操偶師：提供直覺性手勢與上半身姿態以增進虛擬人型角色操作之體驗

洪靖雯
國立台灣大學
台北, 台灣

張睿哲
國立台灣大學
台北, 台灣

陳宏昇
國立台灣大學
台北, 台灣

梁中瀚
國立台灣大學
台北, 台灣

詹力韋
國立陽明交通大學
新竹, 台灣

陳炳宇
國立台灣大學
台北, 台灣

ABSTRACT

在過去遊戲體驗中，以身體方式做為操作的遊玩方式，提供給玩家能直觀地即時操作虛擬角色，獲得沈浸的遊戲體驗，然而這類的操作方式需要現實中大量的遊玩空間及需花費玩家多餘的力氣來進行遊玩體驗；相比之下，以手為操作虛擬角色的方式，提供玩家更精準且低疲勞的遊玩體驗，讓玩家可以在小範圍空間中能很好操作這些虛擬角色，但該方法相比身體操作方式缺少許多身體感測機制，而導致較低的遊玩沈浸感。本篇論文旨在觀察這兩種操作方式間的優缺點與其適合使用的情境，並結合兩者方法的優勢處來作為一種新的遊玩操作方式，提供玩家更直覺且更舒適的遊玩體驗。我們執行一個形成性研究來瞭解玩家會在何時、何地去使用者兩種操作方式，並在該研究結果中得知使用者會偏好使用上半身姿態與手勢在不同遊戲情境下進行操作。基於該形成性研究結果，我們進一步執行使用者偏好設計研究，邀請使用者們來針對現有商業化熱門遊戲中常見的17個遊戲角色動作，進行手勢與姿態的偏好設計，並選擇共識度最高的手勢與姿態最為各個動作的代表操作方式。該論文使用商業化技術框架MediaPipe去偵測使用者姿勢與姿態骨架，並用自己設計的機器學習演算法進行姿態與姿勢辨識。最後，我們展示三個遊戲情境來顯示出操偶師系統在未來的可使用情境。

CCS CONCEPTS

• **Human-centered computing → Gestural input.**

KEYWORDS

Body Posture, Hand Gesture, Camera system, User-Defined Gesture, Video Game, Input Techniques

ACM Reference Format:
洪靖雯, 張睿哲, 陳宏昇, 梁中瀚, 詹力韋, and 陳炳宇. 2021. 操偶師：提供直覺性手勢與上半身姿態以增進虛擬人型角色操作之體驗. In *TAICHI '22: The 8th Annual Conference of Taiwan Association of Computer-Human Interaction, October 15–16, 2021, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/1122445.1122456

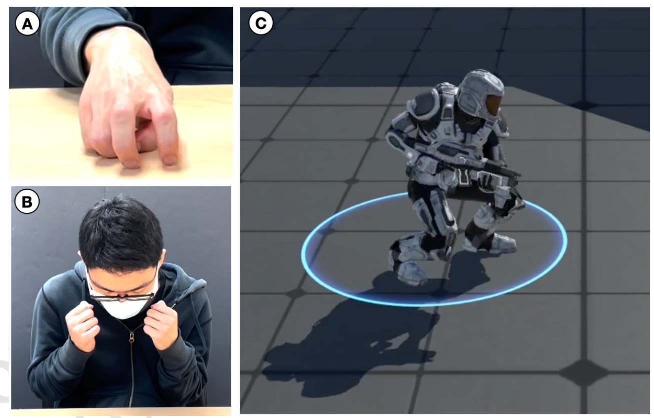

Figure 1: Puppeteer introduces the concept of combining hand gestures and upper-body postures to control avatars actions. Here is an example of the *crouch* action: (a) hand gesture (b) upper-body posture, and (c) avatar animation.

1 INTRODUCTION

Gamepad-controlled avatars are often used as the main avatar manipulation for gaming. However, the limited capabilities of gamed-controlled manipulation lack intuitive control, which affects players' presence, enjoyment, and agency of avatar control in video games [29]. Besides, the manipulation requires players to hold input devices during gaming, which restricts the freedom of hand movements to affect the game experience. Body-controlled avatars provide a more intuitive and free-hand manipulation that allows players to directly control their avatars in the virtual world through real-time body-to-body motion mapping, such as Kinect[1], Vicon[2], Optitrack[3], etc. However, this type of manipulation is not appropriate to use in scenarios where players are in a narrow space or want to sit to play games because the manipulation needs more physical effort and interaction space [17, 19]. On the opposite, finger-controlled avatars provide dexterous and direct manipulation within a close-range space where players only use their hands to control avatar movement, including digital puppetry techniques [4, 12, 17, 23, 38] and iconic gestures [11, 27, 30, 35]. Although finger-controlled systems provide fewer sensory cues than body-controlled systems,

[1]https://en.wikipedia.org/wiki/Kinect
[2]https://www.vicon.com/
[3]https://optitrack.com/

they present less body fatigue and a more convenient method to explore the virtual environment [14]. The two intuitive manipulations have their advantages and limitations, which motivates us to consider whether they have appropriate scenarios to represent. Could we combine the two techniques with their benefits as a new input technique?

Although previous works demonstrated either body-controlled or finger-controlled avatars [1, 4, 12, 30], none of them discussed tradeoffs and user preferences for body-controlled and hand-controlled manipulation that would significantly affect the perception of game experience. In response, to understand and compete the advantages of the two manipulations, we conducted a formative study to know why, when, and how to use body postures or hand gestures for avatar manipulation in 20 games selected from three top-sell categories on STEAM[4]. Through the survey of the 20 games, we decided to focus on representing human avatars' motions, and we further interviewed participants about their preferences for hand-controlled and body-controlled avatars. According to the result of the formative study, players use their bodies to represent avatar actions when the actions are easy to be mimicked by bodies. However, when players want to sit during gaming to reduce body fatigue, or the avatar actions are unrealistic and hard to be represented directly by bodies, they tend to use hands to control their avatars. We discovered that hand gestures provide an alternative to lower-body movement when players do not want to move exaggeratedly.

Based on the above results, we proposed Puppeteer, a novel input system that leverages hand gestures and upper-body postures as an intuitive manipulation to control avatar actions, which is shown in Figure 1. Puppeteer consists of a multi-camera system that can recognize the selected 17 upper-body postures and 17 hand gestures using our self-trained machine learning model, which achieves an average of 90% accuracy for upper-body postures and 91% for hand gestures detection. We performed a formative study investigating users' preference between hand-based and body-based input. We then examined a gesture elicitation study to get gestures/postures users defined to manipulate avatar actions. Based on the defined gestures/postures, we collected data to create two datasets, implemented a prototype system for gesture recognition, and developed three game applications to demonstrate the Puppeteer system. Finally, we also discussed future applications of Puppeteer that goes beyond games and current limitations.

In summary, this paper contributes to:

- A formative study to understand the preferences, timings and reasons to use upper-body postures and hand gestures to control human avatar.
- A gesture elicitation study to understand the best-suited upper-body postures and hand gestures of actions in popular game genres.
- Puppeteer, a multi-camera system that can recognize the elicited postures and gestures.
- Three game applications to demonstrate the Puppeteer system.

[4]https://store.steampowered.com/

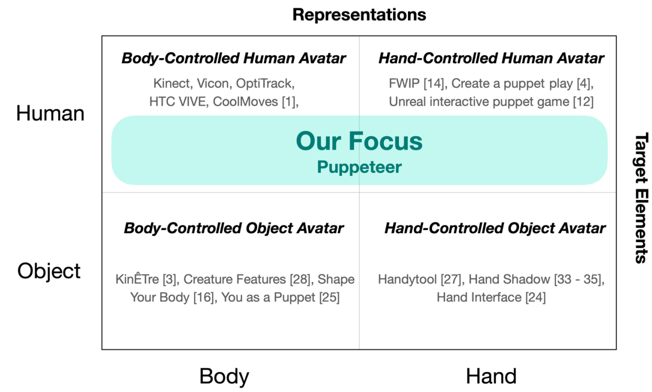

Figure 2: Category of avatar manipulations: The horizontal axis shows what represents an avatar, such as a body or hand. The vertical axis shows the type of avatar expressed, such as an object avatar or human avatar. Our focus is on manipulating human avatars using hands and upper bodies.

