Multimedia Tools and Applications (2020) 79:17707–17730
https://doi.org/10.1007/s11042-019-08520-1

Design and evaluation of a hand gesture recognition approach for real-time interactions

Vaidyanath Areyur Shanthakumar[1] · Chao Peng[2] · Jeffrey Hansberger[3] · Lizhou Cao[2] · Sarah Meacham[4] · Victoria Blakely[4]

Received: 30 November 2018 / Revised: 19 November 2019 / Accepted: 26 November 2019 / Published online: 21 February 2020

Abstract
Hand gestures are a natural and intuitive form for human-environment interaction and can be used as an input alternative in human-computer interaction (HCI) to enhance usability and naturalness. Many existing approaches have employed vision -based systems to detect and recognize hand gestures. However, vision-based systems usually require users to move their hands within restricted space, where the optical device can capture the motion of hands. Also, vision-based systems may suffer from self-occlusion issues due to sophisticated finger movements. In this work, we use a sensor-based motion tracking system to capture 3D hand and finger motions. To detect and recognize hand gestures, we propose a novel angular-velocity method, which is directly applied to real-time 3D motion data streamed by the sensor-based system. Our approach is capable of recognizing both static and dynamic gestures in real-time. We assess the recognition accuracy and execution performance with two interactive applications that require gesture input to interact with the virtual environment. Our experimental results show high recognition accuracy, high execution performance, and high-levels of usability.

Keywords Gesture-based interactive system · Motion tracking · Gesture recognition

1 Introduction

Hand gestures are considered to be an input modality for human-computer interaction (HCI) but not widely used as a primary method of interaction in computer systems. Nowadays, most HCI applications are driven by keyboard and mouse operations to manipulate digital information because of their high degree of precision and ease of providing input by clicking

✉ Chao Peng
cxpigm@rit.edu

1 Computer Science Department, University of Alabama in Huntsville, Huntsville, AL 35899, USA

2 Golisano College of Computing and Information Sciences, Rochester Institute of Technology, 1 Lomb Memorial Drive, Rochester, NY 14623, USA

3 Army Research Lab, Huntsville, AL,USA

4 SMAP Center, University of Alabama in Huntsville, Huntsville, AL 35899, USA

buttons; however, the keyboard-mouse method does not allow users to directly interact with information in the same way we interact with objects by hands.

Hutchins et al. [16] defined the gulf of execution as the gap between the user's goal and the machine instructions that the user needs to follow to accomplish the goal. The gap must be filled with the user's activities to interact with the machine. A smaller gulf of execution will enable faster and more efficient activities to accomplish the goal with a smaller chance of errors. Touch-based gestures produce a direct and intuitive way to interact with digital information. With an appropriate touch user interface (TUI) design, interactions with the touch screen can be faster and more intuitive than interactions with keyboard and mouse. As mentioned by Kieras et al. [18], TUIs reduce the gulf of execution as it is a more direct form of interaction between the user and information than the keypad interface. However, TUIs still do not provide the same degree of freedom and flexibility as hand gestures in a 3D environment. Hand gestures performed without touching the screen extend the interactions beyond a 2D plane of glass (e.g., tapping, sliding, and dragging). Such in-air gestures can be an input modality for touchless user interfaces, which allow users to operate the interface like interacting with real-world objects. Touchless user interfaces with in-air gesture controls could further reduce the gulf of execution between the user and digital information. It can also reduce problems such as *fat finger* [36] or *occlusion* [39] caused by restricted touch space.

In this work, we detect and recognize in-air hand gestures, which are natural and intuitive to perform and can be used to interact with touchless user interfaces. Many existing gesture recognition methods use machine learning and computer vision algorithms to achieve high gesture recognition accuracy (e.g., [24, 27, 41]). However, they suffer slow execution and cannot be used in real-time applications requiring high-frequency interactions such as video games and interactive tools. The goal of this work is to design a gesture recognition system that satisfies the requirement of high accuracy and fast execution performance for real-time applications.

Contributions In this work, we present an adaptive hand model which is articulated using the fingers that have a high degree of independence. We develop a method based on angular velocities to recognize both static and dynamic hand gestures in real-time. We demonstrate the use of this method in recognizing a vocabulary of eight natural hand gestures. Different from machine learning approach, our angular-velocity method do not require data preprocessing like data collection, annotation, or training. The accuracy and execution performance of our method are evaluated with the user study of two experimental applications that require high-frequency interactions. The results show high recognition accuracy, fast execution performance, and high levels of usability.

The rest of the paper is organized as follows. Section 2 reviews existing vision-based and sensor-based methods for gesture recognition, which are related to the work presented in this paper. We present the adaptive hand model in Section 3. The angular-velocity method and the gesture vocabulary are presented in Section 4. Section 5 presents the experimental design. The results of the experimental study are discussed in Section 6. Section 7 concludes our work and proposes the future work.

2 Related work

Hand gesture-based interactions have received increasing attention in HCI and have the potential of being an effective way to interact with computers [35]. To support complex interactions, the design of hand gestures need to be ergonomic [26, 42], intuitive [28], bimanual (performed with both hands), and pantomimic (meaningful without speech) [1]. Nielsen et al. [28] considered human factors in HCI and presented a procedure to map a gesture vocabulary towards functionality. Related to our work, this section reviews both vision-based and sensor-based (non-optical) gesture recognition methods.

2.1 Vision-based methods

Vision-based methods usually use cameras to capture colors and depths [40]. An advantage of using a vision-based motion capture device for gesture recognition is that there is no need to wear any form of sensors, unlike wearable motion capture devices (such as gloves). Rautaray and Agrawal [33] mentioned that vision-based methods have three phases: detection, tracking, and recognition. Song et al. [37] detected gestures using a stereo camera to reconstruct body postures in 3D and then built features that were fed into a classifier, but the recognition accuracy was low. Wang et al. [41] used a Microsoft Kinect and converted the captured hand shapes into super pixels. The gestures were recognized using a distance metric to measure the dissimilarity between hand shapes. Liu et al. [22] applied a rule-based approach on stereo images. Their approach recognized seven dynamic hand gestures and six static (finger spelling numbers) hand gestures. Marin et al. [25] used a Microsoft Kinect and a Leap Motion device to capture images and motion data. Hand gestures were recognized by using support vector machines. Sharma and Verma [35] extracted hand shapes and orientations from video streams, and recognized six static hand gesture images. However, their approach required stable lighting and simple background conditions.

The high accuracy of gesture was reported in those vision-based methods (e.g., over 90% accuracy in [22, 25, 41]), but the accuracy with vision-based methods may be affected by environmental factors [29, 40], such as the number of cameras, lighting conditions, or the capturing range of the cameras.

2.2 Sensor-based methods

Non-optical motion sensors can be used to tracking motions by holding them in hands or attaching them on bodies, such as Wii controller, CyberGlove [17], and inertial systems [4]. Different from optical motion sensors, they are not sensitive to environmental factors like lighting conditions or cluttered backgrounds.

Xu [43] used a Cyber Glove to collect driving motion data, and then they used the data to train a feed-forward neural network for gesture recognition. Neto et al. [27] used a CyberGlove and artificial neural networks to recognize communicative gestures for human-robot interactions. However, their approach could be time-consuming on dynamic gesture recognition and may not support real-time interactions. Lu et al. [23] collected motion data from a custom forearm wearable device, and they utilized a linear classifier and a dynamic time-warping algorithm to recognize hand gestures. However, their approach needed 300 milliseconds to respond to each gesture. Luzhnica et al. [24] developed a custom motion tracking glove and employed a window-sliding method. The execution efficiency of their approach highly depended on the window size. For example, a 50-frame window size resulted in a delay of 0.625 seconds on the system response. Alavi et al. [2] developed a multi-sensor motion capturing system and detected arm and upper-body gestures by using support vector machines and neural networks. Their approach worked well for the users who participated in the process of data collection, but resulted in lower recognition accuracy for the users who did not participate in the process of data collection. Also, the execution of their approach was not fast enough to support high-frequent interactions. Thus, Those existing sensor-based methods reported a high recognition accuracy (above 90%), but they suffer slow execution, so that they may not be suitable for real-time applications.

