proposed three techniques for acquiring unreachable screen targets by tilting the device. Dachselt and Buchholz [17] demonstrated how tilt gestures can be used for both step-wise and continuous interaction with both mobile applications and distant user interfaces. Luna et al. [44] proposed a method for interacting with smart TVs via gestures performed by person's wrist using a smartwatch.

In a different line of research, Liu et al. [41] developed an algorithm to create and use personalized hand and wrist gestures using a single three-axis accelerometer attached to a mobile phone. Mäntyjärvi et al. [50] developed a user-dependent gesture recognition model to enable recognition of a moderately large set of gestures with a low number of training repetitions. Di Geronimo et al. [18] developed a framework to develop Web applications featuring motion-based interaction. These works, however, are outside the context of this work. To the best of our knowledge no prior work studied directional phone tilting gestures as users are touching the screen with the thumb of the moving hand or used phone tilting gestures to control out-of-sight robots.

## 3 TILTWALKER INTERACTION & IMPLEMENTATION

For a minimalistic interface that is not cluttered with interactive elements, provides unobstructed views of the two camera feeds, can be operated with one hand, and satisfies the operation requirements listed in Table 2, we developed TiltWalker that exploits phone tilting gestures for controlling an Ohmni telepresence robot (Fig. 2a). To maneuver the robot, the user taps and holds contact on the screen to go to the "navigation mode", then performs directional tilts. Likewise, to control the neck-camera's tilt and pan, the user double-taps and holds contact to go to the "neck mode", then performs directional tilts. The system requires users to maintain touch contact when controlling the robot to reduce the chance of accidental interactions. Releasing touch contact brings the device back to its initial state. Table 3 presents the interactions supported by TiltWalker.

TiltWalker is developed using the default Ohmni WebAPI[1]. It runs a `Node.js` server on the robot using a docker based on Ubuntu 18.04. When the server receives requests from the client interface, it generates commands executable by the robot, then sends those directly to the robot using a socket. The application uses WebSocket connections for streaming live video feeds from the cameras. The final interface, described in Section 8, was refined based on rigorous lab trials and studies, discussed in the following sections.

## 4 PILOT STUDY: LAGS AND DELAYS

The purpose of this study was to investigate whether TiltWalker's Web architecture causes any extra lags or delays in the robot's response compared to a native client application. Three participants took part in this pilot (M = 30 years, SD = 1.7). They all identified themselves as women. Two of them had never used a telepresence robot, while the other attended an event remotely using a telepresence robot. They were all experiences smartphone users.

### 4.1 Apparatus

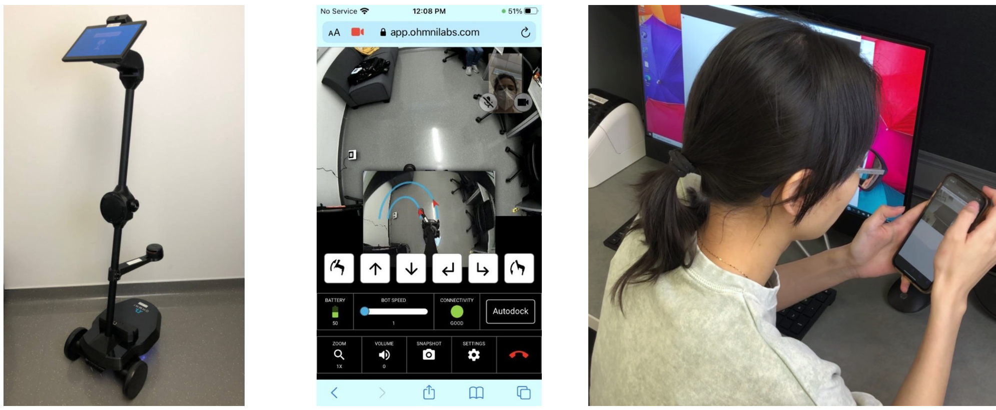
The study used an Ohmni telepresence robot that includes a 4K camera with 13 MP snapshot and superzoom capabilities and a 256.54 mm HD IPS touchscreen (Fig. 2a). A Web application was developed to enabled moving the robot in the four directions and tilting the neck up and down using dedicated virtual buttons on the top of the interface (Fig. 2c). The application was viewed on a Firefox for Android Web browser v97.2.0 on a Samsung Galaxy S6 edge smartphone (132 g, $142.1 \times 70.1 \times 7$ mm, 71.5 $cm^2$ display) at $1440 \times 2560$ pixels resolution and ~577 ppi density. The device included an Invensense MPU6500 v1.0 accelerometer sensor with the maximum range of 39.23 $m/s^2$. It ran on Android 7.0 Nougat.

### 4.2 Design & Procedure

The study used a within-subjects design. It was divided into two parts. In the first part, participants performed robot control tasks from the following set {*turn right*, *turn left*, *move forward*, *move backward*, *tilt the neck up*, *tilt the neck down*}. In the second part, participants drove the robot in a straight line in three trials for 30 seconds each. In summary, the design was: 3 participants × 6 controls tasks *randomized* × 5 times + 1 driving task *30 seconds* × 5 times = 105 trials.

Upon arrival, we described the study procedure to the participants and collected their informed consent forms. Then, they completed a short demographics and technology usage questionnaire. The study started shortly after that. First, they performed the six robot control tasks in a random order by using the default and the custom applications (Fig. 2). The methods were counterbalanced to reduce the effect of learning. Then, they drove the robot for 30 seconds in a straight line at the constant speed of 0.5 mph. During the study, the robot was out of sight in a different room about

[1]WebAPI - Ohmni Developer Manual, https://docs.ohmnilabs.com/webapi.




(a) Ohmni Robot (b) Default Web application (c) A user using the custom Web application

Fig. 2. The devices and the applications used in the studies.

60 meters away from the participants. Hence, participants had to rely exclusively on the video feeds to perform the tasks. Both the robot and the applications were connected to a reliable Wi-Fi network. We did not record any network outage or dropouts during the study. We calculated the following metrics.

- **Lag** (milliseconds) measures the delay between a user action and the robot's corresponding reaction. It was measured as the time difference between the user issuing a command and the robot executing the command.
- **Time offset** (milliseconds) measures lag in continuous actions. It is calculated by comparing the expected and the actual stop points of the robot after a continuous movement, divided by the robot's mean velocity: $\Delta t = \frac{\Delta x}{\bar{v}}$, where $\Delta t$ is time offset, $\Delta x$ is the change in position (or displacement) from the beginning to the final position, and $\bar{v}$ is the average velocity of the robot.

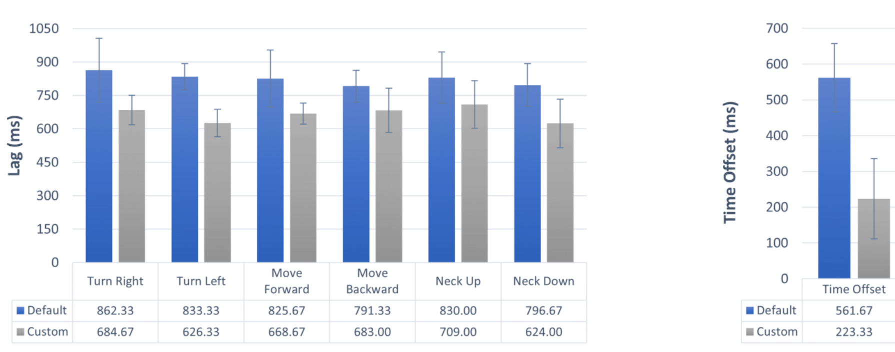



Fig. 3. Average lag (ms) and time offset (ms) for the native and the Web applications. Error bars represent ±1 standard deviation (SD).

### 4.3 Results & Discussion

We did not conduct statistical tests on the data due to the small sample size. On average, the Web application yielded much lower lags for the tasks than the native application (about 20% lower, Fig. 3). The time offset was also 60% lower than the native application. While the exact reason for this performance gain is unknown, we speculate this is for the lightweight nature of our Web architecture. These results must encourage the development of Web applications rather than individual native applications aimed at different operating systems, saving much time and effort. One interesting observation in the study is all participants held the device in portrait position with the dominant hand and interacted using the index finger of the other hand (Fig. 2c), although they were informed that they could use any position, posture, and orientation to interact with the applications. Developers must take this into consideration when designing client applications for telepresence robots.

## 5 USER STUDY 1: HUMAN FACTORS

The purpose of this study was to compare tilting with twisting gestures, and to identify the most comfortable and effective directional tilting and twisting angles in both standing and seated positions. Although there are similar studies in the literature, they either used feature phones [16, 56, 58] or wearable devices [20, 44, 54] that have different holding positions and postures than smartphones or did not explore standing position. These studies focused on the highest possible tilting angles without the consideration for comfort or the legibility of the content. Tilting the device too much can make the content illegible, which is undesired in continuous interactions like when driving a robot. These studies also did not require users to maintain thumb contact with the display, which can also affect wrist movements [13, 75].

### 5.1 Participants

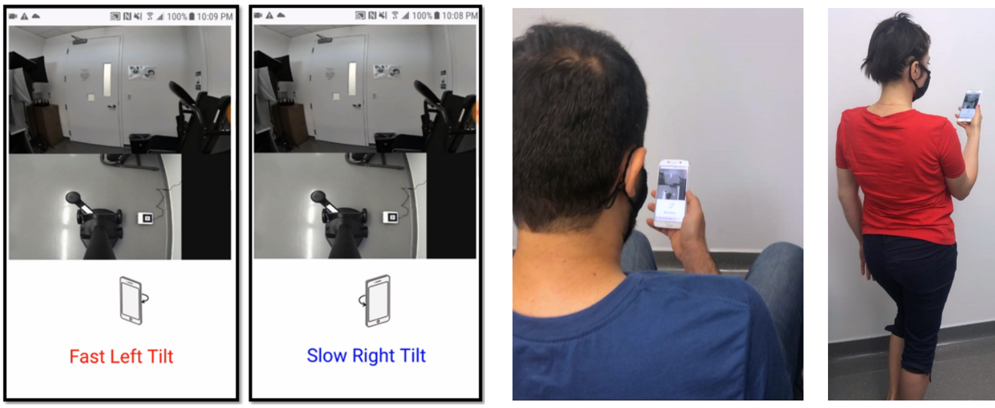
Twelve participants took part in this study. Their age raged from 26 to 39 years (M = 32, SD = 4.3). Seven of them identified themselves as female and five as male. All of them were right-handed and experienced smartphone users (M = 13.3 years' experience, SD = 6.7). They all received US $15 for participating in the study.