2 RELATED WORK

Previous works proposed many approaches for avatar motion manipulations. We divided these approaches into three categories and discussed as below: *Object-Controlled Avatar Manipulation*, *Body-Controlled Avatar Manipulation*, and *Hand-Controlled Avatar Manipulation.*

2.1 Object-Controlled Avatar Manipulation

Input devices are popular in gaming for avatar action control, from single-button controllers to multi-button keyboards, mouses, and joysticks. However, the mapping between input devices and virtual avatars is not natural and intuitive, which affects the perception of presence, enjoyment, and embodiment in video games [29, 31]. Some works discussed more precise sensing tools or tangible user interfaces (TUIs) for accurately manipulating avatar [5, 7, 10, 15, 39, 42]. In addition, other works leveraged everyday physical objects (e.g., mobile phones, virtual reality (VR) controllers, toys) for characters' 3D animation and moving trajectory in the virtual world [6, 9, 41] . Nonetheless, these techniques required built-in sensors or needed users to hold other devices, which decreased freedom of hands movement and affected users' perception during gaming. In addition, the out-of-body mapping also decreased the embodiment of avatars.

2.2 Body-Controlled Avatar Manipulation

The other method to control an avatar's behavior by user input is the whole-body tracking system, which provides an intuitive way for 1-to-1 skeleton mapping between users and avatars. Many commercial systems for motion capture, such as Kinect, detects body skeleton and apply them to avatars' skeleton model; Vicon and Optitrack use multiple cameras and markers to track users' movement. For VR devices, HTC VIVE and Oculus Quest use the head-mounted display (HMD) and controllers to track users' body motion in VR. However, such systems require players to entirely behave with avatar motions, making them need to spend more effort manipulating avatars, especially for exaggerated actions that often appear in video games.

Recently, researchers started to discuss how to use body input techniques to manipulate avatar actions. Some works applied users' skeleton, which Kinect detected on a virtual object model for users to make animations [3, 16, 26]. CoolMoves [1] proposed a motion accentuation method that used a motion capture database to match and blend a user's input from limited input cues by current VR devices into a whole-body avatar motion. BodyAvatar [44] created a Kinect-based system that allowed users to leverage body postures to create 3D models as their virtual embodiment and control their models. You as a Puppet [25] tracked a performer's body and facial movement through Kinect and HMD to control a puppet remotely and got audio feedback from the puppet's vision for more immersive telepresence in puppet manipulation. Creature Features [28] focused on non-human character motions from human body input. Imaginary devices [30] imitated a set of game input devices that allowed users to choose suitable devices for different game scenarios fast.

2.3 Hand-Controlled Avatar Manipulation

On the other side, previous works explored that users manipulate virtual avatars with hand-based input techniques. We further discussed pieces of literature with object avatars and human avatars manipulated by hand gestures.

2.3.1 Hand-Controlled Object Avatar Manipulation. Handytool [27] used iconic hand gestures to control object avatars for becoming virtual tools itself that improved the task performance in VR. Hand Shadow [18, 33–35] reported a method with iconic hand gestures for controlling animal avatars, used for 3D model animation or telecommunication scenarios, respectively. Imaginary devices [30] also proposed a hand gesture to mimic a gun that makes a hand become the gun object avatar. Recently, Hand Interfaces [24] proposed a new interaction technique in AR/VR focusing on virtual object imitation, such as having a thumb-up hand gesture to imitate a joystick.

2.3.2 Hand-Controlled Human Avatar Manipulation. One of the common hand-based input techniques for human avatar movement is finger walking. Finger walking in place (FWIP) [14] was the earliest work to propose the technique that allowed users to slide on a multi-touch input device for locomotion in a virtual world; Fingerwalking [19] as a similar work generated full-body animation through finger walking. Based on the FWIP technique, Ujitoko et al. [37] provided tactile feedback while users performed finger-walking to generate an illusionary feeling of the sense of body ownership of the avatars' invisible legs. Miniature Haptics [38] also provided haptic feedback on fingers to generate whole-body scale haptic illusion as a more practical method for haptic feeling in VR experience.

Another hand-embodied human avatar method proposes a skeleton mapping between a physical hand and a virtual avatar. Luo et al. [20] and Okada et al. [22] used sensing gloves to track users' finger motions to animate virtual avatar actions. Huang et al. [12] proposed a hand-to-body skeleton structure to bind between a hand and a virtual avatar body for animation; Cheng et al. [4] also demonstrated a similar skeleton mapping system as Huang et al. [12] to animate a virtual humanoid avatar.

Some works designed specific hand gestures for avatar motion manipulation. Tung et al. [36] leveraged user-defined gestures among hands, rings, and legs for avatar control in smart glasses scenarios; Zhang et al. [43] also discussed hand gestures for avatar movement in VR. Mani-Pull-Action [17] and Character Motion Control Interface [23] provided an interactive motion control similar to marionettes manipulation that made two hands control avatar actions for more accurate avatar animations. However, these methods decrease avatar embodiment due to their out-of-body mapping to hands, mentioned by Miniature Haptics [38].

Based on our knowledge, only PuppetX [8] proposed a system that allowed users to use full-body gestures to manipulate avatars with self-construct modular components, but it did not discuss tradeoffs and user preferences between hand gestures and body postures. In this paper, we focus on the *body-controlled human avatar manipulation* and the *hand-controlled human avatar manipulation*, which showed on Figure 2. We proposed Puppeteer, which combined defined hand gestures and body postures and allowed users decided when to use hands or bodies to control their avatars based on their preferences. Puppeteer provides a more intuitive game input method to increase avatar embodiment during gaming.

3 STUDY I: FORMATIVE STUDY

To understand the preferences, timings, and reasons that users want to use body postures and hand gestures to control human avatars, we conducted a formative study. To find common actions that often appear in video games on the market currently, we selected 20 video games from the top-seller category on the most prominent digital distribution platform STEAM[5], which is based on PuPoP [32]. First, we performed a survey to search the "video game" keyword on steam and picked the top 20 games. We discovered that 18 games used human avatars to explore virtual worlds in these games. The other two games belonged to the digital collectible card game (DCCG) category, in which players used interfaces to pick cards without controlling avatars. Besides, 16 games were in third-person view, where players view their avatars as onlookers, and four games were first-person view, where players controlled their avatars directly. Based on the survey result, we decided to focus on representing human avatars' motions to apply Puppeteer to more aspects of games. Notably, the human avatar described in this paper includes human and humanoid characters, which can be animated by a human skeleton model. In addition, it was more precise for users to see the whole-body actions of the avatars and more accessible for users to design gestures of the actions when games are third-person view.

Therefore, we reselected the top 20 video games more searched by the "third-person video games" keyword on steam. These games include action games, role-playing games (RPG), shooter games, and adventure games. Two of our authors watched each game's trailer and gameplay video and labeled all character actions. Then We made 20 demonstration videos that contained these labeled actions. Each demonstration video was less than or near one minute.