In our previous work, we designed and evaluated sensor-based hand gesture applications. Based on the results of our previous studies, we are positive on the use of hand gestures in HCI. In the work of Hansberger et al. [13], three hand gestures were adopted to evaluate the levels of arm fatigue within in a gaming environment. The results showed that the fatigue levels from using the supported hand gestures were similar to the use of keyboard. Supported gestures and keyboard interactions produced less fatigue than mid-air type of gestures. Peng et al. [31] presented the design diagram of a gesture-based image categorization interface in a virtual environment. The work focused on the gesture design and the design of functional operations. Peng et al. [30] studied user experience of using gesture input modality in video games. The work discussed the mapping of hand gestures to commands to control the movement of game characters. The video game and motion tracking equipment used in the experiment of that work were the same as the game and equipment used in this work. Hansberger et al. [12] proposed the concept of integrating hand gestures in a multimodal interface in the virtual environment. However, these previous work did not present details of the gesture recognition method and did not include evaluations on recognition accuracy and execution performance. Diliberti et al. [9] developed custom motion capturing gloves to capture hand and finger movements. They also presented a neural network method to train a gesture classification model and use it to recognize new hand gestures. Although the work achieved a high recognition accuracy, a lot of laborious work were involved in order to collect training data sets and tune neural network training parameters.

In this work, we use an inertial measurement unit (IMU) based motion capture system. An IMU is a motion tracking sensor which is usually composed of one or more accelerometers, gyroscopes, and magnetometers. It can produce the 3D rotation value in itself, based on a combination of measurements from the accelerometers, gyroscopes, and magnetometers. An IMU is low cost and in a small size, and it usually requires low processing power. IMUs do not require extensive space for the setup and do not cause self-occlusion issues on hands, which otherwise may occur with optical devices when hands are in motion. In addition, our approach does not require a training phase prior to the phase of recognition.

3 Hand model articulation

The hand model used in this work is a replica of a human's skeletal hand. The model adapts to different hand proportions and supports a fast and accurate gesture recognition from complex hand and finger motions. The hand gestures are directly used for giving authoritative instructions and manipulating objects.

3.1 Degree of finger independence

The biomechanics and physiology of the hand lead to some limitations on the type of motions capable with the hand and fingers. The *degree of finger independence* refers to the amount of muscle-controlled movement versus the amount of passive movement coupled with the movement of other fingers [19]. The fingers of a hand except the thumb are not fully independent.

Based on the previous studies in the field of hand biomechanics (e.g.,[11, 19]), the thumb shows the highest degree of independence. The index finger is the second, and the middle finger is the third. The thumb, index, and middle fingers are the primary fingers contributing to the creation of gestures. For example, the index finger is usually preferred to create a pointing gesture. A thumbs-up is a natural gesture to show approval or satisfaction. In contrast, the ring and pinky fingers are seldom used in a natural gesture.

3.2 Articulation

The articulation of our hand model represents the hand with nine finger joints and one wrist joint. A joint is the junction between two bones. Each finger is represented with three joints. Taken the degree of finger dependence into consideration, our approach uses three fingers: *thumb*, *index finger*, and *middle finger*. The joints of the hand model are denoted as $J = \{w, t_j, i_j, m_j\}$, where j is an integer index in the range of [1, 3]. As illustrated in Fig. 1a, w is the wrist joint, t_j is a joint of the thumb, i_j is a joint of the index finger, m_j is a joint of the middle finger. A vector is constructed between two neighboring joints to represent the bone. Let's denote a vector $\overrightarrow{cd}$ constructed from the joint c to d. Thus, the proximal phalanx of the index finger can be represented as the vector $\overrightarrow{i_1 i_2}$. We assume joint positions are in the world (global) coordinate system, rather than based on relative bone rotations within the skeletal hierarchy. We define a local coordinate system, whose origin is at the chest joint of the avatar, as shown in Fig. 1b. The vector $\vec{y}$ is consistent to the world y axis. The vector $\vec{z}$ corresponds to the direction that the user faces to. To obtain $\vec{z}$, we first compute an assisting vector $\vec{a}$ with left and right shoulder

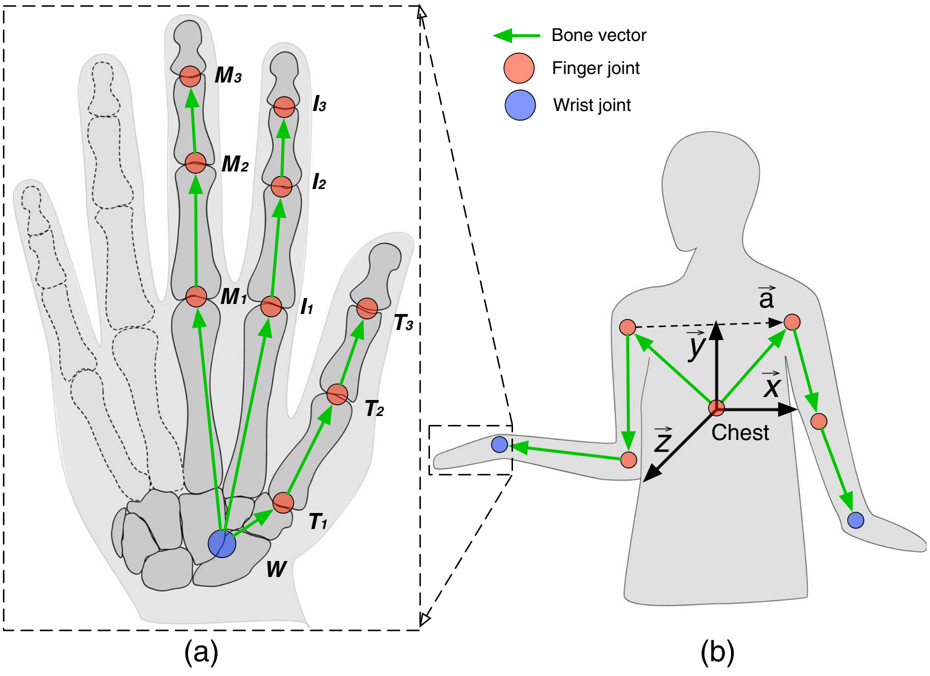


Fig. 1 The upper body and hand articulation

joints, denoted as s_l and s_r, respectively. Equation (1) shows the metric to compute those axes.

$$\begin{aligned}
\vec{y} &=< 0, 1, 0 >;\\
\vec{a} &= (s_l - s_r)/(\|s_l - s_r\|);\\
\vec{z} &= (\vec{a} \times \vec{y})/(\|\vec{a} \times \vec{y}\|);\\
\vec{x} &= \vec{y} \times \vec{z};
\end{aligned} \tag{1}$$

The local coordinate system is dynamically established based on the body posture and the facing direction. With the local coordinate system, users are free to adjust their body positions while performing a hand gesture, such as bending their body forward or turning and laying their back on the couch.

4 Gesture recognition

Hand gestures can be classified into two categories: static gestures and dynamic gestures [15, 29]. Static gestures are postures where the hand and fingers do not move temporally. For example, pointing with the index finger is a static gesture. Theoretically, any posture can be a static gesture, but a high similarity in postures may increase the error rate of interpreting the gesture meaning. Dynamic gestures involve physical movements of the hand and fingers in two or three dimensions. A dynamic gesture is usually composed of three stages: *start*, *update*, and *end* [32]. The start stage detects an initial posture that satisfies the gesture's eligibility criteria. The end stage detects an exclusive posture that indicates the completion of the gesture. The update stage detects in-between changes within the sequence of satisfactory postures. For example, waving hands at somebody is a dynamic gesture.

Our proposed angular-velocity method is capable of recognizing both static and dynamic gestures. In this section, we first present the details of the angular-velocity method, and then we describe the design of the gesture vocabulary, which contains both static and dynamic gestures and can be recognized by the angular-velocity method.

4.1 Angular-velocity method

We use vectors and angles to determine the gesture a user performs. In our method, $\overrightarrow{wm_1}$ and $\overrightarrow{wi_1}$ are used to define the plane of the palm. We define the palm vector, denoted as $\vec{n}$, which is outward-pointing normal vector from the palm plane. It is equal to the normalized cross product of $\overrightarrow{wi_1}$ and $\overrightarrow{wm_1}$. For the left hand, $\vec{n} = (\overrightarrow{wm_1} \times \overrightarrow{wi_1})/(\|\overrightarrow{wm_1} \times \overrightarrow{wi_1}\|)$. For the right hand, $\vec{n} = (\overrightarrow{wi_1} \times \overrightarrow{wm_1})/(\|\overrightarrow{wi_1} \times \overrightarrow{wm_1}\|)$. We also define the vector pointing to the side of the palm, denoted as $\vec{u}$, which is equal to the normalized cross product of $\overrightarrow{wm_1}$ and $\vec{n}$. We also define the parameters of *palm directional angle* and *hand postural angle*. The palm directional angle indicates the facing direction of the palm. The hand postural angle indicates the relative posture to the user's body. The acceptance of the initial palm directional angle and hand postural angle is evaluated in the local coordinate system defined by (1).