### 5.2 Apparatus

The study used the same mobile device as the pilot study (Section 4.1). However, a new Web application was developed that divided the vertical display into three equal parts. The top and the middle parts displayed static images of the two camera views of the robot, and the bottom part provided users instructions on which gesture to perform and in which speed. The instructions used color-coded textual (blue: slow, black: regular, red: fast) and graphical (clip art) visual cues to avoid any confusions (Fig. 4a). Live camera feeds were not provided since controlling the robot was not required.

### 5.3 Design & Procedure

The study investigated four directional tilt two directional twist gestures when holding the device with the dominant hand and touching the display with the thumb of the same hand: *up tilt*: tilting the device towards the body, *down tilt*: tilting the device away from the body, *left tilt*: tilting the device to the left, *right tilt*: tilting the device to the right, and *left twist* and *right twist*: twisting the wrist to the left and the right (not to confuse with rotating the wrist). The study used a within-subjects design. Twelve participants performed the six gestures eight times in three different speeds (slow, regular, fast) and two positions (seated, standing). The two positions were counterbalanced,




(a) The application used in the study (b) Seated position (c) Standing position

Fig. 4. (a) The applications used in the study. The bottom part of the application provides textual and graphical instructions on which gesture to perform and in which speed, (b, c) two participants performing the tasks in seated and standing positions, respectively.

while the three speeds and the six gestures were presented in a random order. In summary, the design was: 12 participants × 2 positions, counterbalanced × 3 speeds, randomized × 4 tilting and 2 twisting gestures, randomized × 8 trials = 3,456 trials in total. Fig. 5 illustrates the gestures investigated in the study.

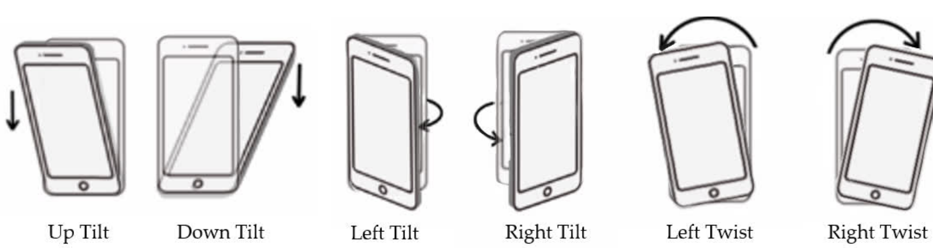



Fig. 5. The six directional gestures explored in the first user study, relative to the initial angles represented in greyed out overlays.

The study was conducted in a quiet lab. The telepresence robot was out of sight in a different room about 60 meters away from the lab. Upon arrival, we described the study procedure to the participants and collected their informed consent forms. Then, they completed a short demographics and technology usage questionnaire. We then demonstrated the experimental application to the participants and enabled them to practice with it for 1–2 minutes. Participants were instructed to hold the device in the portrait position with the dominant hand. The application displayed one task at a time, for example “fast left tilt”. Participants were asked to initiate the task by touching the screen with the thumb without occluding the camera views and tilting (or twisting) the device to the instructed direction until the angle was uncomfortable or the display content were illegible, whichever came first. Once they completed a task, the next task was displayed. In the seated condition, participants could either sit at a desk and rest their elbow on the desk or away from the desk resting their elbow on the thigh. Most participants chose to sit at the desk (N = 11). In the standing condition, they could stand anywhere in the room (N = 7 chose to lean on the wall or a desk). When done with the study, participants completed a short questionnaire and participated in an informal interview session.

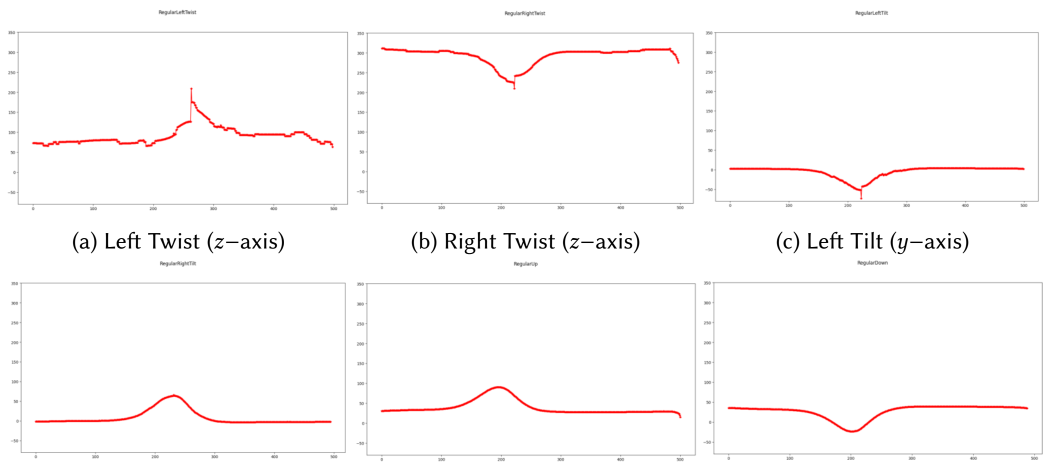
Table 4. Average highest peak range (degree) and angular velocity (degree/sec). The bold values are used to detect tilt gestures and to set parameters in control-display functions (peaks in the $x-$axis: up, down; $y-$axis: left, right; $z-$axis: left and right twists).

| Speed | Gesture | $x-$**axis Range (Min–Max)** | | $y-$**axis Range (Min–Max)** | | $z-$**axis Range (Min–Max)** | |
|---|---|---|---|---|---|---|---|
| | | *Highest Peak (degree)* | *Angular Velocity (degree/sec)* | *Highest Peak (degree)* | *Angular Velocity (degree/sec)* | *Highest Peak (degree)* | *Angular Velocity (degree/sec)* |
| **Slow** | *Up Tilt* | **32, 112** | **0, 5.3** | −14, 89 | 0, 2 | 0, 351 | 0, 2.2 |
| | *Down Tilt* | **32, −48** | **0, 3** | −5, 43 | 0, 2 | 0, 358 | 0, 1.5 |
| | *Left Tilt* | 8, 164 | 0, 1.5 | **0, −80** | **0, 3.1** | 0, 357 | 0, 2.1 |
| | *Right Tilt* | 77, 0 | 0, 2.4 | **0, 80** | **0, 3.2** | 0, 357 | 0, 2.6 |
| | *Left Twist* | 62, −2 | 0.1.8 | 0, −89 | 0, 1.7 | **10, 360** | **0, 2.3** |
| | *Right Twist* | 64, −18 | 0, 1.1 | 5, −88 | 0, 1 | **320, 0** | **0, 1.8** |
| **Regular** | *Up Tilt* | **32, 112** | **0, 7** | −2, 89 | 0, 3.5 | 1, 355 | 0, 2 |
| | *Down Tilt* | **32, −48** | **0, 5.5** | −2, 73 | 0, 1 | 0, 358 | 0, 4 |
| | *Left Tilt* | 11, 166 | 0, 2 | **0, −80** | **0, 4.2** | 4, 354 | 0, 3.2 |
| | *Right Tilt* | 71, 0 | 0, 2.3 | **0, 80** | **0, 5** | 355, 6 | 0, 2 |
| | *Left Twist* | 40, −8 | 0, 1.6 | 1, −88 | 0, 1.9 | **10, 360** | **0, 2.2** |
| | *Right Twist* | 47, −28 | 0, 2.9 | 5, −68 | 0, 1.9 | **320, 0** | **0, 3.1** |
| **Fast** | *Up Tilt* | **32, 112** | **0, 10** | −4, 89 | 0, 4 | 0, 352 | 0, 5 |
| | *Down Tilt* | **32, −48** | **0, 13** | −4, 360 | 0, 6.5 | 0, 356 | 0, 3 |
| | *Left Tilt* | 9, 169 | 0, 4.5 | **0, −80** | **0, 6** | 0, 354 | 0, 2.5 |
| | *Right Tilt* | 357, 0 | 0, 2.3 | **0, 80** | **0, 5.3** | 350, 6 | 0, 2.1 |
| | *Left Twist* | 44, −7 | 0, 4 | 0, −89 | 0, 3 | **10, 360** | **0, 4.7** |
| | *Right Twist* | 51, −22 | 0, 2.8 | 10, −89 | 0, 2.9 | **320, 0** | **0, 3.8** |

### 5.4 Results & Discussion

A paired-sample t-test revealed that there was no significant effect of position (seated, standing) on the height peak ($t_{(1727)} = -0.14, p = 0.89$). We, therefore, do not differentiate the two positions in the following results.

*5.4.1 Highest Mean Peaks.* A repeated-measures ANOVA identified a significant effect of task on highest peak in the respective axis ($F_{5,55} = 56.79, p < .0001$). A Tukey-Kramer Multiple-Comparison test revealed that left and right twists yielded the height peaks in the $z-$axis (182° and 183°), followed by up and right in the $x-$axis (89° and 64°), and down and left in the $y-$axis (−23° and 14°). There was no significant effect of speed ($F_{2,22} = 2.11, p = 0.15$). Table 4 presents the average highest peak range (degree) and angular velocity (degree/sec) in the user study. It is clear from the results that directional tilt gestures can easily be detected by observing peaks in the three axis. This can also be seen in Fig. 6 that illustrates the average peaks for all tilt gestures over all trials for all participants during the regular speed condition. For this visualization, we first calculated the average peak time, then shifted the data from each trial to match the time. The accelerometer data was recorded in every 10 ms, the average rotation data was also calculated for the same interval.



(a) Left Twist ($z$−axis) (b) Right Twist ($z$−axis) (c) Left Tilt ($y$−axis)

(d) Right Tilt ($y$−axis) (e) Up Tilt ($x$−axis) (f) Down Tilt ($x$−axis)

Fig. 6. Visual representations of average peaks for all examined tilt gestures over all trials for all participants in the regular speed condition. For this, we shifted the data from the trials to match the average peak time.