[5]https://store.steampowered.com/search/?filter=topsellers

A total of 85 actions were labeled, of which 19 actions were unique. Five participants (aged 22 - 26, 3 females) were invited to do the study. All participants were familiar with playing video games. Each participant was asked to think about when they preferred to use body postures or hand gestures to control their avatars, followed on ARAnimator [41]. We provided three options (upper-body, lower-body, hand approach) to invoke participants to think aloud. They needed to consider factors that might affect performing gestures/postures, like fatigue, ease-to-perform, comfort, goodness, and intuition. The upper-body and lower-body approaches separately include body postures above and below the waist. For the hand approach, we explained to participants that it includes the puppetry method, which means seeing hands as a small avatar and controlling the avatar through fingers and palms, and iconic gestures, which are metaphors to represent virtual meanings for avatar motion controls. In addition, we also let participants consider the scenario that fast switches actions in gaming because this scenario is common in many games, especially in action games. The participants needed to rank three options for each avatar action and explain the reason for the preferences. To provide enough time for participants, we sent demonstration video links to rewatch online and gave them one day to think about their gesture/posture preference on avatar actions. Then participants returned the next day and explained their preference for labeled actions. The participants watched 20 videos in order, and we interviewed them with considerate factors and discussed their ranking lists. The whole progress of the interview was video recorded. It took about 1 hour to interview each participant.

Based on the interview result, most participants expressed that they used their bodies to control avatar actions if the actions can be directly represented by bodies, especially in games focusing on storytelling. *P2, P3, P4* also mentioned that using body postures increased immersion in gaming. However, when the participants found that the avatar's actions were too exaggerated (e.g., running in the mountain, rolling over quickly) or unrealistic (e.g., flying in the sky, stopping in the air) to mimic, or the avatar switched too fast with multiple actions, they preferred to use their hands to manipulate avatar motions. All participants agreed that the hand approach was easy to perform in gaming with less fatigue, which was appropriate to play in gaming for a long time. Additionally, hand gestures propose an alternative to lower-body motions when players prefer to sit during gaming. Most participants except *P2* expressed that lower-body postures represented limited avatar actions. They thought lower-body postures were suitable for representing lower-body actions, which hand gestures can also express. Besides, *P4* preferred no leg movement if she sits to play games, and she wants the least effort for controlling avatars. She also recommended that other methods can replace the lower-body approach. Based on the interview result, we decided to remove the lower-body approach and focus on the hand and the upper-body methods.

4 STUDY II: USER-DEFINED GESTURES/POSTURES STUDY

To explore how participants define gestures/postures to represent avatar actions, we performed a user-defined gestures/postures study, as mainly followed on ARAnimator [41].

4.1 Apparatus and Procedure

Game Scenarios	Avatar Actions
Action-Adventure Game	Climb, Row a Boat, Punch, Drive, Fly, Open a Door, Crawl
RPG Game	Walk, Run, Jump, Ride, Roll, Defend, Swim
Shooter Game	Shoot, Use a Weapon, Crouch

Table 1: The action list of the three game scenarios shown in the demonstration videos on Study II.

Based on the result of Study I, we implemented three game scenarios that included the most popular game genres on the statistical result of the formative study. Three game scenarios are *Action-Adventure Game*, *RPG Game*, and *Shooter Game*. We bought the scenarios' game scenes from Unity Asset Store[6], and implemented some character models and action animations using Adobe Mixamo[7]. These scenarios included avatar actions labeled from the formative study, as shown in Table 1. We filtered two uncommon labeled actions that belonged to multiple keypresses in keyboard input (e.g., shooting with walking, crouching with walking) and finally selected 17 unique actions in this study. We recruited new 12 participants (aged 21 - 26, 4 females) to conduct this study. In a similar procedure to the formative study, the participants were asked to watch the demonstrations videos containing actions clipped from the three game scenarios and designed hand gestures and upper-body postures that best represent each avatar action. To avoid legacy bias which participants generated because participants were familiar with traditional game input systems in video games, we followed production gesture methods [2, 21] and encouraged participants to define three hand gestures and three body postures. Then, the participants selected one gesture/posture that they most preferred for each action. Each gesture/posture was performed in 5 seconds. The whole defining process was video-recorded for further result analysis and system implementation. To help the participants design a unique gesture/posture for each action with enough time, we provided one day for the participants to go through all demonstration videos and think aloud about different gestures/postures before performing the study. The next day, the participants returned to perform their designed gestures/postures. Besides, the participants needed to consider their defined gestures/postures in terms of goodness, ease-of-perform, intuition, and comfort when they selected their preferred gestures/postures. The whole study took about 2.5 hours for each participant.

4.2 Result and Discussion

We collected a total of 1024 action gestures/postures, which includes 612 (= 17 (actions) × 3 (types of gestures/postures) × 12 (participants)) hand gestures and 612 upper-body postures, and we isolated totally 408 preferred gestures/postures. Finally, we chose one representative hand gesture and one upper-body posture for each action with the largest number selected by the participants,

[6]https://assetstore.unity.com/
[7]https://www.mixamo.com/

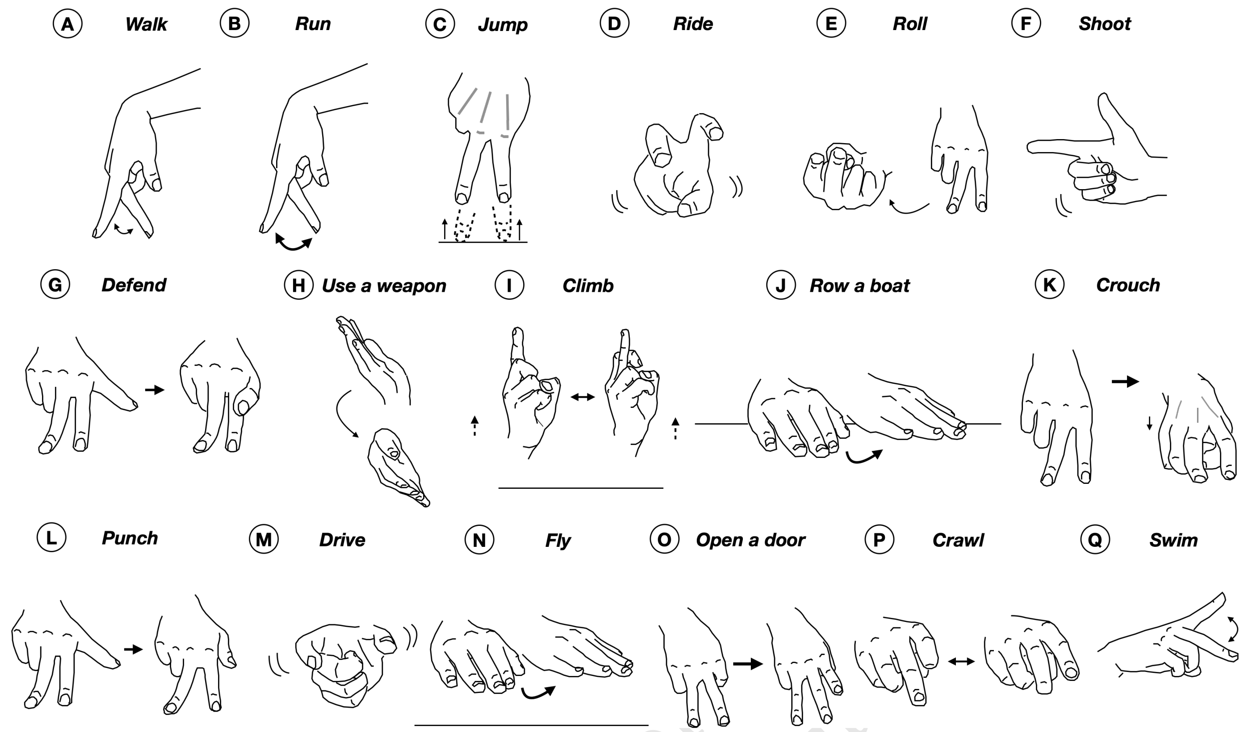

Figure 3: The user-defined hand gestures from Study II.