We define the parameter of *performing time*, which is the time duration that continuously satisfies the gesture parameter requirements. For static gestures, the value of performing time begins accumulating when the first matched posture is found. The value stops accumulating when the posture does not meet the requirements. For dynamic gestures, the value begins accumulating after the start stage, and it stops when reaching the end stage. A static gesture can be detected at the moment when a mismatched posture occurs. A dynamic gesture can be detected in the end stage. In such a moment or stage, if the performance time is in the range of the performing time thresholds, the gesture is detected; otherwise, no gesture returns. Before proceeding to detect another gesture, the value of performing time is set to zero.

For dynamic gestures, the parameter of *angular velocity* checks if the motion of the hand and fingers meet the gesture requirements. While performing a dynamic gesture, the hand movement speed may not always be constant. It is possible that the speed falls out of the threshold range for just a few milliseconds and quickly resumes back into the range. People do not perceive this as two separate dynamic gestures, but the method does. To avoid this anomaly, we define the parameter of *break time* for dynamic gesture recognition. It ensures that a minimum amount of time has to pass before recognizing another dynamic gesture. The value of the break time begins accumulating from zero if the end stage has detected a dynamic gesture. The process to recognize another gesture can start after the break time has reached the threshold range.

4.2 Gesture vocabulary

We designed a vocabulary of gestures with the goal of reducing the burden on users to learn, remember or recall gestures, and make those gestures closely related to the users' natural hand motions. The gestures in our vocabulary are conscious gestures [6] that have meanings without speech as opposed to spontaneous gestures having meanings only within the context of speech. The gestures in the vocabulary are *swipe*, *stop*, *come*, *go*, *pinch*, *drop-in*, *trash-out*, and *pointing*. Figure 2 illustrates the gestures and their vector representations.

In a real hand kinematic model, a hand can be effectively controlled by 27 degrees of freedom (DOF), which include 6 DOFs for the position and orientation of the wrist, 5 DOFs on the thumb, and 4 DOFs on other fingers [20]. Because of movement constraints on the joints, certain hand postures are impossible to perform [20]. For example, fingers can be moved backwards only up to an extent [21]. The wrist can move backwards or forward, but the wrist movement cannot make the palm or dorsal side of the hand touch the forearm. We consider such constraints and angle limits on the joints in our angular-velocity method, and we empirically determine the threshold values of the parameters for each of the gestures, as shown in Table 1, 2, 3, 4, 5 and 6. The following subsections describe the characteristics of each gesture in the vocabulary.

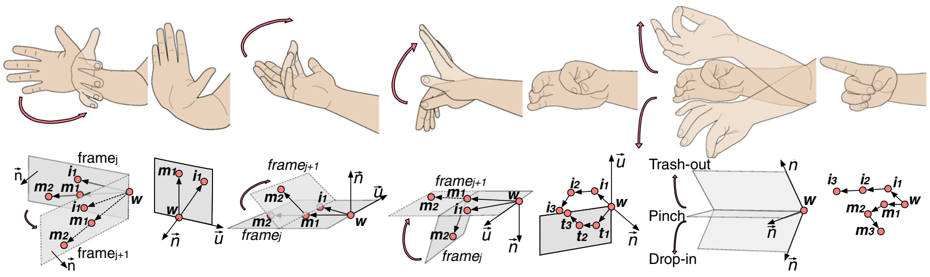


Fig. 2 The gestures (top) and their vector representations (bottom). From left to right, the gestures are swipe, stop, come, go, pinch, drop-in, trash-out, and pointing

Table 1 The thresholds of the parameters used in the swipe gesture

Parameters	Threshold ranges
Swipe palm directional angle $(acos(\overrightarrow{n} \cdot \overrightarrow{y}))$	$[65°, 115°]$
Swipe hand postural angle $(acos(\overrightarrow{wm_1} \cdot \overrightarrow{m_1m_2}))$	$[120°, 180°]$
Swipe angular velocity $(\frac{acos(\overrightarrow{wm_1}^{f_j} \cdot \overrightarrow{wm_1}^{f_{j+1}})}{\Delta t})$	$[900°/\text{sec}, +\infty)$
Swipe performing time	$[0.016$ secs, $+\infty)$
Swipe break time	$[0.1$ secs, $+\infty)$
Swipe direction $(\overrightarrow{wm_1}^{f_j} \times \overrightarrow{wm_1}^{f_{j+1}} \cdot \overrightarrow{y})$	$[-\infty, 0)$ - left hand
	$(0, +\infty]$ - right hand

Table 2 Parameters and threshold ranges used in the stop gesture

Parameters	Threshold ranges
Stop palm directional angle $(acos(\overrightarrow{n} \cdot \overrightarrow{z}))$	$[0°, 25°]$
Stop hand postural angle $(acos(\overrightarrow{u} \cdot \overrightarrow{x}))$	$[0°, 25°]$
Stop performing time	$[0.016$ secs, $+\infty]$

Table 3 Parameters and threshold ranges used in the come and go gesture

Parameters	Threshold ranges	
	Come	Go
Come/Go palm directional angle $(acos(\overrightarrow{n} \cdot \overrightarrow{y}))$	$[0°, 30°]$	$[145°, 180°]$
Come/Go hand postural angle $(acos(\overrightarrow{u} \cdot -\overrightarrow{x}))$	$[0°, 30°]$	$[0°, 45°]$
Come/Go angular velocity $(\frac{(acos(\overrightarrow{wm_1}^{f_j} \cdot \overrightarrow{m_1m_2}^{f_j}) - acos(\overrightarrow{wm_1}^{f_{j+1}} \cdot \overrightarrow{m_1m_2}^{f_{j+1}}))}{\Delta t})$	$[100°/\text{sec}, +\infty)$	$[200°/\text{sec}, +\infty)$
Come/Go performing time	$[0.016$ secs, $+\infty)$	
Come/Go break time	$[0.1$ secs, $+\infty)$	

Table 4 Parameters and threshold ranges used in the pinch gesture

Parameters	Threshold ranges
Pinch palm directional angle $(acos(\overrightarrow{n} \cdot \overrightarrow{z}))$	$[60°, 120°]$
Pinch hand postural angle $(acos(\overrightarrow{u} \cdot \overrightarrow{z}))$	$[50°, 130°]$
Pinch distance($\|i_3 - t_3\|$)	$[0, 1$cm$]$

Table 5 Parameters and threshold ranges used in the drop-in and trash-out gestures

Parameters	Threshold ranges	Gesture types
Drop palm directional angle $(acos(\overrightarrow{n} \cdot \overrightarrow{y}))$	$[0°, 55°)$	Trash-out
	$(125°, 180°]$	Drop-in

Table 6 Parameters and threshold ranges used in the pointing gesture

Parameters	Threshold ranges
Pointing finger Bend Angle $(acos(\overrightarrow{a_1a_2} \cdot \overrightarrow{a_2a_3}))$	[40°, 90°]
Pointing Finger Bend Angle (extended fingers) $(acos(\overrightarrow{i_1i_2} \cdot \overrightarrow{i_2i_3}))$	[150°, 200°]
Pointing finger orientation angle $(acos(\overrightarrow{i_2i_3} \cdot \vec{z}))$	[-30°, 30°]

4.2.1 Swipe

A swipe gesture is characterized by a hand motion of the wrist rotating while the palm is facing sideways and moving towards the direction it is facing. The start stage recognizes the palm fairly straight and facing sideways towards the other hand. The threshold values are specified in Table 1. A perfect palm directional angle is 90 degrees while the ideal hand position would be at 180 degrees. The hand postural angle checks if the hand is straight. In particular for swipe gesture, during the update stage, our method checks the *swipe angular velocity*, which is produced by the wrist rotation. It is defined as an angle value from two vectors, $\overrightarrow{wm_1}^{f_j}$ and $\overrightarrow{wm_1}^{f_{j+1}}$, where $\overrightarrow{wm_1}^{f_j}$ is the $\overrightarrow{wm_1}$ at the frame j, and $\overrightarrow{wm_1}^{f_{j+1}}$ is the $\overrightarrow{wm_1}$ at the next consecutive frame. In our method, the left hand only performs a swipe to the right and vice versa. To ensure this, we define a *swipe direction* parameter. If any parameters fall out of the threshold range, the end stage is triggered.