*5.4.2 Subjective Feedback.* Participants found all gestures relatively easy to perform in terms of physical and mental efforts (ratings from 1–3 on a 5-point Likert scale). But they found the up and down gestures the easiest, followed by the left tilt and twist, and the right tilt and twist. A Friedman test identified this difference to be statistically significant ($\chi^2 = 102.2, df = 5, p < .0001$). Based on the findings and to maintain the "directional" metaphor, we decided to use the left, right, up, and down directional tilts in TiltWalker. Fig. 7 presents median perceived physical and mental efforts in performing the gestures. In the seated condition, almost all participants chose to sit at the desk (N = 11). In the standing condition, most of them chose to lean on the wall or a desk (N = 7) Interestingly, most participants (N = 7) responded that they preferred the standing position as it enabled them to focus more on the tasks. Three preferred the seated position for comfort and fearing that they would drop the device, while the remaining two were neutral about it.

## 6 CONTROL-DISPLAY (CD) MAPPING AND GAIN

Traditionally, CD gain represents the relationship between the movements of a physical input device (e.g., a mouse) and the movements of its corresponding virtual object (e.g., the pointer). In contrast, in this work, the movements of a physical object (i.e., the smartphone) are mapped to the movements of another physical object (i.e., the robot). We used a systematic approach to identify an appropriate CD mapping function for the project, inspired by prior works exploring novel functions [7, 14, 24, 32, 64, 66].

First, we conducted a thorough review of the literature to identify the functions that could be used in the scenario. We considered both spatial, physical, and dynamic relationships between the control and the display. We could not find direct research on mapping input device movements to both the direction and the velocity of physical devices. Most works that enabled controlling physical devices, such as robots, power wheelchairs, and drones, either supported only directional movements or used separate controls, like buttons and joysticks, for determining the velocity [11, 34, 52, 59]. Yet,

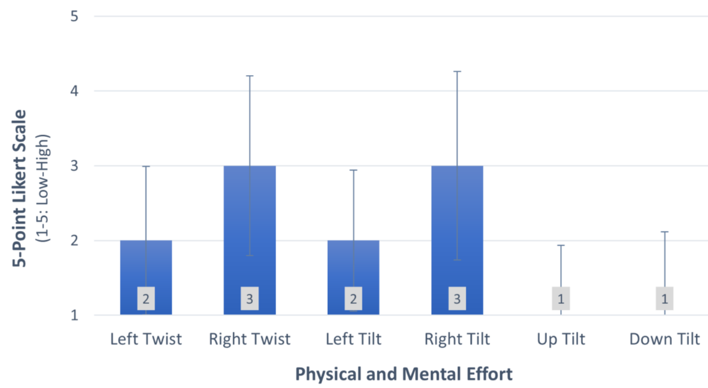



Fig. 7. Median perceived physical and mental effort in performing the examined tilt gestures on a 5-point Likert scale (1–5: low–high). Error bars represent ±1 standard deviation (SD).

we created a list of functions that account for both the direction and the velocity of the display. We then categorized them into position-controlled and velocity-controlled mapping, where the former maps the input device's position to velocity, while the latter maps the speed in which the position is changed to the velocity of movement. We disregarded all functions that use machine learning and probabilistic approaches since we desired to identify functions that could be used without training data. Second, we tested all functions in lab trails and adjusted the relationships based on the parameters determined in first user study. Based on the findings, we removed all linear functions since mapping noisy accelerometer data did not result in smooth movements of the robot. Besides, due to the nature of these functions, a slight tilt of the phone tend to result in a significant change in the robot's velocity. Using a CD gain $< 1$ resolved these issues to some extent but made the interactions substantially slower. Ainasoja et al. [1] also reported similar issues with linear functions. Finally, we finalized two position and two velocity-controlled functions that felt more natural in the trials, viz. did not take too much time or effort to change direction, reach an intended speed, or to maintain a constant speed. We picked two sigmoid functions inspired by Ha and Woo [25] that provides valuers within a range, which is more appropriate since both tilt angles and the robot's speed have upper and lower-bounds. We also picked one quadratic function that magnifies and one logarithmic function inspired by Hayes and Adams [29] that reduces the impacts of tilt angles on the robot's velocity. We kept these conflicting functions to find out which strategy thrives in actual robot maneuvering scenarios. We present the four best performed functions below.

$$V^t_{Robot} = (\alpha R^t)^2 \qquad \text{\underline{Position-Controlled F1}: Quadratic} \quad (1)$$

Where $V^t_{Robot}$ is the robot's velocity at time $t$, $R^t$ is a variable that changes based on the phone's orientation displacement at time $t$, and $\alpha$ is a constant value.

$$V^t_{Robot} = A\left(\frac{1}{1+e^{-(BR^t - C)}} - \frac{1}{3}\right) \qquad \text{\underline{Position-Controlled F2}: Sigmoid} \quad (2)$$

Where $A$, $B$, and $C$ are constants. The update rule of the parameter $R^t$ is $R^{t-1} + \lfloor(\beta^t - \beta^{t-1})/D\rfloor$, where $\beta^t$ is the phone's tilt orientation at time $t$ and $D$ is the discretization parameter to discretize the phone's continuous tilt space ($D = 10$). It ignores any phone tilt displacement where $\Delta\beta < 10$. The initial value of variable $R$ is $R^0 = 0$.

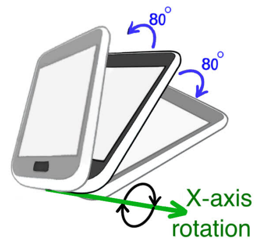



Fig. 8. Optimal ranges for up and down tilts.

Results of the first study (Table 4) revealed that for the up tilt, the maximum phone angle is likely to be within the range $(32°, 112°)$, and within $(32°, -48°)$ for the down tilt. Hence, we use 32° as the phone's initial orientation in the $x-$axis, from which the maximum tilt for up and down is 80° (Fig. 8). Likewise, for the right tilt, the maximum phone angle is usually within the range $(0°, 80°)$, and for the left tilt it is within the range $(0°, -80°)$. Hence, $(0°)$ is used as the phone's initial orientation around the $y-$axis, from which the maximum tilt is $(80°)$ to the right and the minimum is $(-80°)$ to the left. Note that we discretize the phone's continuous tilt space into discrete units of an equal size of 10. Based on this, $R^t$ is estimated to hold an integer value between $(-8, 8)$. In robot's motion space, we set the maximum and the minimum speed values as: $V^{min}_{Robot} = 0$ and $V^{max}_{Robot} = 10$, Which is equal to 2 mph. Instead of using negative values for $V^t$ we sent the motion command to robot for moving in the opposite direction. We then calculate the values of the constants $\alpha$, $A$, $B$, and $C$ by mapping the upper and the lower bounds of $R^t$ to match with $V^{min}_{Robot}$ and $V^{max}_{Robot}$, leading to $\alpha = 2/3$, $A = 200$, $B = 1/42$, and $C = log_e 2$.

$$V^t_{Robot} = R^t \frac{\omega^t \log_e 10}{c_1 \log_e (\omega^t)^{c_2} + 1} \qquad \text{Velocity-Controlled F1: Logarithmic} \quad (3)$$

$$V^t_{Robot} = A \left( \frac{1}{1 + e^{-(B\omega^t - C)}} - \frac{1}{3} \right) R^t \qquad \text{Velocity-Controlled F2: Sigmoid} \quad (4)$$

Where $\omega^t$ is the angular velocity of wrist at time $t$ and $c_1$, $c_2$, $A$, $B$ and $C$ are constant values. Results of the first study (Table 4) revealed that for the up tilt, the maximum angular velocity is likely to be within the range $(0, 10)$ deg/s and for the down tilt within the range $(0, 13)$ deg/s. Likewise, for the right tilt, the maximum angular velocity of tilt is likely to be within the range $(0, 5.3)$ deg/s and for the left tilt, within $(0, 6)$ deg/s. Like the position-controlled functions, we use these details to calculate the constant values as: $c_1 = 5$, $c_2 = 1/42$, $A = 9$, $B = 1/22$, and $C = log_e 2$.

## 7 USER STUDY 2: MAPPING FUNCTIONS

We compared the performance of the four CD mapping functions described above in a user study to identify the best function to be used with the final TiltWalker interface.

### 7.1 Participants

Twelve participants took part in the study. Their age raged from 22 to 41 years (M = 29.2, SD = 6.1). Four of them identified themselves as female and eight as male. Eleven of them were right-handed, one refused to respond to this question. All of them were experienced smartphone users, i.e., owned and frequently used smartphones for over six years (M = 10.3, SD = 1.9). None of them participated in the previous studies. They all received US $15 for their time.

### 7.2 Apparatus

The study used the same apparatus as the previous studies. However, we developed a new Web application that displayed the live feeds of the froward-facing and the downward-facing cameras on the top and bottom parts of the display (Fig. 9a). The application did not provide users with feedback on speed or direction change.

### 7.3 Design

The study used a within-subjects design. All participants navigated the robot to a target and back twice using the four mapping functions. Both the function and the direction were counterbalanced. In summary, the design was: 12 participants × 4 functions *counterbalanced* × 2 directions *counterbalanced* × 2 trials = 192 trials in total. The study recorded the following performance metrics.

- **Task completion time** (seconds) signifies the average time users took to navigate the robot to the target from the initial position.
- **Error rate** signifies the average error committed per task. An error was recorded when the robot deviated one feet from its path (~0.3 m), moved or turned to wrong directions, or stepped over an obstacle.

In addition, we collected user responses to 5-point Likert scales asking them to rate the speed, accuracy, learnability, and ease-of-use of the examined functions as perceived in the study.