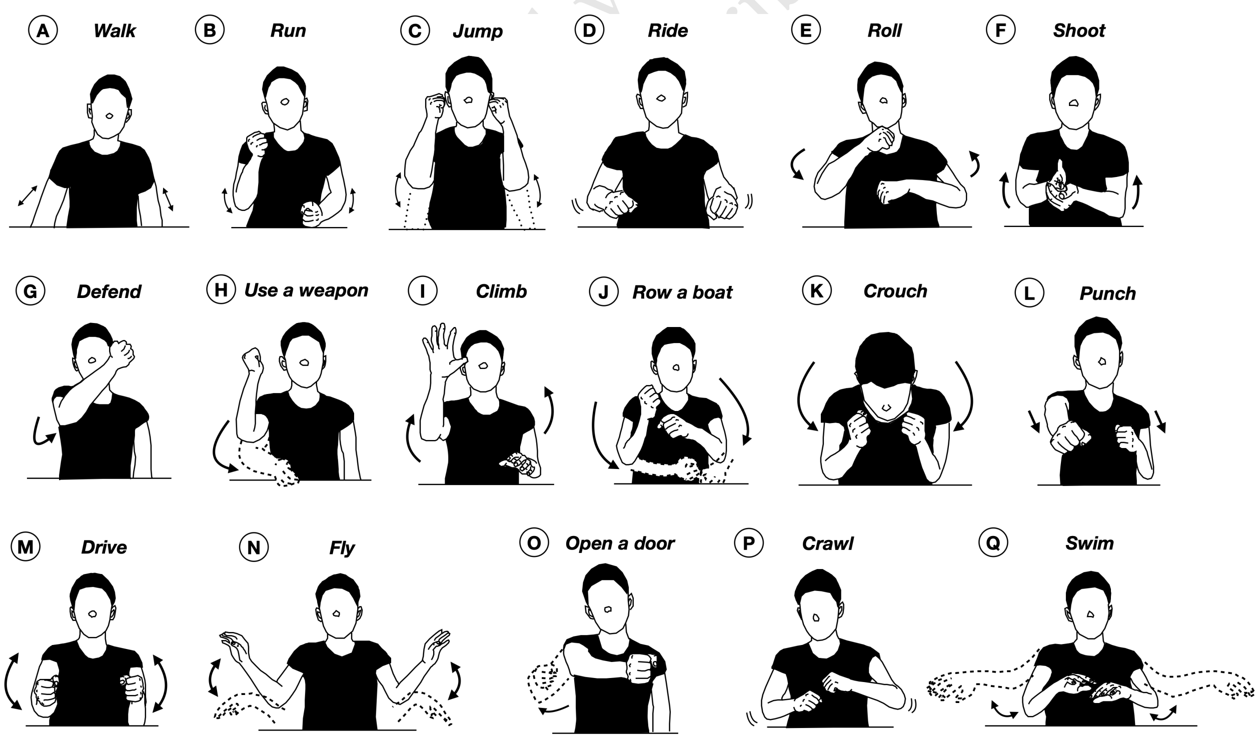

Figure 4: The user-defined upper-body postures from Study II.

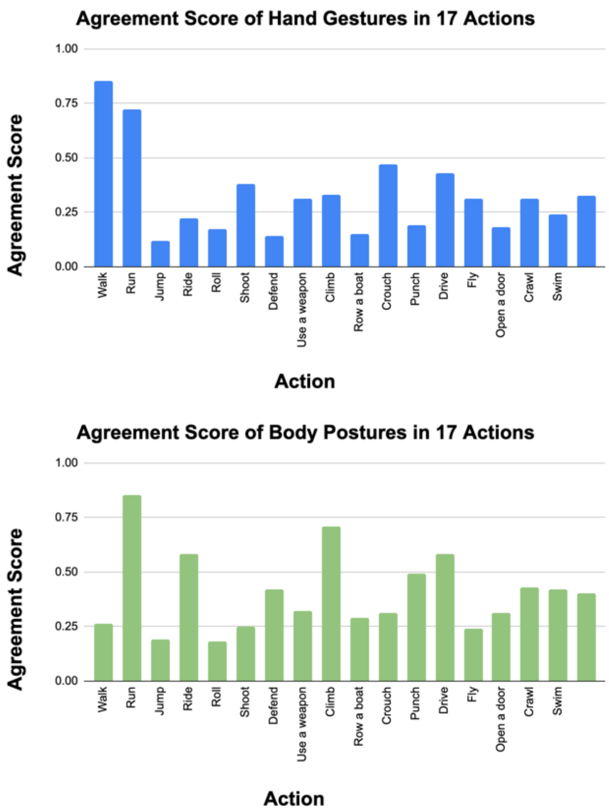

Figure 5: Agreement rates of the user-defined hand gestures and the upper-body postures for the actions on Study II.

and we reduced the set to 17 hand gestures and 17 upper-body postures. Figure 3 and Figure 4 showed the selected hand gestures and upper-body postures. The detailed design meaning of each gesture and posture was listed in our supplementary materials. To evaluate the degree of consensus among user-defined gestures/postures, we calculated the agreement score A using the equation of previous works [40, 41]:

$$A_t = \sum_{P_i} \left(\frac{|P_i|}{|P_t|}\right)^2 \tag{1}$$

where t is one of the actions, P_t is the set of collected gestures/postures for t, and P_i is a subset of identical gesture/posture from P_t. The range for A is [0,1]. The agreement rates of hand gestures and body postures are shown in Figure 5. For hand gestures, the agreement rates were from 0.12 (medium agreement, $0.100 < AR < 0.300$) to 0.85 (very high agreement, $AR > 0.500$). For body postures, the agreement rates were from 0.18 (medium agreement, $0.100 < AR < 0.300$) to 0.85 (very high agreement, $AR > 0.500$). The mean AR of hand gestures and body postures were (0.32, 0.40).

Four participants (*P3, P9, P10, P11*) preferred to use upper-body postures to control avatar actions that their upper body can represent. Half of the participants expressed some actions were appropriate for upper-body postures (e.g., *open a door*, *row a boat*, *use a weapon*, *drive*, *swim*, and *punch*). However, the participants wanted to use hand gestures when avatar actions contained lower-body or whole-body motions, such as *walk*, *run*, *jump*, and *crouch*. *P7* also mentioned that the actions which contained avatars left from the ground were also proper to represent by hand gestures (e.g., *fly* and *roll*). These results are consistent with the formative study. Besides, some gestures/postures were designed similarly (e.g., *walk* and *run*) because such gestures differed only on motion parameters (e.g., speed), which was also found in ARAnimator [41]. Some hand gestures (*climb* and *swim*, *fly* and *drive*) only differ from the hand position on the ground or the air. We will discuss how to distinguish these similar gestures in the system implementation section. Overall, participants controlled most avatar actions with the puppetry method that they saw their upper bodies or hands as puppets to manipulate avatars' motions. For example, the participant used the index finger and the middle finger as the avatar's legs and moved the two fingers to represent the avatar's walking. However, some gestures/postures were dominant to be chosen because they were most intuitive to human experiences, such as finger guns, in which participants made their hands like handguns to represent the gun action. We listed all design meanings of hand gestures and upper-body postures in the supplementary materials.

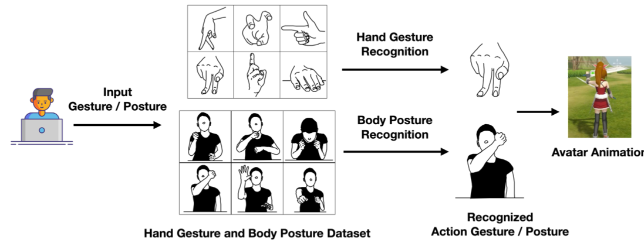

Figure 6: A system procedure of Puppeteer.