4.2.2 Stop

A stop gesture is characterized by an open palm facing outwards from the front of the body with four fingers straight up and together. We assume that $\overrightarrow{wm_1}$ represents the hand pointing direction. Table 2 lists the parameter threshold ranges for the stop gesture.

4.2.3 Come and Go

A come gesture starts with a posture where the palm faces upwards and the fingers are together and straight; then, the four finger tips, primarily the middle fingertip, move towards the hollow of the palm. A go gesture provides the inverse command of a come gesture. It is characterized by the palm of the hand facing downwards and the fingers bent towards the palm as the starting posture of the gesture. Then, the four fingers move away from the palm (stretched out) so as to make the palm fairly straight.

The required palm facing direction in the start stage is evaluated by the palm directional angle. Come and go gestures also require the hand to be positioned in front of the body, which is determined by the hand postural angle. The palm directional angle and the hand postural angle have threshold ranges because people may not be able to comfortably pose the palm at the ideal angle due to individual differences. Table 3 lists the parameters used in the come and go gestures and their threshold ranges.

In the update stage, we check the angular rotation velocity at the first and second joints of the middle finger. As indicated in Table 3, the angular velocity is defined as the difference between the angle formed by $\overrightarrow{wm_1}^{f_j}$ and $\overrightarrow{m_1m_2}^{f_j}$ and the angle formed by $\overrightarrow{wm_1}^{f_{j+1}}$ and $\overrightarrow{m_1m_2}^{f_{j+1}}$. The angular velocity is a positive value for a come gesture in which the middle finger moves towards the palm; and it is a negative value for a go gesture. The recognition process moves to the end stage if an angle parameter or the velocity parameter is out of the threshold ranges.

4.2.4 Pinch

Pinch is a gesture performed using the thumb and the index finger and pinching them together to hold an object. It can be used in conjunction with other gestures. For example, a person can pick an object, and move it with arm/hand movements or throw it away with a wrist rotation.

A pinch gesture requires the palm facing sideways to the other hand and needs the tip of the thumb to touch the tip of the index finger. The wrist is free to rotate about the $\vec{z}$ axis. Table 4 lists the parameters and their threshold ranges. We do not use the performing time to determine if a pinch gesture is detected or not; instead, we check the distance between the tip of the thumb and the tip of the index finger, which is denoted as the parameter of *pinch distance*.

4.2.5 Drop-in and trash-out

Drop-in and trash-out are a pair of gestures used to give commands like "put it in" and "throw it away". They are also used to indicate the direction in which a person intends to move an object. Both gestures usually combine with a pinch gesture. For example, while performing a pinch gesture, a drop-in gesture changes the palm orientation from facing sideway to facing down, and a trash-out gesture changes it from facing sideway to facing up.

Both drop-in and trash-out gestures are dynamic gestures. The start stage involves detecting a pinch gesture. The end stage involves evaluating the palm directional angle to check if it meets the requirements of either drop-in or trash-out gestures. In the update stage, the pinch gesture should remain active, and the value of palm directional angle is updated. In the end stage, if the value corresponds to the status of the palm facing down, a drop-in gesture is detected; or if it corresponds to the status of the palm facing up, a trash-out gesture is detected. Table 5 shows the threshold ranges of the palm directional angle. The pinch gesture deactivates after a drop-in or trash-out gesture is detected.

4.2.6 Pointing

Pointing forward gesture as the name suggests is used to point at something that is in front of a person. Generally, in real life, we use pointing forward gesture to point at something. Similarly, in virtual environments, pointing gesture can be used to point at an object like we use a mouse pointer.

Pointing forward is a static gesture. The pointing forward gesture is characterized by hand forming a fist but index finger extended and pointing forward. We use the *Finger Bend Angle* as the angle between the vectors $\overrightarrow{a_1a_2}$ and $\overrightarrow{a_2a_3}$, where a can be any finger in the set $\{t, m, r, p\}$. The fingers would be involved in an ideal fist if the Finger Bend Angle is 60 degrees. We check the Finger Bend Angle for all the bent fingers to be in the threshold range as shown in the Table 6. We also check the Finger Bend Angle (for the extended fingers) of the index finger. This ideally should be 180 degrees. The threshold ranges for these parameters is as shown in Table 6. In addition to this, we check the orientation of the index finger for pointing forward by checking the angle between $\overrightarrow{i_2i_3}$ and the *z-axis* which is called the *Pointing Orientation Angle*. Ideally, this angle should be 0 degrees. The threshold range for this is also shown in Table 6.

5 Experimental design

We examined the usability and reliability of our gestures with two experimental applications: one is a gesture-based video game (Section 5.1), and the other one is a gesture-based 3D interface for image categorization in a virtual environment (Section 5.2).

5.1 "Happy Ball" video game

Video games present emerging phenomena, which motivate users to perform predefined gestures to interact with gaming content. Integrating hand gestures with the gameplay mechanics provides meaningful repetition of same gestures and promotes user's vigilance and interest.

For our experiment, we developed a video game called "Happy Ball" including three mini games. The player character is a ball, and the gameplay goal is to keep the ball happy. The happiness level (life status) of the ball is represented with nine facial expressions textured on the surface of the ball. The happiness levels in decreasing order are elated, joyful, happy, satisfied, neutral, unhappy, sad, depressed, and crying. The three mini games depict the seasons of Winter, Summer and Fall. In accordance with the gameplay mechanics in each mini game, the user performs only a subset of gestures from the gesture vocabulary to control the ball's movement. The mappings between gestures and ball controls are listed as follows:

(1) In the winter game, a user can perform a swipe gesture to move the ball left and right.
(2) In the winter game, a user can perform a stop gesture so that the ball stops bouncing.
(3) In the summer game, a user can perform a stop gesture so that the ball stops moving.
(4) In the summer game, a user can perform a come gesture to make the ball move forward.
(5) In the summer game, a user can perform a go gesture to make the ball move backward.
(6) In the fall game, a user can perform a drop-in gesture to drop objects on the ball.
(7) In the fall game, a user can perform a trash-out gesture to throw away objects.

Figure 3 shows the screen shots of the mini games. The game restarts if the happiness level goes down to crying. The game does not provide a way to increase the happiness level. We created a *configuration file* for each mini game to determine the objects and where to put them in the gaming scene. We did not use a random strategy to propagate game objects because we want to ensure the same difficulty level for all user-study participants. In the following subsections, we present the design concepts and the gestures used in each mini game.

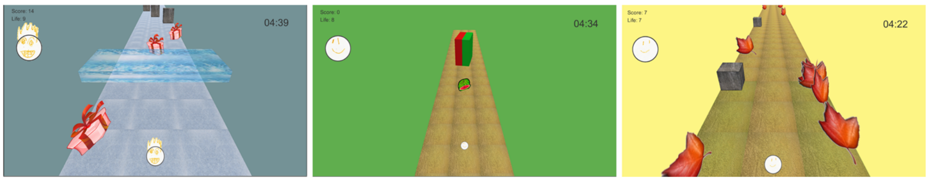


Fig. 3 The screen shots of the three "Happy Ball" mini games. From the left to right, they are winter game, summer game, and fall game

5.1.1 Winter game

The ball bounces and self-propels on a three-lane path. The gameplay goal is to command the ball to move left and right using the swipe gestures to collect gifts (rewards) and avoid obstacles on the path (walls) and perform a stop gesture to avoid the obstacles above the path (floating ice blocks). Being hit by a wall or an ice block will decrease the happiness level by one, and collecting a gift will increase the level by one.

5.1.2 Summer game

This is similar to the "Red Light - Green Light" children's game. A rewarding object (watermelon) is placed in front of a flippable board (like a traffic light). The gameplay goal is to move the ball forward using the go gesture to get the watermelon and bring it back to the starting line using the come gesture. The board spins and then stops at the red. When the board is spinning, the ball can be in motion; otherwise, at the red, the player should stop the ball using the stop gesture to avoid the decrease of the happiness level. Every watermelon brought back increases the score by one. Every time the player fails to stop the ball on red-light results in loss of a happiness level.

5.1.3 Fall game

This mini game has a path with three lanes. The ball stays on the middle lane and moves forward at a constant speed. It cannot be moved to either of the side lanes. Leaves (rewards) and stones (punishments) appear on the left and right lanes. While the ball is moving, the player has to command the ball to pick up leaves using the pinch gesture and drop them on itself using the drop-in gesture, and pick up stones using the pinch gesture and throw them away using the trash-out gesture. Every dropped-in leaf increases the score by one. Every missed or dropped-in stone decreases the happiness level by one.