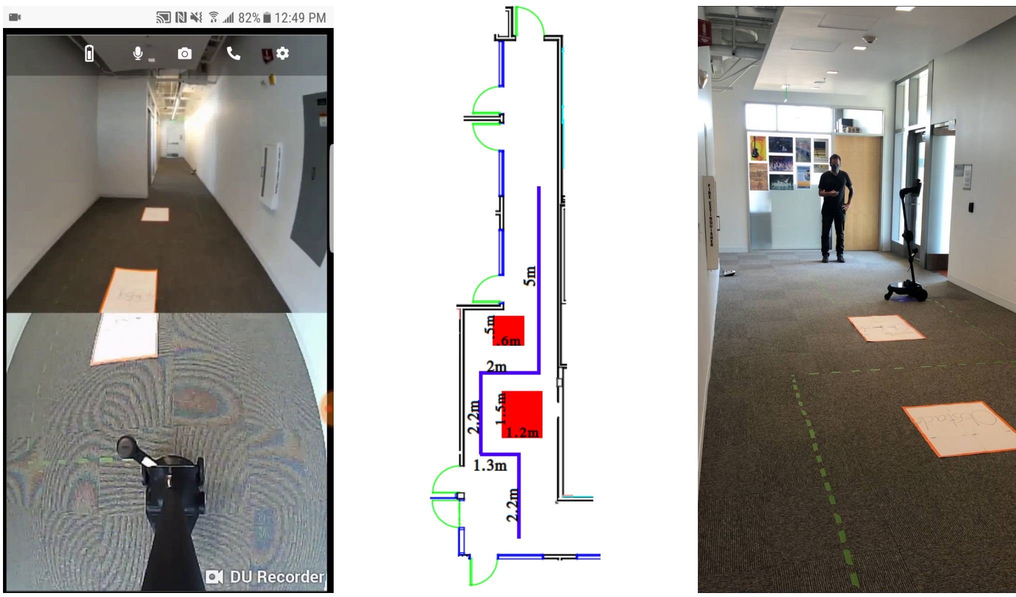
### 7.4 Procedure

The study was conducted in a quiet, empty corridor. The telepresence robot was kept within sight since the experimental applicant did not provide feedback on direction or speed change. Upon arrival, we described the study procedure to the participants and collected their informed consent forms. Then, they completed a short demographics and technology usage questionnaire. We then demonstrated the experimental application and enabled the participants to practice with it for 1–2 minutes. Participants were instructed to hold the device in the portrait position with the dominant hand. In the study, participants navigated the robot through an obstacle path to a target and brought it back to them (in reverse motion) using the four mapping functions. This task covered operation requirements R1–4 listed in Table 2. The application used the tilt-based interaction approach described in Section 3, except for in the position-controlled condition, where the angle of the device determined the speed of the robot. The functions were counterbalanced using a Latin square. The direction of the path was also counterbalanced, where half of the participants started with bringing back the robot to them, while the other half started with navigating it to the target. Fig. 9b illustrates the obstacle path with scales. An experimenter kept manual logs of the robot's deviation from the path, movements and turns to the wrong direction, and stepping over an obstacle. When done with the study, participants completed a short questionnaire and participated in an informal interview session.

### 7.5 Results

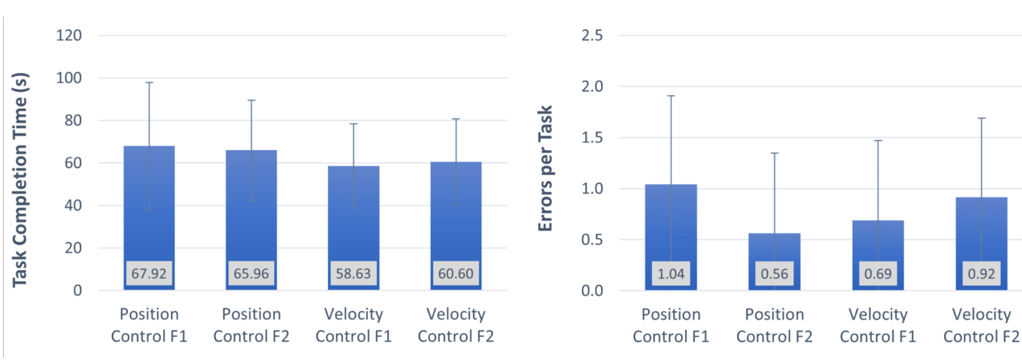
A Shapiro-Wilk test revealed that the response variable residuals were normally distributed. A Mauchly's test indicated that the variances of populations were equal. Hence, we used a repeated-measures ANOVA for all quantitative within-subjects factors. We used a Friedman test for the questionnaire data.

*7.5.1 Task Completion Time.* An ANOVA identified a significant effect of function on task completion time ($F_{3,11} = 4.19, p < .05$). The velocity-controlled F1 (Eq. 3) yielded the fastest task completion




(a) (b) (c)

Fig. 9. (a) The custom application and (b) the obstacle path used in the study, (c) a participant taking part in the study.




(a) Task completion time (seconds) (b) Error rate (errors per task)

Fig. 10. Average task completion time and error rates for the examined functions. Error bars represent ±1 standard deviation (SD).

time (58.6 seconds), followed by velocity-controlled F2 (Eq. 4), position-controlled F1 (Eq. 1), and position-controlled F2 (Eq. 2). Fig. 10a illustrates this.

*7.5.2 Error Rate.* An ANOVA failed to identify a significant effect of function on error rate ($F_{3,11} = 2.05, p = .13$). On average position-controlled F2 (Eq. 2) and velocity-controlled F1 (Eq. 3) yielded lower error rates than the other functions (Fig. 10b).

*7.5.3 Subjective Evaluation.* A Friedman failed to identify a significant effect of function on perceived speed ($\chi^2 = 6.98, df = 3, p = .07$), accuracy ($\chi^2 = 5.72, df = 3, p = .13$), learnability

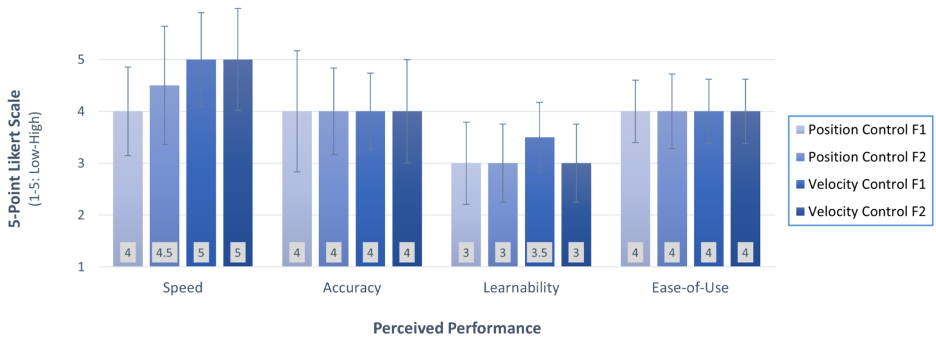



Fig. 11. Median perceived speed, accuracy, learnability, and ease-of-use of the examined CD mapping functions on a 5-point Likert scale (1–5: low–high). Error bars represent ±1 standard deviation (SD).

($\chi^2 = 4.80, df = 3, p = .19$), or ease-of-use ($\chi^2 = 2.40, df = 3, p = .49$). Fig. 11 illustrates median perceived performance of the mapping functions.

### 7.6 Discussion

Results revealed that participants performed the tasks significantly faster and yielded a relatively lower error rate with the velocity-controlled F1 (Eq. 3) than the other functions. The position-controlled functions (Eqs. 1, 2) were the slowest since they mapped specific angles to specific velocity of movements, which were difficult for the participants to master. Participants also perceived these functions as relatively slower than the other functions and found them much difficult to learn (Fig. 11). The sigmoid velocity-controlled function (Eq. 4) was much faster but was more error prone than the logarithmic function since it amplified the movements, which often caused unexpected rapid changes in the robot, requiring the participants to readjust the velocity. The logarithmic function (Eq. 3) mapped tilting speed to the robot's speed which was much easier for the participants to learn and provided much smoother changes in velocity than the other functions. We, thus, use it in TiltWalker.

## 8 THE TILTWALKER INTERFACE

The TiltWalker interface was designed in an iterative design process, where the most recent version was tested in lab trials then refined based on the findings. The final interface divides the display into two parts. The top part displays the forward-facing camera view and the bottom part displays the downward-facing camera view. A menu bar is placed on the top of the display, which contains icons for three most frequently options in telepresence robots: mic, camera, and call. Users could place other options in the menu by pressing the settings icon. The menu also provides feedback on speed and direction change and the battery level. It uses a translucent background, thus does not fully occlude the camera view. Users could also drag the menu to the bottom of the screen. Instead of alert windows and invasive labels, it uses a virtual gauge to provide feedback on both speed and direction change. The needle of the gauge indicates rotation direction and the colors of the panel indicate speed. In the color wheel, cool to hot colors (blue to read) represent the lower and the upper-bound of speed, respectively. Section 3 described the interactions supported by the interface with the velocity-controlled F1 (Eq. 3). Fig. 12 presents the final interface of TiltWalker.

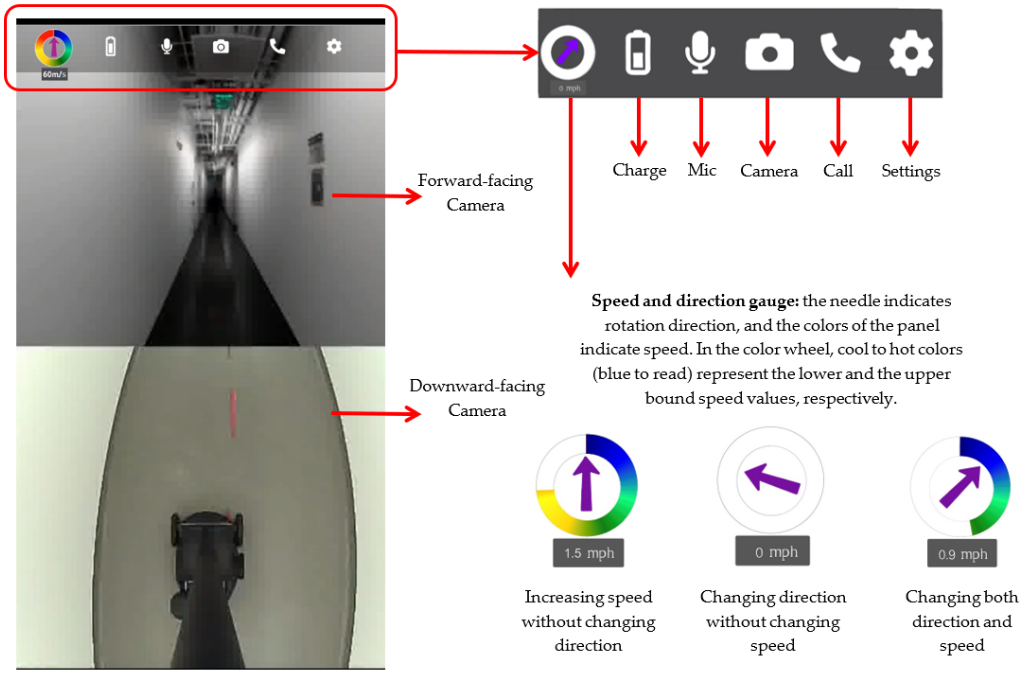



Fig. 12. The final user interface and visual feedback used in TiltWalker.