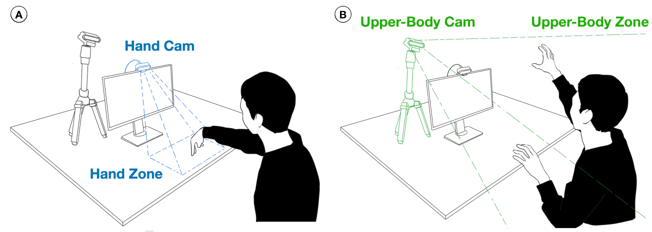

Figure 7: A system setup of Puppeteer. (a) *Hand Cam* detects a user's hand gestures in *Hand Zone*. (b) *Upper-Body Cam* recognizes the user's upper-body postures in *Upper-Body Zone*.

5 PUPPETEER SYSTEM IMPLEMENTATION

Based on the collected virtual avatar actions corresponding to defined hand gestures and upper-body postures, we developed a prototype input system named Puppeteer. Recently, many frameworks based on machine learning have enabled real-time hand and body keypoint detection from RGB frames. We chose Google MediaPipe[8]

[8]https://google.github.io/mediapipe/

framework for our gestures/postures keypoint detection. However, our camera capturing angle and detected targets differ markedly from those used to train public models, and these frameworks did not provide satisfying results to our system. We, therefore, develop our self-defined machine learning method for the specific needs.

The system detection procedure is shown on Figure 6. The participant inputs a hand gesture or an upper-body posture into the recognition system, and the system distinguishes the input gesture/posture and recognizes it as 1 of 17 avatar actions. Then, the system performs the corresponding avatar animation based on the action recognition. The recognition system was run in Python on a PC desktop, and avatar animations were implemented in Unity3D and Maximo. To decrease the confusion in distinguishing between hand gestures and upper-body postures, we designed two detection zone to separately detect hand and upper-body input, which were recorded by two cameras and shown in Figure 7. The camera for the hand zone (called *Hand Cam*) was placed on the top of a desktop screen, and one for the upper-body zone (called *Upper-Body Cam*) was placed on a tripod. The system first identifies whether participants input hand gestures. If the system gets the detection data from *Hand Cam*, it automatically changes to the hand gesture recognition mode and searches the gestures in the self-trained dataset. If the system does not get the hand detection data or can not recognize the input in the hand gestures dataset, it switches to recognize upper-body postures.

We discussed the detail of the gesture/posture recognition below.

5.1 Data Collection

To collect the selected hand gestures and upper-body postures for the self-trained datasets, we invited 12 participants to perform the gestures and postures. The participant was asked to perform gestures/postures like the demonstration videos shown on the screen. Each gesture/posture was repeated five times by each participant. We recorded the performed gestures and postures with *Hand Cam* and *Upper-Body Cam*. The two cameras were both Logitech 4K Webcam. We developed a simple graphic user interface (GUI) to show the views from the camera's recording. The GUI checked and stored the recorded videos to see if the performed gestures/postures keypoints could be detected correctly by the MediePipe framework. We created two datasets for collected hand gestures and upper-body postures separately. The total number of videos for hand gestures is 1020 (= 12 (users) × 17 (types of actions) × 5 (repetitions)), and the one for body postures is also 1020. We set the frame rate as 10fps, so there are 50 frames for each video.

5.2 Gesture/Posture Classification

Then, we applied the MediaPipe framework to real-time get hand and upper body keypoint detection on the videos. The MediaPipe API provides 21 keypoints for a hand and 25 keypoints for the upper body, and each keypoint contained position data (x,y,z.) We used these keypoints to generate three types of feature vectors for the recognition: (1) **Angles** – the angles between fingers and a palm, (2) **Distances** – the distances between two fingers, and (3) **Displacements** – the displacements between the x/y position of the current frame and that of the last frame. These features can know the degrees of finger/upper-body rotation, translation, and movement path of finger/upper-body motions. For each frame, the feature numbers of hand gestures are 73 (= 15 (angles) + 16 (distances) + 42 (displacements)) dimensions, and that of upper-body postures are 78 (= 12 (angels) + 18 (distances) + 48 (displacements)) dimensions. Because the numbers of the collected videos are few for recognition, we augmented our videos by resorting to frames' orders in one video. So we augmented videos more than 24 times, and the total number of videos for hand gestures and upper-body postures was 24,480 (= 1020 (numbers of original videos) × 24 (times)).

We defined 17 clusters for the avatar actions and labeled the collected videos to these clusters. We used a principal component analysis (PCA) reconstruction-error-based detector [13] as loss function to classify the gestures/postures to the action clusters. When the participant inputs a new gesture/posture, the system will calculate the distance between the new gesture and the 17 action clusters' centroids. The system recognizes an input gesture/posture belonging to the action with the lowest distance between them. Based on the optimization, we finally chose 200 feature dimensions for hand gestures recognition and 50 feature dimension for upper-body postures recognition.

5.3 System Evaluation

We performed a 3-fold cross-validation to evaluate the trained models. We randomly split the data from the 12 participants into three subsets, in which two for the training model and one for validating. Then the three subsets were switched as training data and validation data. Finally, we calculated the average accuracy of hand gestures is 90%, and upper-body postures detection is 91%. The accuracies of each action for hand gestures and upper-body postures are shown in confusion matrices (Figure 8).

The system can correctly recognize most actions for hand gestures above 85% recognition accuracy. *Use a weapon*, *climb*, *open a door*, and *swim* actions have the best accuracy (100%). Some actions are confused by the recognition, such as *defend* (74.1%) is recognized as *use a weapon*, and *drive* (70.7%) is identified as *punch* and *climb*. For the confusion of *defend*, it is because this action are similar as *use a weapon* in the *angle* and the *distance* features, which leads to *defend*'s wrong identification to *use a weapon*. For *drive*, the noticeable features are the movement of a thumb and an index finger, which is the same as *punch* and *climb*, and that causes a false distinction between the two actions.

For the upper-body postures recognition, most actions are discernible above 91% accuracy, especially *jump*, *roll*, *shoot*, and *use a weapon* have the best accuracy (100%). There are four pairs of actions are confused for the recognition: (1) *walk* (72.9%) versus *run* and *crouch*, (2) *defend* (69.5%) versus *punch*, (3) *row a boat* (78.0%) versus *fly* (71.2%), and (4) *crawl* (69.5%) versus *swim* and *walk*. For the first pair, because of the occlusion of hands detection, it makes *walk* easily to be recognized from other actions which only have differences in hands movement, such as *run* and *crouch*. For the second pair, the most obvious difference is the left arms movement, which causes the false identification between *defend* and *punch*. For the third pair, the participants do not move their torsos, and two hands are mainly beside torsos, so most *angles* features are the same in the two actions. Their only differences are hands' movements,