5.2 Image categorization interface

Modern technologies with cameras, phones, scanners, satellites, and surveillance systems can produce a massive amount of digital images. Categorizing those images can be a challenging task due to extreme complexity of the contents in unsorted image collections. In a situation where a large collection of images must be manually sorted or categorized, traditional user interfaces have limitations: (1) limited display space for browsing or retrieval tasks, (2) deep and complex menus for sorting and navigating through the collection and (3) lack of technical support to take inputs from natural human manipulation actions. Performing hand gestures for digital image manipulation was suggested decades ago by Hauptmann et al. [14] in order to improve the naturalness and engagement in interaction.

For our experiment, we developed a 3D interface for users to categorize images with his or her hands in virtual reality. The interface is displayed with a VR device, which provides extra levels of immersion. Images are organized in a curved virtual display wall. The wall can be extended infinitely and rotated with respect to the eye position. We call this visualization an "Image Wall". Our design focuses on a set of interaction functions between hand gestures and the interface. The user can use one gesture at a time to control the interface or manipulate images as they progress. Those interaction functions are listed as follows:

(1) A user can perform a swipe gesture to flip through images.

(2) A user can point on the interface to highlight individual images.
(3) A user can perform a pinch gesture to select the highlighted image.
(4) A user can perform a come gesture to move the image towards the user (zoom-in).
(5) A user can perform a go gesture to move the image back into the interface (zoom-out).
(6) When an image is selected, a user can perform a swipe gesture to put the image into a category on the left or on the right of the interface.

6 Evaluation

We conducted a user study to assess the execution efficiency, accuracy, and reliability of the hand gestures described in Section 4. The user study also assessed the usability and acceptance of the hand gestures. The evaluation of participants' experience was conducted with a *usability questionnaire* based on the System Usability Scale (SUS) [5] and an *acceptance questionnaire* based on Technology Acceptance Model (TAM) [8]. The usability and acceptance questionnaires were administered to each participant after each of these sessions of the user study. The study was video recorded only for the purpose of evaluation.

6.1 Apparatus

We used a Perception Neuron motion capture suit. It is wireless and with the capability for a customizable configuration. For our study, only the gloves and torso straps were used to track the movements of upper body, arm and hand joints, in accordance with the illustration in Fig. 1. There are a total of 23 motion sensors, including 9 for each hand, 1 for each arm, and 3 for the torso. Each sensor is able to transmit motion data wirelessly to the AXIS Neuron PRO software system. Note that our approach is independent from Perception Neuron products. The input of our approach only requires an articulated 3D hand model with joint rotations relative to the skeletal hierarchy. We used a Perception Neuron suit for this study, because the suit was available for us, and it has been integrated with the game engine in our lab. Our approach can work seamlessly with camera-based systems such as Leap Motion [10], or sensor-based systems such as CyberGlove [17], or any other types of motion tracking devices that can produce such joint rotations. We used the Unity game engine to implement the “Happy Ball” games, the image categorization tool, and the angular-velocity method of gesture recognition. Motion data was streamed from the AXIS Neuron PRO software to the Unity in real-time. An Oculus Rift was used to present the image categorization tool in the immersive virtual environment.

6.2 Procedure

Prior to the start of the study, each participant was asked to complete a demographic questionnaire about his or her previous gaming experience. They were then instructed to wear the motion-tracking gloves on both hands and the torso straps. Participants were given a brief verbal instruction on how to perform the gestures. They were also informed that a video recording would be collected during the user study.

The user study is divided into three sessions. The first session is to evaluate the accuracy of our gesture recognition approach. The accuracy is defined as the percentage of correctly recognized gestures over the total requested gestures. Participants were not familiar with the angular-velocity techniques or parameter thresholds. We created an interface tool that contains the implementation of the gesture recognition approach. The tool displays one gesture name at a time, and then a progress bar runs from one end to the other end of the screen horizontally in a span of three seconds. During this time span, the participant needs to finish performing the gesture. Then, the tool displays a color bar in either green, red, or yellow. The green color indicates the correct recognition of the performed gesture; the red color indicates the performed gesture is recognized as a different gesture in the gesture vocabulary; and the yellow color indicates either no gesture is performed or it is unknown in the vocabulary. To evaluate the accuracy, we requested participants to perform each gesture 20 times, so we obtained a total of 240 samples for each gesture. Thus, the accuracy is calculated by dividing the number of gestures performed correctly (by all participants) by the total 240 samples.

In the second session, participants played three “Happy Ball” mini games, as shown in Fig. 4. At the beginning of each mini game, each participant was given a verbal instruction on how to play the game. Then, the participant played the mini game for two minutes to practice and then played five minutes for our experiment. After the participant finished playing all three mini games, he or she answered the usability and acceptance questionnaires.

In the third session, participants completed a simple image categorization using with the VR interface, as shown in Fig. 5. At the beginning of this session, each participant was given a verbal instruction on how to use the VR interface. Then, the participant practiced the use of the interface for two minutes. After that, the participant was given five minutes to classify the images presented in the interface, followed by answering the usability and acceptance questionnaires. When all three sessions were completed, a short interview was given to obtain feedback about the study.

6.3 Participants

Participation in this study was voluntary. It took about two hours for a participant to complete this study. A total of twelve participants participated this study (all adults, mean age is 23.83 years). Among all participants, eight of them had never played a gesture-based video game, two participants had experience using a Microsoft Kinect, and two participants had experience using Wii controllers. Ten out of all participants stated that they are no skills on playing gesture-based games. All participants were motivated to participate this study, and all of them expressed that they want to play gesture-based games.

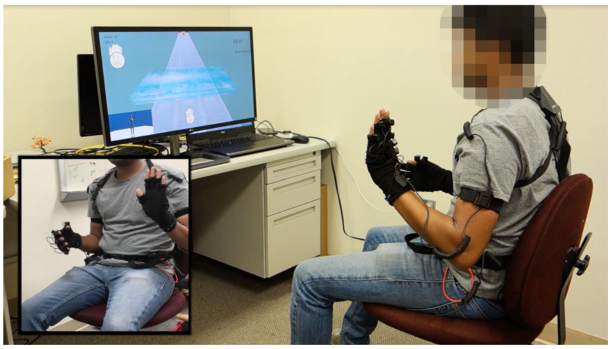


Fig. 4 A user study image showing that the user performs the stop gesture in the winter game to make the ball slide under the ice block

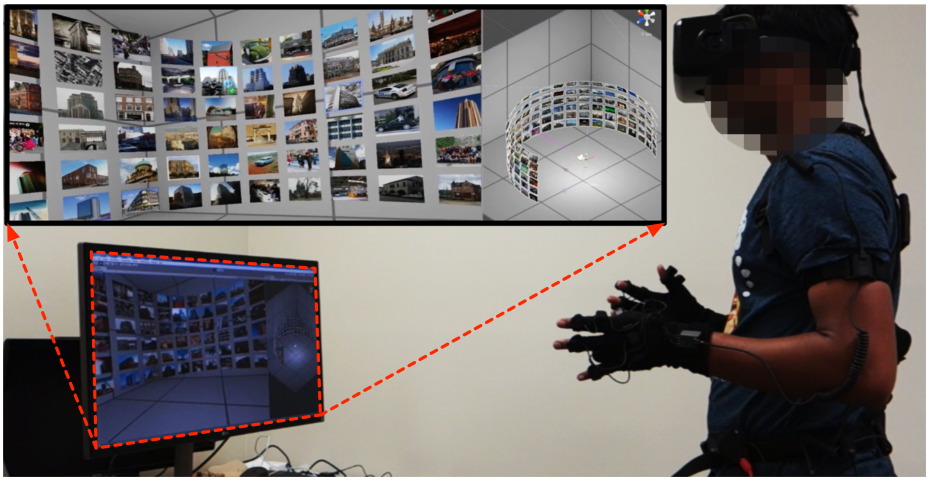


Fig. 5 A user study image showing that the user uses gestures to categorize images in the immersive VR environment

6.4 Results

6.4.1 Execution efficiency

We ran the experimental applications on a gaming laptop installed with the Windows operating system and a NVIDIA graphics card. The frame refresh rates of the applications reached 60 frames per second. This means the runtime settings in each application are updated in 16.67 milliseconds, which is the sum of the times spent on the gesture recognition, software logic updates, and graphics rendering. This indicates the execution of our angular-velocity method for gesture recognition is efficient, and it does not slow down the overall performance of the applications. At the execution time to update a frame, our method recognizes the input gesture by checking if the corresponding thresholding criteria are satisfied. The operations to recognize a gesture are 3D vector dot products, trigonometric functions, and additions for timing variables, which are simple algebraic operations and return numerical results instantly. The high execution efficiency suggests a very low latency of gesture recognition and make the gestures suitable to be incorporated in interactive systems.