## 9 USER STUDY 3: COMPARATIVE EVALUATION OF TILTWALKER

We compared TiltWalker with the default Ohmni Lab mobile application in a user study. We also considered including an academic mobile solution as a baseline but decided against it as they are either optimized for tablets [1] or special usage scenarios like healthcare [78] or museum exploration [45]. Adopting these to our device and context is a time-consuming task and outside the scope of the work.

### 9.1 Participants

Twelve participants took part in the study. Their age raged from 18 to 38 years (M = 25.3, SD = 5.1). Four of them identified themselves as female and eight as male. Nine of them were right-handed and three were left-handed mobile users. They all were experienced smartphone users, i.e., owned and frequently used smartphones for over five years (M = 10.9, SD = 4.2). None of them participated in the previous user studies. They all received US $15 for their time.

### 9.2 Apparatus

The study used the same apparatus as the previous studies. It used the final TiltWalker application with the interaction and interface described in Sections 3 and 8, respectively.

### 9.3 Design

The study used a within-subjects design. All participants followed a person (the walker) through a corridor from three different start points while trying to maintaining the same distance between the robot and the walker using the default application and TiltWalker. The method was counterbalanced and the start points were randomized. In summary, the design was: 12 participants × 2 methods (default, TiltWalker), *counterbalanced* × 1 path × 3 start position (1.5 m, 4.5 m, 9 m), *random* = 72 trials in total.

We recorded the same performance metrics as Section 7.3. But in the error rate calculation, 0.5 m offset (~1.5 feet) from the expected distance between the walker and the robot was counted as one error instead of 0.3 m deviation from straight line (see Section 9.4). We also collected user responses to 5-point Likert scales asking them to rate the speed, accuracy, learnability, and ease-of-use of the examined methods as perceived in the study. Participants also rated their "confidence" in using the apps, and preference of the "feedback" methods (textual vs. virtual gauge and unobstructed camera feeds) and "interaction" techniques (tap vs. tilt). They also completed a NASA-TLX questionnaire [51]. For analysis, we calculated raw TLX scores by individual sub-scales, which is a common practice in the literature [28].

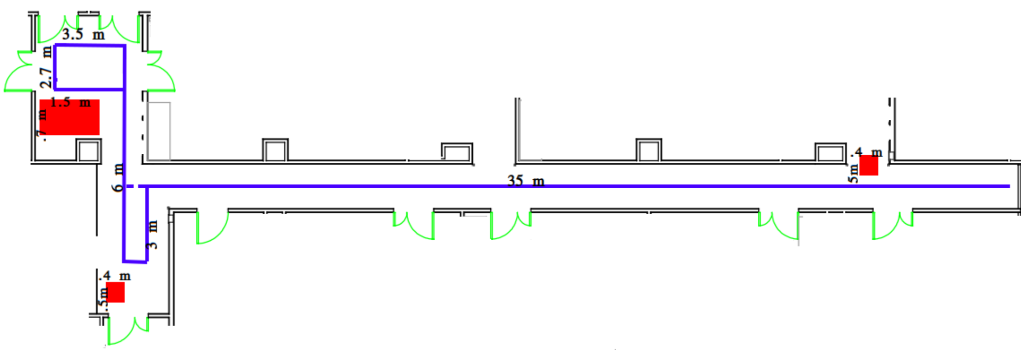


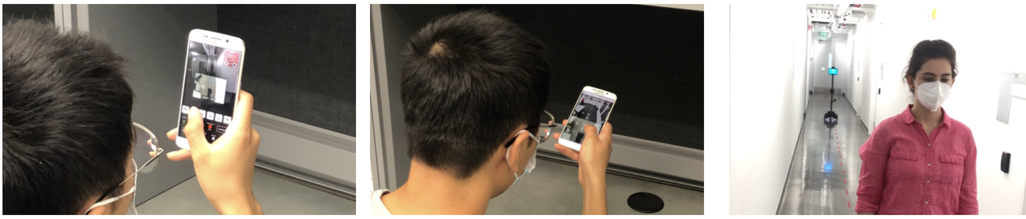

Fig. 13. The obstacle path (with scales) used in the final study.

## 9.4 Procedure

The study was conducted in a quiet lab. The telepresence robot was out of sight in a corridor about 100 meters away from the lab. Upon arrival, we described the study procedure to the participants and collected their informed consent forms. Then, they completed a short demographics and technology usage questionnaire. We then demonstrated the experimental application to the participants and enabled them to practice with it for 3–5 minutes. Participants were instructed to hold the device in the portrait position with the dominant hand. The use of the non-dominant hand was restricted in the TiltWalker condition to examine one-hand interactions. In the default condition, they could use both hands. In the study, participants drove the out-of-sight robot to follow the walker along a corridor to a destination (Fig. 14). There were three start points: 1.5 m, 4.5 m, 9.0 m away from the walker. Participants were instructed to maintain a constant distance of 1.5 m between the robot and the walker. There were also three obstacles of different sizes placed on the floor, which users were instructed to avoid (Fig. 13). The purpose of these restrictions were to enforce them to variate the camera view and the robot's direction and speed to keep up with the walker. This task, thus, covered all control operation requirements listed in Table 2. When done with the study, participants completed the questionnaires and participated in an informal interview session.

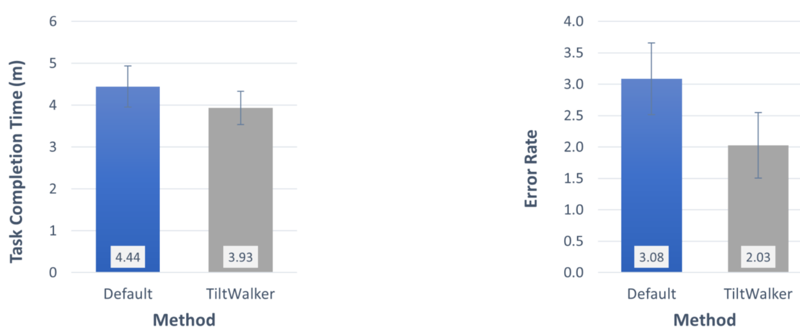
## 9.5 Results

A Shapiro-Wilk test revealed that the response variable residuals were normally distributed, thus we used a paired-sample t-test for quantitative within-subjects factors. We used a Wilcoxon Signed-Rank test for the subjective data.



(a) Default (b) TiltWalker (c) Following the walker

Fig. 14. A participant interacting with (a) the default application and (b) TiltWalker to (c) follow a walker through a corridor.




(a) Task completion time (minutes) (b) Error rate (errors per task)

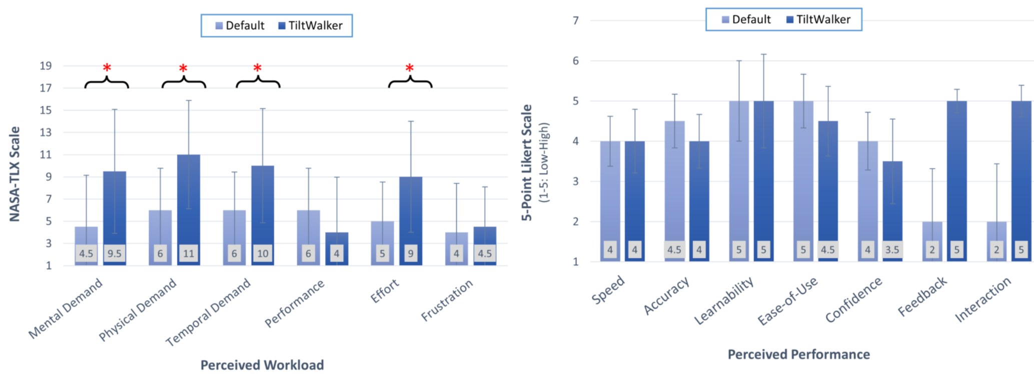
Fig. 15. Average task completion time and error rates for the two methods. Error bars represent ±1 standard deviation (SD).

*9.5.1 Task Completion Time.* A t-test identified a significant effect of method on task completion time ($t_{(11)} = 6.31, p < .0001$). The average task completion times for the default and TiltWalker were 4.44 s (SD = 0.49) and 3.93 s (SD = 0.40), respectively. Fig. 15a illustrates this.

*9.5.2 Error Rate.* A t-test identified a significant effect of method on error rate ($t_{(11)} = 4.79, p < .001$). The average error rates for the default and TiltWalker were 3.08 (SD = 0.57) and 2.03 (SD = 0.52), respectively. Fig. 15b illustrates this.

*9.5.3 Perceived Workload.* A Wilcoxon Signed-Rank test identified a significant effect of method on mental demand ($z = -2.03, p < .05$), physical demand ($z = -2.67, p < .01$), temporal demand ($z = -1.99, p < .05$), and effort ($z = -2.21, p < .05$). However, no significant effect was identified on performance ($z = -0.85, p = .40$) or frustration ($z = -0.27, p = .79$). Fig. 16a illustrates median perceived workload of the methods.

*9.5.4 Perceived Performance.* A Wilcoxon Signed-Rank test failed to identify a significant effect of method on perceived speed ($z = -0.82, p = .41$), accuracy ($z = -1.41, p = .16$), learnability ($z = -1.29, p = .20$), ease-of-use ($z = -1.63, p = .10$), and confidence ($z = -1.63, p = .10$). However, participants preferred TiltWalker's feedback ($z = -2.95, p < .005$) and tilt-based interaction ($z = -1.67, p < .01$) significantly more than those of the default method (Fig. 16b).




(a) NASA-TLX (1–20: very low–high/perfect–failure) (b) User preference Likert scale (1–5: low–high)

Fig. 16. Median raw NASA-TLX and user preference ratings of the two methods. Error bars represent ±1 standard deviation (SD). Red asterisks indicate statistically significant differences.

## 9.6 Discussion

TiltWalker was significantly faster. Participants completed the tasks about 11.5% faster with TiltWalker than the default method. A deeper analysis revealed that participants' speed with TiltWalker increased by 16% in the last trial compared to the first. This pattern fitted well to a power trendline ($R^2 = 0.88$). This suggests that learning occurred with TiltWalker even in the short duration of the study. The default method, in contrast, yielded almost the same task completion time in the first and the last trials (4.44 s vs. 4.40 s), which naturally did not fit well to a power trendline ($R^2 = 0.06$). TiltWalker was also significantly more accurate. To complete the task, participants had to perform all operations listed in Table 2. Analysis revealed that they were able to perform these 34% more accurately than the default method. They were more proficient in avoiding obstacles and adjusting or maintaining the direction and speed with TiltWalker than the default method. We also observed learning in this aspect. Participants were 35% more accurate in the last trial compared to the first, which fitted well to a power trendline ($R^2 = 0.90$). No such trend was observed with the default method. These results suggest that the performance of TiltWalker could improve further with practice.