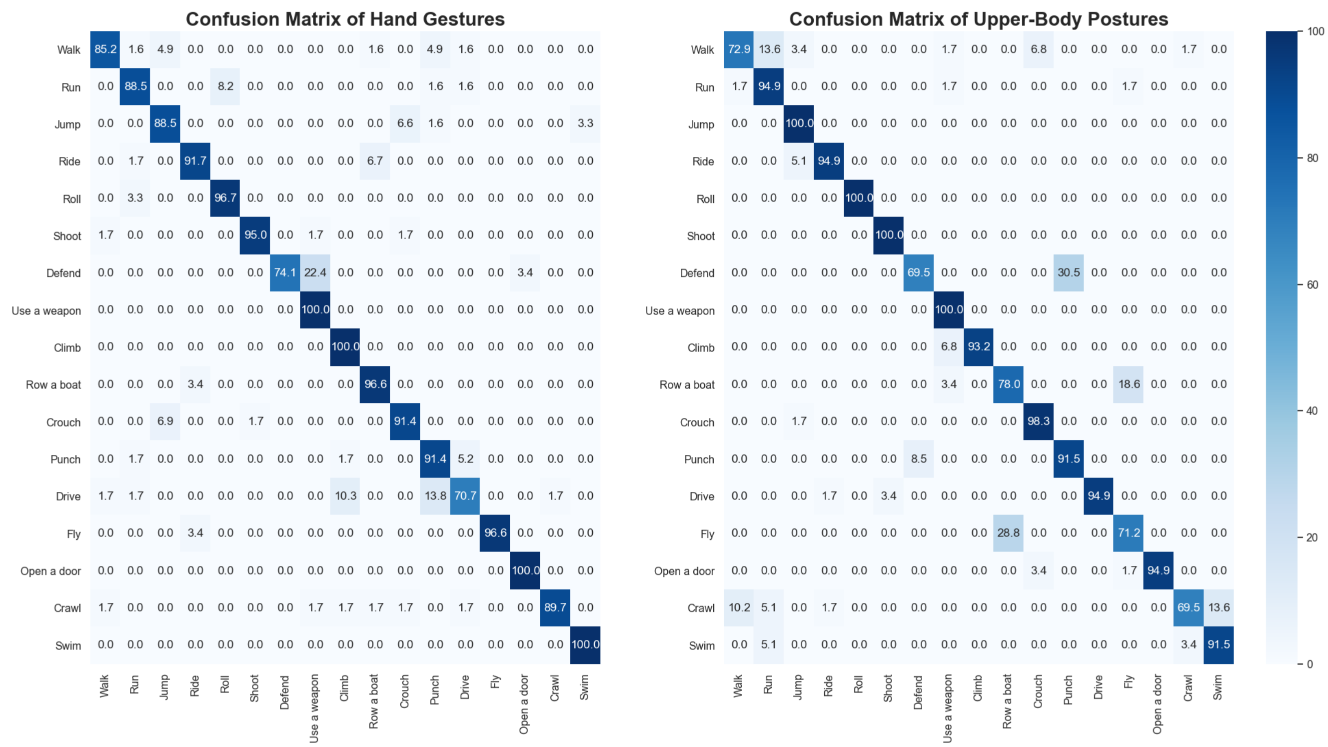

Figure 8: (Left) The confusion matrix of the hand gestures. (Right) The confusion matrix of upper-body postures.

which causes recognition confusion. For the last pair, *crawl* only differs at arms' movement to *swim* and hands' position to *walk*, which leads to the classification error.

In addition to the above discussion, some false recognitions are caused by individual behavior differences in performing the 17 actions. Another reason is that our collected numbers of data are still few for recognition. Improving the collected data will decrease the distinguish error of the system. Overall, the current accuracy of the system is sufficient for our demonstration applications.

6 APPLICATION

Game Scenarios	Avatar Actions
Action-Adventure Game	Walk, Run, Jump, Climb, Row a Boat, Punch, Drive, Fly, Open a Door, Crawl, Swim
RPG Game	Walk, Run, Jump, Ride, Roll, Shoot, Defend, Use a Weapon, Swim
Shooter Game	Walk, Run, Jump, Shoot, Use a Weapon, Crouch

Table 2: The action list of the three game scenarios shown in the application.

Followed on Study II, we created three game applications to demonstrate Puppeteer interaction, including three popular game genres – *Action-Adventure Game*, *RPG Game* and *Shooter Game*, as shown in 9. All actions in the three applications are shown in Table 2. The three games have some different actions for players to experience. The shooter game focuses on defeating enemies using guns and weapons. The adventure-action game and the RPG game provide more actions than the shooter game and focus on exploring the virtual environment.

Avatar Manipulation	Avatar Actions
Hand Gestures	Walk, Run, Jump, Roll, Crouch, Fly, Crawl
Upper-Body Postures	Ride, Shoot, Defend, Use a Weapon, Climb, Row a Boat, Punch, Drive, Open a Door, Swim

Table 3: The action list of using hand gestures and upper-body postures in the three scenarios.

According to the users' feedback in the gesture elicitation study, we designed the actions controlled by the hand gestures or upper-body postures and listed in Table 3. Players switch hand-controlled or upper-body-controlled manipulations while experiencing the applications. We described the detailed procedures of the three applications below. Players can perform *walk*, *run* and *jump* at any time because these actions are primary movement control and follow the hints to conduct the remaining actions during gaming.

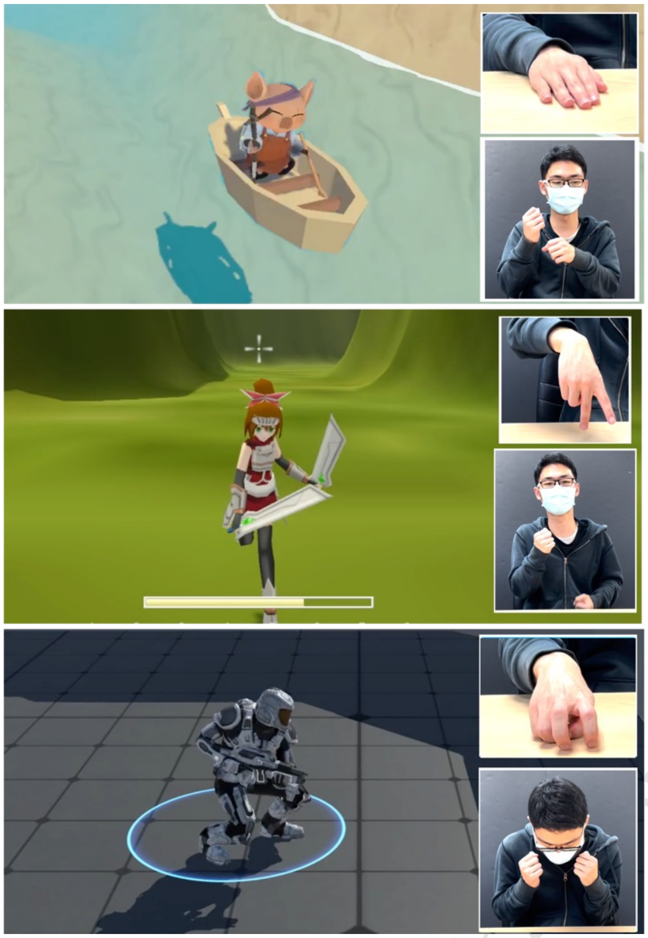

Figure 9: Three applications demonstrate usage of Puppeteer: Action-Adventure Game (upper), RPG Game (middle), and Shooter Game (bottom).

6.0.1 Action-Adventure Game. For the action-adventure game, eleven actions were experienced by players. The player controls a humanoid character to explore the virtual world. Each character initially has five health points. If the health points are empty, the game will end, and the players have to restart the game. First, the player is at the bottom of a valley and has to climb a mountain. When they achieve the top of the mountain, they find the bridge to the other peak broken, so they have to fly over mountains. However, there are some monsters on the other mountain, so the player starts a fight and maybe attacked by the monsters to decrease their health points. After they successfully beat the enemies, they continue on the adventure and find a building, opening a door and receiving some food to recover their health points. Next, they leave the building and drive a car down the mountain. They encounter a river to prevent their movement, so they change to row a boat to cross over the river. When they arrive at the edge of the river, they find a low height of rock cave, so they crawl over rock obstacles and finally find a box to get a treasure.