6.4.2 Recognition accuracy

The recognition accuracy was measured using the gesture data obtained from the first session of the user study, as described in Section 6.2. The accuracy results are shown in Fig. 6. The overall mean accuracy is 97.3%, which takes both static and dynamic gestures into account. Two data points were excluded from the accuracy analysis. A participant performed a gesture but it was not the requested gesture. We observed this case from the video replays and determined that the participant misread the instructional words on the screen. Thus, we excluded the data associating to this case from the accuracy measurement. The other case was when another participant performed a go gesture, the hardware malfunctioned. Some joint angles returned by the motion capture suit were not correct, so the captured motion data could not imitate the movement performed by the participant's hand. We excluded the go gesture of that participant from the accuracy evaluation.

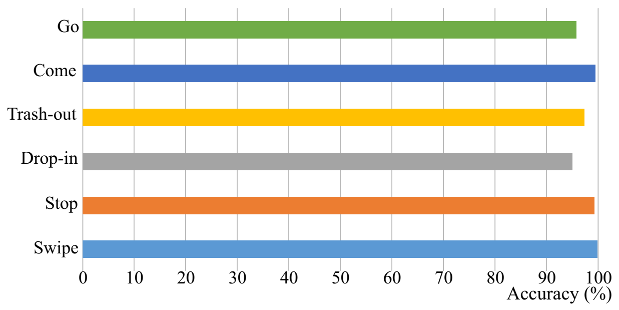


Fig. 6 Accuracy scores of six gestures in the gesture vocabulary

The drop-in gesture accuracy is slightly lower than other gestures. By observing the video replays, we found that some participants initiated their hand postures for a drop-in gesture with the palm facing upward. Such an instance of incorrect initializations occurred unintentionally by the participants when they wanted to perform the drop-in gesture followed after finishing a trash-out gesture. This is a typical cocking situation that also appears in daily gestural actions, e.g., throwing by moving the arm forward requires the arm to be first brought backwards. Threshold values of the parameters help reduce the occurrence of cocking situations, so that we achieved a reliable recognition result with 95%-100% accuracy.

The accuracy evaluation described in this subsection was conducted in a controlled environment, which means participants had to perform the gestures in the order that the system required. Real tasks within the applications were excluded from this evaluation, because in real tasks, the accuracy of gesture recognition can be mainly affected by the rate of the user's reaction to an event of the application rather than the recognition method itself. For example, in our previous study [30], the recognition accuracy in the "Happy Ball" game was 82.6%. This was due to wrong decisions on the gameplay, and consequently the participants performed wrong types of gesture or completely missed (doing nothing) when a gesture was required by the game. Thus, in this work, we did not evaluate the accuracy of the gesture recognition method in real tasks.

6.4.3 Usability and acceptance

We developed the usability questionnaire based on SUS [5] to measure the perceived usability of the systems. The usability questionnaire consists of ten questions seeking a five-point Likert scale response (5: strongly agree and 1: strongly disagree). The questions of the usability questionnaire are listed in Table 7. The SUS scores from the participants are listed in Table 8. The scores have been converted to an equivalent number out of 100 using the formula:

$$SUSscore = (\sum_{i=1}^{10} Q_i) \times 2.5, \text{ where } Q_i = \begin{cases} R_i - 1, & \text{if } i \text{ is odd} \\ 5 - R_i, & \text{if } i \text{ is even} \end{cases} \tag{2}$$

Since the questionnaire includes negative statement, in (2), for all odd-numbered question we have $Q_i = R_i - 1$, and for all even-numbered question we have $Q_i = 5 - R_i$, where $1 \leq i \leq 10$. Here, Q_i is the SUS score for the rating R_i obtained for the ith question. This score directly accounts for the usability of the system. The mean SUS score of an evaluation can be mapped to adjectives, which are "Worst Imaginable" (12.5),

Table 7 The usability questionnaire developed based on SUS

	Questions
1.	I think that I would like to use this system frequently.
2.	I found the system unnecessarily complex.
3.	I thought the system was easy to use.
4.	I think that I would need the support of a technical person to be able to use this system.
5.	I found the various functions in this system were well integrated.
6.	I thought there was too much inconsistency in this system.
7.	I would imagine that most people would learn to use this system very quickly.
8	I found the system very cumbersome to use.
9.	I felt very confident using the system.
10.	I needed to learn a lot of things before I could get going with this system.

"Awful" (20.3), "Poor" (35.7), "OK" (50.9), "Good" (71.4), "Excellent" (85.5) and "Best Imaginable" (90.9) [3]. The mean SUS score of 12 participants, for the gameplay was 90.83; and that for the image categorization interface was 91.04. Henceforth, high degrees of usability are exhibited in both experimental applications.

We developed the acceptance questionnaire to investigate the factors that influence the adoption of the gesture-based interaction technology. The development of the acceptance questionnaire was based on the TAM theory that is often used to understand how well users would accept and use new technologies. There are two major factors that impact the acceptance of technology: Perceived Ease Of Use (PEOU) and Perceived Usefulness (PU). PEOU refers to how easy the user thinks to use the system. PU refers to how useful the technology is [8]. Other factors such as Physical Engagement (PE) and Social Influence (SI) used in [34, 38] were not considered, because they do not apply to the goal of our study. Table 9

Table 8 Mean ratings in the usability questionnaire

Participant no.	Gameplay	Image categorization interface
P1	100	100
P2	95	100
P3	100	85
P4	90	87.5
P5	90	90
P6	95	77.5
P7	72.5	85
P8	90	100
P9	82.5	95
P10	87.5	85
P11	95	95
P12	92.5	92.5
Overall	90.83	91.04

Table 9 The acceptance questionnaire developed based on TAM

Factors	Questions
Perceived Ease of Use (PEOU)	1. I quickly understood how to perform the gestures.
	2. I found the gestures natural to perform.
	3. I found the gestures simple to perform.
	4. It requires no physical effort to perform the gestures or play the game/perform tasks in the system.
	5. It does not require a lot of cognitive effort to learn to perform the gestures or play the game/use the system.
	6. I could play the game/perform the tasks in the system with little or no effort.
Perceived Usefulness (PU)	7. It was fun to use hand gestures to play the game/ perform tasks in the system.
	8. It was immersive to use hand gestures to play the game/perform tasks in the system.
	9. I found the gameplay/image categorization task to be natural.
	10. I could easily play the game/perform the tasks in the system.
	11. I found the gameplay/actions in the system to be instantaneous.
	12. There was no delay in the desired action being performed on the screen after the gesture was performed.
	13. I would use hand gestures to play the game/interact with the system over keyboard-mouse and touch input.

shows the list of questions in the acceptance questionnaire. Each question requests a five-point Likert scale response from the participant (5: strongly agree and 1: strongly disagree). Table 10 shows the descriptive statistics of the two factors in the gameplay and image categorization tasks. The means of TAM scores for the gameplay and image categorization task are above 4.33 and above 4.25, respectively. This suggests that gestures in the experimental applications were found to be easy-to-use and useful for interactive tasks.

The first six questions in the acceptance questionnaire correspond to the usability of the system, and they received high ratings for the gameplay and image categorization tasks. This suggests that our method is capable of being used in different types of interactions. For example, the swipe gesture used to move the ball in the games has also been used to spin the virtual image wall in the image categorization interface. As a broader impact, the swipe gesture could be extended to be used for switching tabs or videos in a web browser. The largest difference between the mean values was found for the eighth question, where the difference is 0.50. This suggests that using gestures to perform image categorization tasks was more favorable to the participants than using the gestures to play the games.