The workload questionnaire revealed that TiltWalker required significantly higher mental, physical, and temporal demands than the default method. We anticipated this since participants were new to the tilt-based interaction method. Based on prior works on workload [76], we speculate that these demands will reduce with practice. Interestingly, participants did not find any significant difference between the two methods' performance, thus were not frustrated using the method (Fig. 16a). In the preference questionnaire, participants rated the two methods somewhat comparably on speed, accuracy, learnability, ease-of-use, and confidence, but rated TiltWalker's tilt-based interaction approach and visual feedback through virtual gauge and unobstructed camera feed significantly higher than default method (Fig. 16b). In the post-study interview, most participants praised the minimalistic, uncluttered interface of TiltWalker and wanted to keep using it on their devices. For example, one participant (female, 23 years) commented *"I like the visual feedback and camera view of TiltWalker. It gives better and clearer views, and default is kind of dark and small."* Another (male, 38 years) stated, *"[TiltWalker's] camera view was much better [...I liked that it offered] adjustable velocity."*. Many also praised the method's interaction strategy. One participant (female, 29 years) commented, *"it seems like the TiltWalker allows for easier control to move the robot faster."*. Some also felt that their performance with TiltWalker will increase with practice.

## 10 LIMITATIONS & FINAL REFLECTION

We acknowledge several limitations of the work. First, some participants found the mobile device used in the study to be too slim and slippery. They feared that they would drop the device when performing the tilt gestures, which may have impacted their performance to some extent. We, however, argue this to be a device-specific problem. Upon investigation, we found numerous complaints on various online forums[2] about Samsung Galaxy S6 edge being too slippery. We later tested our system on a OnePlus One smartphone, where no such complaints were raised. Second, TiltWalker required significantly higher mental, physical, and temporal demands than the default method. This is not surprising since new interaction modalities tend to yield higher cognitive loads in early use [26]. Based on the finding that users liked tilt-based interaction significantly more than touch interaction, and prior research on cognitive load theory [76], these demands are likely to reduce over time. Third, the studies investigated controlling the robot through obstacles to reach targets but did explore scenarios after reaching a target, such as approaching a person or engaging in a conversation. We will explore this in a future work. Fourth, upon completion of the final study, we felt that participants needed more practice with the tilt gestures. Some of them requested to redo the tasks with TiltWalker with increased confidence that they would be able to complete the tasks much faster and more accurately. We did not allow that since it would have induced a bias favoring the new method. Finally, the studies did not explore the long-term effects of the tilt-gesture on user comfort and ergonomics. However, we hope to investigate this in a future research.

The contribution of the work is fourfold. First, it presents the lower and the upper-bounds of angles for performing four tilt and two twist gestures without compromising user comfort and the legibility of the display content, which could be used in the development of other tilt-based mobile applications. Second, it proposes and validates a function effective in mapping tilt angles to the velocity of movements, which could essentially be used with other input devices and both physical and virtual displays. Third, the refined virtual gauge presented in the paper could be used in other mobile clients for telepresence robots, reducing clutter and information overload. Finally, it develops and evaluates TiltWalker that substantially improves maneuvering performance compared to a default client, and in theory, can be used with a range of telepresence and other robots. The findings of this work must also encourage developers to consider Web platforms for building cross-platform clients.

## 11 CONCLUSION

We presented a mobile client for controlling a telepresence robot with one hand by performing tilt gestures. It was designed based on a series of lab trials and user studies. First, we justified the use of a Web platform. We then studied how far and fast users can tilt without compromising the comfort and the legibility of the display content and identified a velocity-based function well-suited for control-display mapping in the context of a telepresence robot. Finally, we evaluated the proposed method in a user study. Results revealed that it is significantly faster and more accurate, and participants preferred its interactions and interface significantly more than those of the default method. These findings can assist in designing more effective tilt-based mobile applications, providing more meaningful feedback on robot's current state, and affording more effective control of various out-of-sight robots.

[2]Solution for slippery S6? - Android Forums at AndroidCentral.com, https://forums.androidcentral.com/samsung-galaxy-s6/524806-solution-slippery-s6.html

## 12 FUTURE WORK

In the future, we will evaluate TiltWalker in realistic crowded environments, such as at a shopping mall, since we expect it to be more efficient in these scenarios than the default clients. We will support more complex control commands, such as turning while moving forward and tilting the robot's neck on-the-go. We will also provide graphical feedback through augmented reality to overlay projected trajectories of the robot on the respective video feed.

## REFERENCES


[1] Antti Ainasoja, Said Pertuz, and Joni-Kristian Kämäräinen. 2019. Smartphone Teleoperation for Self-balancing Telepresence Robots:. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS - Science and Technology Publications, Prague, Czech Republic, 561–568. https://doi.org/10.5220/0007406405610568

[2] Mathias Baglioni, Eric Lecolinet, and Yves Guiard. 2011. JerkTilts: Using Accelerometers for Eight-Choice Selection on Mobile Devices. In *Proceedings of the 13th international conference on multimodal interfaces (ICMI '11)*. Association for Computing Machinery, New York, NY, USA, 121–128. https://doi.org/10.1145/2070481.2070503

[3] Greg Baker, Tom Bridgwater, Paul Bremner, and Manuel Giuliani. 2020. Towards an Immersive User Interface for Waypoint Navigation of a Mobile Robot. https://doi.org/10.48550/arXiv.2003.12772 arXiv:2003.12772 [cs].

[4] Anil Ufuk Batmaz, Jens Maiero, Ernst Kruijff, Bernhard E. Riecke, Carman Neustaedter, and Wolfgang Stuerzlinger. 2020. How Automatic Speed Control Based on Distance Affects User Behaviours in Telepresence Robot Navigation Within Dense Conference-Like Environments. *PLOS ONE* 15, 11 (Nov. 2020), e0242078. https://doi.org/10.1371/journal.pone.0242078 Publisher: Public Library of Science.

[5] Amna Batool, Seng W. Loke, Niroshinie Fernando, and Jonathan Kua. 2020. Towards a System for Aged Care Centres based on Multiuser–Multidevice Interactions in IoT Collectives. In *MobiQuitous 2020 - 17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous '20)*. Association for Computing Machinery, New York, NY, USA, 470–475. https://doi.org/10.1145/3448891.3448928

[6] Federica Bazzano, Fabrizio Lamberti, Andrea Sanna, Gianluca Paravati, and Marco Gaspardone. 2017. Comparing Usability of User Interfaces for Robotic Telepresence:. In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS - Science and Technology Publications, Porto, Portugal, 46–54. https://doi.org/10.5220/0006170300460054

[7] Géry Casiez, Daniel Vogel, Ravin Balakrishnan, and Andy Cockburn. 2008. The Impact of Control-Display Gain on User Performance in Pointing Tasks. *Human–Computer Interaction* 23, 3 (Aug. 2008), 215–250. https://doi.org/10.1080/07370020802278163 Publisher: Taylor & Francis _eprint: https://www.tandfonline.com/doi/pdf/10.1080/07370020802278163.

[8] Steven J. Castellucci, I. Scott MacKenzie, Mudit Misra, Laxmi Pandey, and Ahmed Sabbir Arif. 2019. TiltWriter: Design and Evaluation of a No-Touch Tilt-Based Text Entry Method for Handheld Devices. In *Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia (MUM '19)*. ACM, 1–8. https://doi.org/10.1145/3365610.3365629 Article 7.

[9] Amedeo Cesta, Gabriella Cortellessa, and Lorenza Tiberio. 2011. Long-Term Evaluation of a Mobile Remote Presence Robot for the Elderly. *ERCIM News* (Jan. 2011), 20. https://ercim-news.ercim.eu/en84/special/long-term-evaluation-of-a-mobile-remote-presence-robot-for-the-elderly

[10] Youli Chang, Sehi L'Yi, Kyle Koh, and Jinwook Seo. 2015. Understanding Users' Touch Behavior on Large Mobile Touch-Screens and Assisted Targeting by Tilting Gesture. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, Seoul Republic of Korea, 1499–1508. https://doi.org/10.1145/2702123.2702425

[11] J. Chestnutt, P. Michel, K. Nishiwaki, J. Kuffner, and S. Kagami. 2006. An Intelligent Joystick for Biped Control. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* 860–865. https://doi.org/10.1109/ROBOT.2006.1641817 ISSN: 1050-4729.

[12] Sung-Jung Cho, Roderick Murray-Smith, and Yeun-Bae Kim. 2007. Multi-Context Photo Browsing on Mobile Devices Based on Tilt Dynamics. In *Proceedings of the 9th international conference on Human computer interaction with mobile devices and services - MobileHCI '07*. ACM Press, Singapore, 190–197. https://doi.org/10.1145/1377999.1378006

[13] Julien M. Christensen, James W. McBarron, John T. McConville, William R. Pogue, Robert C. Williges, and Wesley E. Woodson. 1995. *Man-Systems Integration Standards NASA-STD-3000.* Technical Report Volume II Revision B. The National Aeronautics and Space Administration (NASA), Washington, D.C., USA. 481 pages.

[14] A. Cockburn, D. Ahlström, and C. Gutwin. 2012. Understanding Performance in Touch Selections: Tap, Drag and Radial Pointing Drag with Finger, Stylus and Mouse. *International Journal of Human-Computer Studies* 70, 3 (March 2012), 218–233. https://doi.org/10.1016/j.ijhcs.2011.11.002

[15] Catalina I. Constantin and I. Scott MacKenzie. 2014. Tilt-Controlled Mobile Games: Velocity-Control Vs. Position-Control. In *2014 IEEE Games Media Entertainment*. 1–7. https://doi.org/10.1109/GEM.2014.7048091

[16] Andrew Crossan and Roderick Murray-Smith. 2004. Variability in Wrist-Tilt Accelerometer Based Gesture Interfaces. In *Mobile Human-Computer Interaction - MobileHCI 2004 (Lecture Notes in Computer Science)*, Stephen Brewster and Mark Dunlop (Eds.). Springer, Berlin, Heidelberg, 144–155. https://doi.org/10.1007/978-3-540-28637-0_13

[17] Raimund Dachselt and Robert Buchholz. 2008. Throw and Tilt – Seamless Interaction across Devices Using Mobile Phone Gestures. *Beherrschbare Systeme - Dank Informatik* Band 1 (2008), 7.