6.0.2 RPG Game. There are nine actions were experienced in the RPG game. The player becomes a knight, and their goals are to fight against a daemon and save the world. In the beginning, the player stands in a grassland. They walk in the prairie and find an ostrich, so they put on and ride the ostrich. Then, they encounter a barrier that blocks their way. They have to use a bow and shoot arrows to destroy the barrier. After they succeed, they meet a large lake and swim to cross over. When they arrive on the edge of the lake, the daemon appears. The player needs to use their swords to fight the enemy, a shield to defend against attacks, and a fast rollover to dodge attacks. After a fight, the player finally defeats the daemon and wins the game.

6.0.3 Shooter Game. In the shooter game, the player manipulates a soldier and needs to pass five levels to arrive at the destination. Some enemies and obstacles appear on the road to the destination, and they have to fight. The players performed six actions in the game. Automatic doors appear between two levels. The player moves to the next level and crouches down to get through a passage. Then, an obstacle obstructs the road, so the player uses a dagger to clean up the barrier. On some levels, there are enemies to attack the player. The player has to use their dagger to fight or a gun to shoot the enemies. After beating the enemies and successfully arriving at the last level, they win the game.

Each application contains the actions appropriate to represent by hand gestures and upper-body postures. Players will frequently switch between gestures and postures during gaming and experience the combination of the two manipulations.

7 LIMITATION AND FUTURE WORK

Although the system showed high accuracy for the gesture/posture recognition, the system evaluation only validates collected recorded data in a static setting. We need an extra study to observe the actual situation when participants play games and input many hand gestures and upper-body postures into the system. Besides, we separately detected hand gestures and upper-body postures for the system to recognize them easily. In the future, we want to improve our recognition algorithm and make the Puppeteer system only need one camera to detect all gestures and postures and successfully distinguish the two input techniques.

This paper focused on combining hand and upper-body input to control avatar actions. We implemented a prototype system to demonstrate this concept. We plan to extend our datasets to collect more hand gestures and upper-body postures for avatar control in future work. Besides, Puppeteer can be explored to apply in situations where it is not convenient to use their legs, such as people have legs hurt or playing games on mobile transportation. Puppeteer may provide a practical way for avatar control in the virtual environment in real life.

8 CONCLUSION

We present Puppeteer, a concept that combines hand gestures and upper-body postures to provide a new game input interaction by multiple cameras detection. We built a prototype using two cameras mounted on a screen and a tripod separately. We performed the MediaPipe framework for keypoint detection and the self-trained models for gesture/posture recognition. Based on the system evaluation, the Puppeteer achieves an average of 90% accuracy for upper-body postures and 91% for hand gestures detection. Three demonstration applications enabled by Puppeteer allows participants to switch to input hand gestures and upper-body postures to manipulate their virtual avatars. We believe Puppeteer provides a new avatar manipulation for convenient and easy interaction of hands and upper bodies in video games.

ACKNOWLEDGMENTS

Removed for review.

REFERENCES

[1] Karan Ahuja, Eyal Ofek, Mar Gonzalez-Franco, Christian Holz, and Andrew D. Wilson. 2021. CoolMoves: User Motion Accentuation in Virtual Reality. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 2, Article 52 (jun 2021), 23 pages. https://doi.org/10.1145/3463499

[2] Edwin Chan, Teddy Seyed, Wolfgang Stuerzlinger, Xing-Dong Yang, and Frank Maurer. 2016. User Elicitation on Single-Hand Microgestures. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 3403–3414. https://doi.org/10.1145/2858036.2858589

[3] Jiawen Chen, Shahram Izadi, and Andrew Fitzgibbon. 2012. *KinÊTre: Animating the World with the Human Body.* Association for Computing Machinery, New York, NY, USA, 435–444. https://doi.org/10.1145/2380116.2380171

[4] Yu-Ting Cheng, Timothy K Shih, and Chih-Yang Lin. 2017. Create a puppet play and interative digital models with leap Motion. In *2017 10th International Conference on Ubi-media Computing and Workshops (Ubi-Media)*. IEEE, 1–6.

[5] Mira Dontcheva, Gary Yngve, and Zoran Popović. 2003. Layered Acting for Character Animation. In *ACM SIGGRAPH 2003 Papers* (San Diego, California) *(SIGGRAPH '03)*. Association for Computing Machinery, New York, NY, USA, 409–416. https://doi.org/10.1145/1201775.882285

[6] Maxime Garcia, Remi Ronfard, and Marie-Paule Cani. 2019. Spatial Motion Doodles: Sketching Animation in VR Using Hand Gestures and Laban Motion Analysis. In *Motion, Interaction and Games* (Newcastle upon Tyne, United Kingdom) *(MIG '19)*. Association for Computing Machinery, New York, NY, USA, Article 10, 10 pages. https://doi.org/10.1145/3359566.3360061

[7] Oliver Glauser, Wan-Chun Ma, Daniele Panozzo, Alec Jacobson, Otmar Hilliges, and Olga Sorkine-Hornung. 2016. Rig Animation with a Tangible and Modular Input Device. *ACM Trans. Graph.* 35, 4, Article 144 (jul 2016), 11 pages. https://doi.org/10.1145/2897824.2925909

[8] Saikat Gupta, Sujin Jang, and Karthik Ramani. 2014. PuppetX: A Framework for Gestural Interactions with User Constructed Playthings. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces* (Como, Italy) *(AVI '14)*. Association for Computing Machinery, New York, NY, USA, 73–80. https://doi.org/10.1145/2598153.2598171

[9] Robert Held, Ankit Gupta, Brian Curless, and Maneesh Agrawala. 2012. 3D puppetry: a kinect-based interface for 3D animation.. In *UIST*, Vol. 12. Citeseer, 423–434.

[10] Narukawa Hiroki, Natapon Pantuwong, and Masanori Sugimoto. 2012. A puppet interface for the development of an intuitive computer animation system. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE, 3136–3139.

[11] Christian Holz and Andrew Wilson. 2011. Data miming: inferring spatial object descriptions from human gesture. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 811–820.

[12] An-Pin Huang, Fay Huang, and Jing-Siang Jhu. 2018. Unreal Interactive Puppet Game Development Using Leap Motion. In *Journal of Physics: Conference Series*, Vol. 1004. IOP Publishing, 012025.

[13] James A Jablonski, Trevor J Bihl, and Kenneth W Bauer. 2015. Principal component reconstruction error for hyperspectral anomaly detection. *IEEE Geoscience and Remote Sensing Letters* 12, 8 (2015), 1725–1729.

[14] Ji-Sun Kim, Denis Gračanin, Krešimir Matković, and Francis Quek. 2008. Finger walking in place (FWIP): A traveling technique in virtual environments. In *International Symposium on Smart Graphics*. Springer, 58–69.

[15] Fabrizio Lamberti, Gianluca Paravati, Valentina Gatteschi, Alberto Cannavo, and Paolo Montuschi. 2017. Virtual character animation based on affordable motion capture and reconfigurable tangible interfaces. *IEEE transactions on visualization and computer graphics* 24, 5 (2017), 1742–1755.

[16] Luís Leite and Veronica Orvalho. 2012. Shape Your Body: Control a Virtual Silhouette Using Body Motion. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI EA '12)*. Association for Computing Machinery, New York, NY, USA, 1913–1918. https://doi.org/10.1145/2212776.2223728

[17] Luis LEite and Veronica Orvalho. 2017. Mani-Pull-Action: Hand-Based Digital Puppetry. *Proc. ACM Hum.-Comput. Interact.* 1, EICS, Article 2 (jun 2017), 16 pages. https://doi.org/10.1145/3095804

[18] Hui Liang, Jian Chang, Ismail K Kazmi, Jian J Zhang, and Peifeng Jiao. 2017. Hand gesture-based interactive puppetry system to assist storytelling for children. *The Visual Computer* 33, 4 (2017), 517–531.