We also found that the use of the VR display increased the rating score for immersion-related evaluation. For example, the image categorization interface received the full score

Table 10 Responses of the participants in the acceptance questionnaire

Question no.	"Happy Ball" game		Image categorization	
	Mean	Std. Dev.	Mean	Std. Dev.
Q_1	4.92	0.29	4.92	0.29
Q_2	4.92	0.29	4.83	0.39
Q_3	4.83	0.39	4.92	0.29
Q_4	4.75	0.45	4.83	0.58
Q_5	4.75	0.45	4.58	1.16
Q_6	4.58	0.67	4.83	0.39
Q_7	4.92	0.29	5.00	0.00
Q_8	4.50	0.67	5.00	0.00
Q_9	4.58	0.67	4.92	0.29
Q_{10}	4.50	0.67	4.83	0.39
Q_{11}	4.58	0.67	4.83	0.39
Q_{12}	4.33	0.98	4.67	0.49
Q_{13}	4.41	0.67	4.25	0.75

for the eighth question in the acceptance questionnaire. Participants expressed that they felt they were being situated in the virtual environment when wearing the VR display device.

A one-way within-subjects ANOVA analysis [7] ($p > 0.05$) was conducted to examine any potential TAM differences between the gameplay task and the image categorization task. According to the result ($F(1, 22) = 1.60, p = 0.22$) there was no significant difference between tasks.

Participants also answered the post-study interview questions about the naturalness of using the gesture-based applications. All of the participants agreed that the come, stop, drop-in, trash-out, pointing and swipe gestures were natural gestures. 83.3% of the participants agreed that the go gesture was natural to use. Participants were asked to rate the reliability of gestures on a scale of 1 (unreliable) to 10 (reliable). Performing gestures for the gameplay task received an average rating of 9.17. Performing gestures for the image categorization task received an average rating of 9.33. Participants were also asked to rate the engaging effectiveness of the gestures on a scale of 1 (least engaging) to 10 (Most engaging). The mean ratings were 9.29 and 9.33 for the gameplay and image categorization tasks, respectively.

6.5 Limitations

The meanings of the hand gestures used in this work are easy to interpret, neutral, and straightforward in the United States. However, we did not consider the impact of worldwide cultures when designing the gesture vocabulary. It is possible that a gesture presenting a neutral or positive meaning in one country is an offensive or insulting sign in another country. For example, the come gesture used in this work is commonly used in the United States to ask somebody to step forward, but it could be an undesirable gesture in some asian countries. People coming from different geographical regions or growing up with different cultural or religion backgrounds may move their hands very differently for the same gesture, which could affect the recognition accuracy of our approach. Although considering the impact of culture is not related to the contributions of this work, it would be worth to add cultural considerations in the design of gesture-based interfaces and interactive applications.

In our user study, all participants agreed to wear the Perception Neuron motion capture suit, but there could be considerations on intrusiveness of the apparatus. Each participant had to wear the gloves and torso straps during the two-hour user study. Although the suit is safe and lightweight, some participants hesitated due to the consideration on comfort. They wanted to know more about the suit's characteristics before wearing it. We felt that the perception neuron suit may not be the most comfortable equipment to use in this study. It took a significant amount of time to put on, calibrate, and test. The wires connecting the sensors to the hub device could get tangled onto the participant's body and arms.

The gesture vocabulary designed in this work contains in-air gestures, which are primarily suitable for touchless interactions. Those gestures can be used in the applications that prefer interactions at a distance without a physical touch on a tangible device; however, the gesture vocabulary may not be very suitable in the applications that require touch-based gestures, in which contact points, finger moving directions, and touch strokes should be detected and recognized in real-time.

7 Conclusion and future work

In this paper, we proposed a novel angular-velocity method to recognize both static and dynamic hand gestures in real- time. We developed an adaptive hand model composed of the articulated finger joints that possess high degrees of independence. By using the adaptive hand model, we developed a vocabulary of hand gestures. Our gesture recognition method was found to have an average accuracy of 97.3%. In the user study, the gestures were used as the input modality in three mini games and a 3D image categorization interface. The result shows that the speed of gesture recognition is fast enough to support the use of highly interactive applications. We also found that the participants were satisfied with the experience of using hand gestures. They felt the gesture-based interactions were natural, useful, and easy-to-use.

In the future, we would like to invite people across a wide range of ages to participate the study and evaluate the accuracy based on the input from them all. The parameter threshold values were obtained empirically in this work. In the future, we plan to incorporate an automatic way to determine the threshold values of parameters and allow users to customize their personal preferences. We also plan to develop a hybrid approach that integrates our angular-velocity method with artificial neural networks for a broader gesture-based applications. The responses for perceived usefulness in the acceptance questionnaire could also be attributed to the design of user interfaces and the cognitive styles of users, which are the user-centered topic that we would like to study in the future. Also, we would like to evaluate the efficiency of our approach in recognizing gestures in sign languages as well as the possibility of using our approach to assist deaf people. Our approach may help deaf people by translating essential and sensitive words automatically.

Acknowledgements This work was supported by DOD grant W911NF-16-2-0016. We thank anonymous reviewers for their comments. We thank the people who participated in the user study.

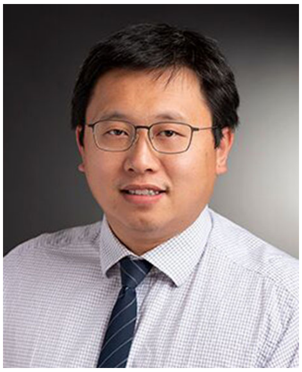
References

1. Aigner R, Wigdor D, Benko H, Haller M, Lindbauer D, Ion A, Zhao S, Koh JTKV (2012) Understanding mid-air hand gestures: a study of human preferences in usage of gesture types for hci. Tech. rep.

https://www.microsoft.com/en-us/research/publication/understanding-mid-air-hand-gestures-a-study-of-human-preferences-in-usage-of-gesture-types-for-hci/
2. Alavi S, Arsenault D, Whitehead A (2016) Quaternion-based gesture recognition using wireless wearable motion capture sensors. Sensors 16(5):605. https://doi.org/10.3390/s16050605
3. Bangor A, Kortum P, Miller J (2009) Determining what individual sus scores mean: adding an adjective rating scale. Journal of Usability Studies 4(3):114–123. https://uxpajournal.org/determining-what-individual-sus-scores-mean-adding-an-adjective-rating-scale
4. Brodie M, Walmsley A, Page W (2008) Fusion motion capture: a prototype system using inertial measurement units and gps for the biomechanical analysis of ski racing. Sports Technol 1(1):17–28. https://doi.org/10.1002/jst.6
5. Brooke J et al (1996) Sus-a quick and dirty usability scale. Usability Evaluation in Industry 189(194):4–7
6. Cassell J (1998) A framework for gesture generation and interpretation. Computer Vision in Human-Machine Interaction, pp 191–215
7. Cohen Y, Cohen JY (2008) Analysis of variance, in statistics and data with R: an applied approach through examples. Wiley, New York. https://doi.org/10.1002/9780470721896
8. Davis FD (1989) Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly: 319–340
9. Diliberti N, Peng C, Kaufman C, Dong Y, Hansberger JT (2019) Real-time gesture recognition using 3d sensory data and a light convolutional neural network. In: Proceedings of the 27th ACM international conference on multimedia, MM '19. ACM, New York, pp 401–410. https://doi.org/10.1145/3343031.3350958
10. Guna J, Jakus G, Pogačnik M, Tomažič S, Sodnik J (2014) An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking. Sensors 14(2):3702. https://doi.org/10.3390/s140203702
11. Häger-Ross CK, Schieber MH (2000) Quantifying the independence of human finger movements: comparisons of digits, hands, and movement frequencies. J Neurosci Official J Society Neurosci 20(22): 8542–50
12. Hansberger JT, Peng C, Blakely V, Meacham S, Cao L, Diliberti N (2019) A multimodal interface for virtual information environments. In: Chen JY, Fragomeni G (eds) Virtual, augmented and mixed reality. Multimodal Interaction. Springer, Cham, pp 59–70
13. Hansberger JT, Peng C, Mathis SL, Areyur Shanthakumar V, Meacham SC, Cao L, Blakely VR (2017) Dispelling the gorilla arm syndrome: the viability of prolonged gesture interactions. Springer, Cham, pp 505–520. https://doi.org/10.1007/978-3-319-57987-0_41
14. Hauptmann AG (1989) Speech and gestures for graphic image manipulation. SIGCHI Bull 20(SI):241–245. https://doi.org/10.1145/67450.67496
15. Hummels C, Stappers PJ (1998) Meaningful gestures for human computer interaction: beyond hand postures. In: Third IEEE international conference on automatic face and gesture recognition, 1998. Proceedings, pp 591–596. https://doi.org/10.1109/AFGR.1998.671012
16. Hutchins EL, Hollan JD, Norman DA (1985) Direct manipulation interfaces. Hum-Comput Interact 1(4):311–338. https://doi.org/10.1207/s15327051hci0104_2
17. Kessler GD, Hodges LF, Walker N (1995) Evaluation of the cyberglove as a whole-hand input device. ACM Trans Comput-Hum Interact 2(4):263–283. https://doi.org/10.1145/212430.212431
18. Kieras D, Meyer D, Ballas J (2001) Towards demystification of direct manipulation: cognitive modeling charts the gulf of execution. In: Proceedings of the SIGCHI conference on human factors in computing systems, CHI '01. ACM, New York, pp 128–135. https://doi.org/10.1145/365024.365069
19. Lang CE, Schieber MH (2004) Human finger independence: limitations due to passive mechanical coupling versus active neuromuscular control. J Neurophys 92(5):2802–2810. https://doi.org/10.1152/jn.00480.2004
20. Lee J, Kunii TL (1995) Model-based analysis of hand posture. IEEE Comput Graphics Appl 15(5):77–86
21. Lin J, Wu Y, Huang TS (2000) Modeling the constraints of human hand motion. In: Proceedings of workshop on human motion. IEEE, Austin, Texas, USA, pp 121–126. https://doi.org/10.1109/HUMO.2000.897381
22. Liu K, Kehtarnavaz N (2016) Real-time robust vision-based hand gesture recognition using stereo images. J Real-Time Image Process 11(1):201–209. https://doi.org/10.1007/s11554-013-0333-6
23. Lu Z, Chen X, Li Q, Zhang X, Zhou P (2014) A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices. IEEE Trans Human Mach Sys 44(2):293–299. https://doi.org/10.1109/THMS.2014.2302794
24. Luzhnica G, Simon J, Lex E, Pammer V (2016) A sliding window approach to natural hand gesture recognition using a custom data glove. In: 2016 IEEE symposium on 3D user interfaces (3DUI). IEEE, pp 81–90