[18] Linda Di Geronimo, Ersan Aras, and Moira C. Norrie. 2015. Tilt-and-Tap: Framework to Support Motion-Based Web Interaction Techniques. In *Engineering the Web in the Big Data Era (Lecture Notes in Computer Science)*, Philipp Cimiano, Flavius Frasincar, Geert-Jan Houben, and Daniel Schwabe (Eds.). Springer International Publishing, Cham, 565–582. https://doi.org/10.1007/978-3-319-19890-3_36

[19] Yanmei Dong, Yunde Jia, Weichao Shen, and Yuwei Wu. 2020. Can You Easily Perceive the Local Environment? A User Interface with One Stitched Live Video for Mobile Robotic Telepresence Systems. *International Journal of Human–Computer Interaction* 36, 8 (May 2020), 736–747. https://doi.org/10.1080/10447318.2019.1685194 Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10447318.2019.1685194.

[20] Mark D. Dunlop, Marc Roper, and Gennaro Imperatore. 2017. Text Entry Tap Accuracy and Exploration of Tilt Controlled Layered Interaction on Smartwatches. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '17)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3098279.3098560

[21] Parisa Eslambolchilar and Roderick Murray-Smith. 2004. Tilt-Based Automatic Zooming and Scaling in Mobile Devices – A State-Space Implementation. In *Mobile Human-Computer Interaction - MobileHCI 2004*, David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Stephen Brewster, and Mark Dunlop (Eds.). Vol. 3160. Springer Berlin Heidelberg, Berlin, Heidelberg, 120–131. https://doi.org/10.1007/978-3-540-28637-0_11 Series Title: Lecture Notes in Computer Science.

[22] Connor Esterwood and Lionel Robert. 2021. Robots and COVID-19: Re-imagining Human–Robot Collaborative Work in Terms of Reducing Risks to Essential Workers. https://doi.org/10.2139/ssrn.3767609

[23] Deborah I. Fels, Judith K. Waalen, Shumin Zhai, and Patrice L. Weiss. 2001. Telepresence Under Exceptional Circumstances: Enriching the Connection to School for Sick Children.. In *Interact*. Tokyo, Japan, 617–624.

[24] Evan Graham. 1996. Virtual Pointing on a Computer Display: Non-Linear Control-Display Mappings. In *Proceedings of the Graphics Interface 1996 Conference, May 22-24, 1996, Toronto, Ontario, Canada*. Canadian Human-Computer Communications Society, 39–46. http://graphicsinterface.org/wp-content/uploads/gi1996-5.pdf

[25] Taejin Ha and Woontack Woo. 2011. ARWand: Phone-Based 3D Object Manipulation in Augmented Reality Environment. In *2011 International Symposium on Ubiquitous Virtual Reality*. 44–47. https://doi.org/10.1109/ISUVR.2011.14

[26] Adrian M Haith and John W Krakauer. 2018. The Multiple Effects of Practice: Skill, Habit and Reduced Cognitive Load. *Current Opinion in Behavioral Sciences* 20 (April 2018), 196–201. https://doi.org/10.1016/j.cobeha.2018.01.015

[27] Beverly L. Harrison, Kenneth P. Fishkin, Anuj Gujar, Carlos Mochon, and Roy Want. 1998. Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '98)*. ACM Press/Addison-Wesley Publishing Co., USA, 17–24. https://doi.org/10.1145/274644.274647

[28] Sandra G. Hart. 2006. Nasa-Task Load Index (NASA-TLX); 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50, 9 (Oct. 2006), 904–908. https://doi.org/10.1177/154193120605000909 Publisher: SAGE Publications Inc.

[29] Sean T. Hayes and Julie A. Adams. 2017. Control-Display Ratio Enhancements for Mobile Interaction. *International Journal of Human-Computer Studies* 99 (March 2017), 57–71. https://doi.org/10.1016/j.ijhcs.2016.10.006

[30] Ken Hinckley, Jeff Pierce, Mike Sinclair, and Eric Horvitz. 2000. Sensing Techniques for Mobile Interaction. In *Proceedings of the 13th annual ACM symposium on User interface software and technology - UIST '00*. ACM Press, San Diego, California, United States, 91–100. https://doi.org/10.1145/354401.354417

[31] Baptiste Isabet, Maribel Pino, Manon Lewis, Samuel Benveniste, and Anne-Sophie Rigaud. 2021. Social Telepresence Robots: A Narrative Review of Experiments Involving Older Adults before and during the COVID-19 Pandemic. *International Journal of Environmental Research and Public Health* 18, 7 (Jan. 2021), 3597. https://doi.org/10.3390/ijerph18073597 Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.

[32] Heejin Kim, Seungjae Oh, Sung H. Han, and Min K. Chung. 2019. Motion–Display Gain: A New Control–Display Mapping Reflecting Natural Human Pointing Gesture to Enhance Interaction with Large Displays at a Distance. *International Journal of Human–Computer Interaction* 35, 2 (Jan. 2019), 180–195. https://doi.org/10.1080/10447318.2018.1447422 Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10447318.2018.1447422.

[33] Andrey Kiselev, Annica Kristoffersson, Francisco Melendez, Cipriano Galindo, Amy Loutfi, Javier Gonzalez-Jimenez, and Silvia Coradeschi. 2015. Evaluation of using semi-autonomy features in mobile robotic telepresence systems. In

*2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*. IEEE, 147–152.

[34] Atsunori Kogawa, Moeko Onda, Yoshihiro Kai, and Department of Mechanical Engineering, Tokai University 4-1-1 Kitakaname, Hiratsuka, Kanagawa 259-1292, Japan. 2021. Development of a Remote-Controlled Drone System by Using Only Eye Movements: Design of a Control Screen Considering Operability and Microsaccades. *Journal of Robotics and Mechatronics* 33, 2 (April 2021), 301–312. https://doi.org/10.20965/jrm.2021.p0301

[35] Sven Kratz and Fred Rabelo Ferriera. 2016. Immersed Remotely: Evaluating the Use of Head Mounted Devices for Remote Collaboration in Robotic Telepresence. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 638–645. https://doi.org/10.1109/ROMAN.2016.7745185 ISSN: 1944-9437.

[36] Annica Kristoffersson, Silvia Coradeschi, and Amy Loutfi. 2013. A Review of Mobile Robotic Telepresence. https://doi.org/10.1155/2013/902316 ISSN: 1687-5893 Pages: e902316 Publisher: Hindawi Volume: 2013.

[37] Daniel Labonte, Francois Michaud, Patrick Boissy, Helene Corriveau, Richard Cloutier, and Marc Andre Roux. 2006. A Pilot Study on Teleoperated Mobile Robots in Home Environments. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 4466–4471. https://doi.org/10.1109/IROS.2006.282082 ISSN: 2153-0866.

[38] Jens Laigaard, Trine Ungermann Fredskild, and Grzegorz Lukasz Fojecki. 2022. Telepresence Robots at the Urology and Emergency Department: A Pilot Study Assessing Patients' and Healthcare Workers' Satisfaction. *International Journal of Telemedicine and Applications* 2022 (March 2022), e8787882. https://doi.org/10.1155/2022/8787882 Publisher: Hindawi.

[39] Min Kyung Lee and Leila Takayama. 2011. "Now, I Have a Body": Uses and Social Norms for Mobile Remote Presence in the Workplace. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. Association for Computing Machinery, New York, NY, USA, 33–42. https://doi.org/10.1145/1978942.1978950

[40] Chuanyi Liu, Chang Liu, Hao Mao, and Wei Su. 2019. Tilt-Scrolling: A Comparative Study of Scrolling Techniques for Mobile Devices. In *Intelligent Computing Methodologies (Lecture Notes in Computer Science)*, De-Shuang Huang, Zhi-Kai Huang, and Abir Hussain (Eds.). Springer International Publishing, Cham, 189–200. https://doi.org/10.1007/978-3-030-26766-7_18

[41] Jiayang Liu, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan. 2009. uWave: Accelerometer-Based Personalized Gesture Recognition and Its Applications. *Pervasive and Mobile Computing* 5, 6 (Dec. 2009), 657–675. https://doi.org/10.1016/j.pmcj.2009.07.007

[42] Xuesen Liu. 2021. *Investigation of Tilt Gesture Interaction to Manipulate Text Property in Mobile Device*. Thesis. University of Michigan-Dearborn, Dearborn, Michigan, USA. https://doi.org/10.7302/2329 Accepted: 2021-08-25T19:30:50Z.

[43] Adrien Lociciro, Antoine Guillon, and Laetitia Bodet-Contentin. 2021. A Telepresence Robot in the Room of a COVID-19 Patient Can Provide Virtual Family Presence. *Canadian Journal of Anesthesia/Journal canadien d'anesthésie* 68, 11 (Nov. 2021), 1705–1706. https://doi.org/10.1007/s12630-021-02039-6

[44] Mateus M. Luna, Thyago P. Carvalho, Fabrizzio Alphonsus A. M. N. Soares, Hugo A. D. Nascimento, and Ronaldo M. Costa. 2017. Wrist Player: A Smartwatch Gesture Controller for Smart TVs. In *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 2. 336–341. https://doi.org/10.1109/COMPSAC.2017.266 ISSN: 0730-3157.

[45] Maria Luce Lupetti, Luca Giuliano, and Claudio Germak. 2016. Virgil Robot at Racconigi Castle: A Design Challenge. In *Proceedings of the Seventh International Workshop on Human-Computer Interaction, Tourism and Cultural Heritage (HCITOCH '16)*. Blue Herons Editions, Turin, Italy, 1–20. https://doi.org/10.978.8896471/586

[46] Nasiru Adamu Marafa, Walter de Britto Vidal, and Carlos Humberto Llanos. 2020. A Design Methodology and Development of a Mobile Telepresence Robot for Paraplegics. *Product Management & Development* 18, 2 (2020), 181–195. https://doi.org/10.4322/pmd.2020.030

[47] Giovanni Mosiello, Andrey Kiselev, and Amy Loutfi. 2013. Using Augmented Reality to Improve Usability of the User Interface for Driving a Telepresence Robot. *Paladyn, Journal of Behavioral Robotics* 4, 3 (Dec. 2013), 174–181. https://doi.org/10.2478/pjbr-2013-0018 Publisher: De Gruyter Open Access.