[19] Noah Lockwood and Karan Singh. 2012. Fingerwalking: motion editing with contact-based hand performance. In *Proceedings of the 11th ACM SIGGRAPH/Eurographics conference on Computer Animation*. 43–52.

[20] Zhiqiang Luo, I-Ming Chen, Song Huat Yeo, Chih-Chung Lin, and Tsai-Yen Li. 2010. Building hand motion-based character animation: The case of puppetry. In *2010 International Conference on Cyberworlds*. IEEE, 46–52.

[21] Meredith Ringel Morris, Andreea Danielescu, Steven Drucker, Danyel Fisher, Bongshin Lee, MC Schraefel, and Jacob O Wobbrock. 2014. Reducing legacy bias in gesture elicitation studies. *interactions* 21, 3 (2014), 40–45.

[22] Yoshihiro Okada. 2003. Real-time character animation using puppet metaphor. In *Entertainment Computing*. Springer, 101–108.

[23] Masaki Oshita, Yuta Senju, and Syun Morishige. 2013. Character Motion Control Interface with Hand Manipulation Inspired by Puppet Mechanism. In *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry* (Hong Kong, Hong Kong) *(VRCAI '13)*. Association for Computing Machinery, New York, NY, USA, 131–138. https://doi.org/10.1145/2534329.2534360

[24] Siyou Pei, Alexander Chen, Jaewook Lee, and Yang Zhang. 2022. Hand Interfaces: Using Hands to Imitate Objects in AR/VR for Expressive Interactions. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 166–181.

[25] Mose Sakashita, Tatsuya Minagawa, Amy Koike, Ippei Suzuki, Keisuke Kawahara, and Yoichi Ochiai. 2017. You as a Puppet: Evaluation of Telepresence User Interface for Puppetry. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 217–228. https://doi.org/10.1145/3126594.3126608

[26] Andrea Sanna, Fabrizio Lamberti, Gianluca Paravati, Gilles Carlevaris, and Paolo Montuschi. 2013. Automatically mapping human skeletons onto virtual character armatures. In *International Conference on Intelligent Technologies for Interactive Entertainment*. Springer, 80–89.

[27] Eunbi Seol and Gerard J Kim. 2019. Handytool: Object manipulation through metaphorical hand/fingers-to-tool mapping. In *International Conference on Human-Computer Interaction*. Springer, 432–439.

[28] Yeongho Seol, Carol O'Sullivan, and Jehee Lee. 2013. Creature Features: Online Motion Puppetry for Non-Human Characters. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Anaheim, California) *(SCA '13)*. Association for Computing Machinery, New York, NY, USA, 213–221. https://doi.org/10.1145/2485895.2485903

[29] Paul Skalski, Ron Tamborini, Ashleigh Shelton, Michael Buncher, and Pete Lindmark. 2011. Mapping the road to fun: Natural video game controllers, presence, and game enjoyment. *New Media & Society* 13, 2 (2011), 224–242.

[30] Christian Steins, Sean Gustafson, Christian Holz, and Patrick Baudisch. 2013. Imaginary Devices: Gesture-Based Interaction Mimicking Traditional Input Devices. In *Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Munich, Germany) *(MobileHCI '13)*. Association for Computing Machinery, New York, NY, USA, 123–126. https://doi.org/10.1145/2493190.2493208

[31] Ron Tamborini and Paul Skalski. 2006. The role of presence in the experience of electronic games. *Playing video games: Motives, responses, and consequences* 1 (2006), 225–240.

[32] Shan-Yuan Teng, Tzu-Sheng Kuo, Chi Wang, Chi-huan Chiang, Da-Yuan Huang, Liwei Chan, and Bing-Yu Chen. 2018. PuPoP: Pop-up Prop on Palm for Virtual Reality. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 5–17. https://doi.org/10.1145/3242587.3242628

[33] Amato Tsuji and Keita Ushida. 2021. Telecommunication Using 3DCG Avatars Manipulated with Finger Plays and Hand Shadow. In *2021 IEEE 10th Global Conference on Consumer Electronics (GCCE)*. IEEE, 39–40.

[34] Amato Tsuji and Keita Ushida. 2021. A Telepresence System using Toy Robots the Users can Assemble and Manipulate with Finger Plays and Hand Shadow. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 661–662.

[35] Amato Tsuji, Keita Ushida, and Qiu Chen. 2018. Real Time Animation of 3D Models with Finger Plays and Hand Shadow. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*. 441–444.

[36] Ying-Chao Tung, Chun-Yen Hsu, Han-Yu Wang, Silvia Chyou, Jhe-Wei Lin, Pei-Jung Wu, Andries Valstar, and Mike Y. Chen. 2015. User-Defined Game Input for Smart Glasses in Public Space. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*.

Association for Computing Machinery, New York, NY, USA, 3327–3336. https://doi.org/10.1145/2702123.2702214

[37] Yusuke Ujitoko and Koichi Hirota. 2019. Interpretation of tactile sensation using an anthropomorphic finger motion interface to operate a virtual avatar. *arXiv preprint arXiv:1902.07403* (2019).

[38] Bo-Xiang Wang, Yu-Wei Wang, Yen-Kai Chen, Chun-Miao Tseng, Min-Chien Hsu, Cheng An Hsieh, Hsin-Ying Lee, and Mike Y. Chen. 2020. *Miniature Haptics: Experiencing Haptic Feedback through Hand-Based and Embodied Avatars.* Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3313831.3376292

[39] Meng Wang, Kehua Lei, Zhichun Li, Haipeng Mi, and Yingqing Xu. 2018. TwistBlocks: Pluggable and Twistable Modular TUI for Armature Interaction in 3D Design. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction* (Stockholm, Sweden) *(TEI '18)*. Association for Computing Machinery, New York, NY, USA, 19–26. https://doi.org/10.1145/3173225.3173231

[40] Jacob O. Wobbrock, Htet Htet Aung, Brandon Rothrock, and Brad A. Myers. 2005. Maximizing the Guessability of Symbolic Input. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems* (Portland, OR, USA) *(CHI EA '05)*. Association for Computing Machinery, New York, NY, USA, 1869–1872. https://doi.org/10.1145/1056808.1057043

[41] Hui Ye, Kin Chung Kwan, Wanchao Su, and Hongbo Fu. 2020. ARAnimator: In-Situ Character Animation in Mobile AR with User-Defined Motion Gestures. *ACM Trans. Graph.* 39, 4, Article 83 (jul 2020), 12 pages. https://doi.org/10.1145/3386569.3392404

[42] Wataru Yoshizaki, Yuta Sugiura, Albert C. Chiou, Sunao Hashimoto, Masahiko Inami, Takeo Igarashi, Yoshiaki Akazawa, Katsuaki Kawachi, Satoshi Kagami, and Masaaki Mochimaru. 2011. An Actuated Physical Puppet as an Input Device for Controlling a Digital Manikin. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 637–646. https://doi.org/10.1145/1978942.1979034

[43] Fan Zhang, Shaowei Chu, Ruifang Pan, Naye Ji, and Lian Xi. 2017. Double hand-gesture interaction for walk-through in VR environment. In *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*. IEEE, 539–544.

[44] Yupeng Zhang, Teng Han, Zhimin Ren, Nobuyuki Umetani, Xin Tong, Yang Liu, Takaaki Shiratori, and Xiang Cao. 2013. BodyAvatar: Creating Freeform 3D Avatars Using First-Person Body Gestures. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (St. Andrews, Scotland, United Kingdom) *(UIST '13)*. Association for Computing Machinery, New York, NY, USA, 387–396. https://doi.org/10.1145/2501988.2502015