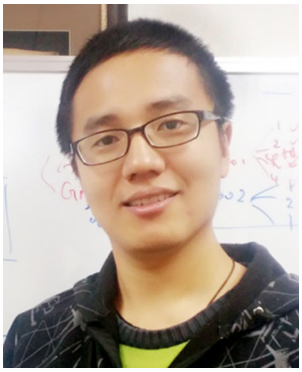
25. Marin G, Dominio F, Zanuttigh P (2016) Hand gesture recognition with jointly calibrated leap motion and depth sensor. Multimed Tools Appl 75(22):14991–15015. https://doi.org/10.1007/s11042-015-2451-6
26. Morris MR, Wobbrock JO, Wilson AD (2010) Understanding users' preferences for surface gestures. In: Proceedings of graphics interface 2010, GI '10. Canadian Information Processing Society, Ottawa, Ontario, Canada, pp 261–268. http://dl.acm.org/citation.cfm?id=1839214.1839260
27. Neto P, Pereira D, Pires JN, Moreira AP (2013) Real-time and continuous hand gesture spotting: an approach based on artificial neural networks. In: 2013 IEEE international conference on robotics and automation, pp 178–183. https://doi.org/10.1109/ICRA.2013.6630573
28. Nielsen M, Störring M, Moeslund TB, Granum E (2003) A procedure for developing intuitive and ergonomic gesture interfaces for hci. In: International gesture workshop. Springer, pp 409–420
29. Pavlovic VI, Sharma R, Huang TS (1997) Visual interpretation of hand gestures for human-computer interaction: a review. IEEE Trans Pattern Anal Mach Intell 19(7):677–695. https://doi.org/10.1109/34.598226
30. Peng C, Hansberger J, Shanthakumar VA, Meacham S, Blakley V, Cao L (2018) A case study of user experience on hand-gesture video games. In: 2018 IEEE games, entertainment, media conference (GEM), pp 453–457. https://doi.org/10.1109/GEM.2018.8516520
31. Peng C, Hansberger JT, Cao L, Shanthakumar VA (2017) Hand gesture controls for image categorization in immersive virtual environments. In: 2017 IEEE virtual reality (VR), pp 331–332. https://doi.org/10.1109/VR.2017.7892311
32. Ramamoorthy A, Vaswani N, Chaudhury S, Banerjee S (2003) Recognition of dynamic hand gestures. Pattern Recogn 36(9):2069–2081. https://doi.org/10.1016/S0031-3203(03)00042-6
33. Rautaray SS, Agrawal A (2015) Vision based hand gesture recognition for human computer interaction: a survey. Artificial Intelligence Review 43(1):1–54. https://doi.org/10.1007/s10462-012-9356-9
34. Rice M, Wan M, Foo MH, Ng J, Wai Z, Kwok J, Lee S, Teo L (2011) Evaluating gesture-based games with older adults on a large screen display. In: Proceedings of the 2011 ACM SIGGRAPH symposium on video games. ACM, pp 17–24
35. Sharma RP, Verma GK (2015) Human computer interaction using hand gesture. Procedia Comput Sci 54:721–727. https://doi.org/10.1016/j.procs.2015.06.085
36. Siek KA, Rogers Y, Connelly KH (2005) Fat finger worries: how older and younger users physically interact with pdas. In: IFIP conference on human-computer interaction. Springer, pp 267–280
37. Song Y, Demirdjian D, Davis R (2012) Continuous body and hand gesture recognition for natural human-computer interaction. ACM Trans Interact Intell Syst 2(1):5:1–5:28. https://doi.org/10.1145/2133366.2133371
38. Venkatesh V, Davis FD (2000) A theoretical extension of the technology acceptance model: four longitudinal field studies. Management Sci 46(2):186–204
39. Vogel D, Casiez G (2012) Hand occlusion on a multi-touch tabletop. In: Proceedings of the SIGCHI conference on human factors in computing systems, CHI '12. ACM, New York, pp 2307–2316. https://doi.org/10.1145/2207676.2208390
40. Wachs JP, Kölsch M, Stern H, Edan Y (2011) Vision-based hand-gesture applications. Commun ACM 54(2):60–71. https://doi.org/10.1145/1897816.1897838
41. Wang C, Liu Z, Chan SC (2015) Superpixel-based hand gesture recognition with kinect depth camera. IEEE Trans Multimed 17(1):29–39. https://doi.org/10.1109/TMM.2014.2374357
42. Wobbrock JO, Morris MR, Wilson AD (2009) User-defined gestures for surface computing. In: Proceedings of the SIGCHI conference on human factors in computing systems, CHI '09. ACM, New York, pp 1083–1092. https://doi.org/10.1145/1518701.1518866
43. Xu D (2006) A neural network approach for hand gesture recognition in virtual reality driving training system of spg. In: 18th international conference on pattern recognition (ICPR'06), vol 3, pp 519–522. https://doi.org/10.1109/ICPR.2006.109

Vaidyanath Areyur Shanthakumar is a Ph.D. student in the Department of Computer Science at University of Alabama in Huntsville (UAH). He received his Master degree in Computer Science from UAH advised by Dr. Chao Peng. His research interests include machine learning, real-time gesture recognition, and human-computer interaction.



Chao Peng is an Assistant Professor in the School of Interactive Games and Media in Rochester Institute of Technology. He received his Ph.D. in Computer Science from Virginia Tech and M.F.A in Computer Art from University of Alaska Fairbanks. His primary research areas are game development, computer graphics, GPU computing, virtual reality, and 3D interaction.

Jeffrey Hansberger is a research psychologist for the Army Research Laboratory. He received his Ph.D. in Human Factors and Applied Cognition from George Mason University. Dr. Hansberger currently is assigned to the ARL Field Element at the Redstone Arsenal, AL where he supports applied research on Human-Computer Interaction, information environments in Virtual Reality, and multi-modal input user interfaces.



Lizhou Cao is a Ph.D. student in Rochester Institute of Technology advised by Dr. Chao Peng. He received his Master degree in Computer Science from University of Alabama in Huntsville. His research interests include game-based multimedia applications and computer graphics.

Sarah Meacham is a Research Associate in the Systems Management and Production (SMAP) center at University of Alabama in Huntsville. She received her Master degree in Experimental Psychology from the University of Alabama in Huntsville. Research interested include humancomputer interaction, virtual reality, and interface usability.

Victoria Blakely is a Research Associate in the Systems Management and Production (SMAP) Center at University of Alabama in Huntsville. Victoria Blakely has a M.A. in Experimental Psychology with a focus in Industrial-Organizational Psychology from the University of Alabama in Huntsville. Her research focus includes User Interface Research, User Experience, and Human-Computer Interaction with an applied focus.