[48] Maral Muratbekova-Touron and Emmanuelle Leon. 2021. "Is There Anybody Out There?" Using a Telepresence Robot to Engage in Face Time at the Office. *Information Technology & People* ahead-of-print, ahead-of-print (Jan. 2021). https://doi.org/10.1108/ITP-01-2021-0080

[49] Robin R. Murphy, Vignesh Babu Manjunath Gandudi, and Justin Adams. 2020. Applications of Robots for COVID-19 Response. https://doi.org/10.48550/arXiv.2008.06976 arXiv:2008.06976 [cs].

[50] Jani Mäntyjärvi, Sanna Kallio, Panu Korpipää, Juha Kela, and Johan Plomp. 2005. Gesture Interaction for Small Handheld Devices to Support Multimedia Applications. *Journal of Mobile Multimedia* (April 2005), 092–112. https://journals.riverpublishers.com/index.php/JMM/index

[51] NASA. 1986. *NASA Task Load Index (TLX) V 1.0: Paper and Pencil Package*. Technical Report. Human Performance Research Group, NASA Ames Research Center, Moffett Field, CA, USA.

[52] Aditya Nawab, Keshav Chintamani, Darin Ellis, Gregory Auner, and Abhilash Pandya. 2007. Joystick Mapped Augmented Reality Cues for End-Effector Controlled Tele-Operated Robots. In *2007 IEEE Virtual Reality Conference*.

263–266. https://doi.org/10.1109/VR.2007.352496 ISSN: 2375-5334.

[53] Carman Neustaedter, Samarth Singhal, Rui Pan, Yasamin Heshmat, Azadeh Forghani, and John Tang. 2018. From Being There to Watching: Shared and Dedicated Telepresence Robot Usage at Academic Conferences. *ACM Transactions on Computer-Human Interaction* 25, 6 (Dec. 2018), 1–39. https://doi.org/10.1145/3243213

[54] Tao Ni, Doug A. Bowman, Chris North, and Ryan P. McMahan. 2011. Design and Evaluation of Freehand Menu Selection Interfaces Using Tilt and Pinch Gestures. *International Journal of Human-Computer Studies* 69, 9 (Aug. 2011), 551–562. https://doi.org/10.1016/j.ijhcs.2011.05.001

[55] Ian Oakley and Junseok Park. 2007. A Motion-Based Marking Menu System. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems (CHI EA '07)*. Association for Computing Machinery, New York, NY, USA, 2597–2602. https://doi.org/10.1145/1240866.1241048

[56] Ian Oakley, Jussi Ängeslevä, Stephen Hughes, and Sile O'Modhrain. 2004. Tilt and Feel: Scrolling with Vibrotactile Display. In *Proceedings of EuroHaptics 2004 (EuroHaptics '04)*. Munich, Germany.

[57] Grace Pan, Thea Weiss, Seyed Amirhosein Mohaddesi, John Szura, and Jeffrey Krichmar. 2022. The Benefits of Autonomous Navigation in Telepresence Robots and its Effect on Cognitive Load. https://doi.org/10.31234/osf.io/28tys

[58] Richard Pinsenschaum and Flaithri Neff. 2016. Evaluating Gesture Characteristics When Using a Bluetooth Handheld Music Controller. In *Proceedings of the Audio Mostly 2016 (AM '16)*. Association for Computing Machinery, New York, NY, USA, 209–214. https://doi.org/10.1145/2986416.2986443

[59] Y. Rabhi, M. Mrabet, and F. Fnaiech. 2015. Optimized Joystick Control Interface for Electric Powered Wheelchairs. In *2015 16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*. 201–206. https://doi.org/10.1109/STA.2015.7505092

[60] Irene Rae, Bilge Mutlu, and Leila Takayama. 2014. Bodies in Motion: Mobility, Presence, and Task Awareness in Telepresence. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. Association for Computing Machinery, New York, NY, USA, 2153–2162. https://doi.org/10.1145/2556288.2557047

[61] Mahfuz Rahman, Sean Gustafson, Pourang Irani, and Sriram Subramanian. 2009. Tilt Techniques: Investigating the Dexterity of Wrist-Based Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. Association for Computing Machinery, New York, NY, USA, 1943–1952. https://doi.org/10.1145/1518701.1518997

[62] Arsénio Reis, Rui Xavier, Isabel Barroso, Maria João Monteiro, Hugo Paredes, and João Barroso. 2018. The Usage of Telepresence Robots to Support the Elderly. In *2018 2nd International Conference on Technology and Innovation in Sports, Health and Wellbeing (TISHW)*. 1–6. https://doi.org/10.1109/TISHW.2018.8559549

[63] Jun Rekimoto. 1996. Tilting Operations for Small Screen Interfaces. In *Proceedings of the 9th annual ACM symposium on User interface software and technology (UIST '96)*. Association for Computing Machinery, New York, NY, USA, 167–168. https://doi.org/10.1145/237091.237115

[64] Yuan Ren and Ahmed Sabbir Arif. 2021. Stepper, Swipe, Tilt, Force: Comparative Evaluation of Four Number Pickers for Smartwatches. *Proceedings of the ACM on Human-Computer Interaction* 5, ISS (Nov. 2021), 500:1–500:21. https://doi.org/10.1145/3488545

[65] Francisco Javier Rodríguez Lera, Juan Felipe García Sierra, Camino Fernández Llamas, and Vicente Matellán Olivera. 2011. Augmented reality to improve teleoperation of mobile robots. (Sept. 2011). Accepted: 2013-01-23T18:09:53Z.

[66] I. Scott Mackenzie and Stan Riddersma. 1994. Effects of Output Display and Control—Display Gain on Human Performance in Interactive Systems. *Behaviour & Information Technology* 13, 5 (Sept. 1994), 328–337. https://doi.org/10.1080/01449299408914613

[67] Gary Smith. 2011. A Boy and His Bot. *Sports Illustrated Vault* (Aug. 2011), 54–62. https://vault.si.com/vault/2011/08/01/a-boy-and-his-bot

[68] Leila Takayama. 2015. Telepresence and Apparent Agency in Human–Robot Interaction. In *The Handbook of the Psychology of Communication Technology*. John Wiley & Sons, Ltd, 160–175. https://doi.org/10.1002/9781118426456.ch7 Section: 7 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118426456.ch7.

[69] Robert J. Teather and I. Scott MacKenzie. 2020. Position vs. Velocity Control for Tilt-Based Interaction. In *Graphics Interface 2014* (1 ed.), Paul G. Kry and Andrea Bunt (Eds.). A K Peters/CRC Press, 51–58. https://doi.org/10.1201/9781003059325-7

[70] Keng Peng Tee, Rui Yan, Yuanwei Chua, Zhiyong Huang, and Somchaya Liemhetcharat. 2014. Gesture-Based Attention Direction for a Telepresence Robot: Design and Experimental Study. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 4090–4095. https://doi.org/10.1109/IROS.2014.6943138 ISSN: 2153-0866.

[71] Luca Tonin, Tom Carlson, Robert Leeb, and José del R. Millán. 2011. Brain-Controlled Telepresence Robot by Motor-Disabled People. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 4227–4230. https://doi.org/10.1109/IEMBS.2011.6091049 ISSN: 1558-4615.

[72] Bradley van Tonder and Janet Wesson. 2010. Is Tilt Interaction Better Than Keypad Interaction for Mobile Map-Based Applications?. In *Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists (SAICSIT '10)*. Association for Computing Machinery, New York, NY, USA, 322–331.

https://doi.org/10.1145/1899503.1899539

[73] Mingzhong Wang, Chongdan Pan, and Pradeep Kumar Ray. 2021. Technology Entrepreneurship in Developing Countries: Role of Telepresence Robots in Healthcare. *IEEE Engineering Management Review* 49, 1 (2021), 20–26. https://doi.org/10.1109/EMR.2021.3053258 Conference Name: IEEE Engineering Management Review.

[74] Mette Weibel, Martin Kaj Fridh Nielsen, Martha Krogh Topperzer, Nanna Maria Hammer, Sarah Wagn Møller, Kjeld Schmiegelow, and Hanne Bækgaard Larsen. 2020. Back to School with Telepresence Robot Technology: A Qualitative Pilot Study About How Telepresence Robots Help School-Aged Children and Adolescents with Cancer to Remain Socially and Academically Connected with Their School Classes During Treatment. *Nursing Open* 7, 4 (2020), 988–997. https://doi.org/10.1002/nop2.471 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nop2.471.

[75] J. S. Weiner. 1971. Fitting the Task to the Man—An Ergonomic Approach. *British Journal of Industrial Medicine* 28, 2 (April 1971), 210. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1009274/

[76] Christopher D. Wickens, Justin G. Hollands, Simon Banbury, and R. Parasuraman. 2013. *Engineering Psychology and Human Performance* (fourth edition ed.). Pearson, Boston.

[77] Daniel Wigdor and Ravin Balakrishnan. 2003. TiltText: Using Tilt for Text Input to Mobile Phones. In *Proceedings of the 16th annual ACM symposium on User interface software and technology (UIST '03)*. Association for Computing Machinery, New York, NY, USA, 81–90. https://doi.org/10.1145/964696.964705

[78] Nguyen Dao Xuan Hai, Luong Huu Thanh Nam, and Nguyen Truong Thinh. 2019. Remote Healthcare for the Elderly, Patients by Tele-Presence Robot. In *2019 International Conference on System Science and Engineering (ICSSE)*. 506–510. https://doi.org/10.1109/ICSSE.2019.8823334 ISSN: 2325-0925.

[79] YangLillian, JonesBrennan, NeustaedterCarman, and SinghalSamarth. 2018. Shopping Over Distance through a Telepresence Robot. *Proceedings of the ACM on Human-Computer Interaction* (Nov. 2018). https://doi.org/10.1145/3274460 Publisher: ACM PUB27 New York, NY, USA.

[80] Ludek Zalud. 2006. ARGOS - System for Heterogeneous Mobile Robot Teleoperation. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 211–216. https://doi.org/10.1109/IROS.2006.282495 ISSN: 2153-0866.

[81] Guangtao Zhang and John Paulin Hansen. 2022. Telepresence Robots for People with Special Needs: A Systematic Review. *International Journal of Human–Computer Interaction* (Feb. 2022), 1–17. https://doi.org/10.1080/10447318.2021.2009673