IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 30, 2019, accepted November 13, 2019, date of publication November 15, 2019,
date of current version November 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2953798

A Novel Human-3DTV Interaction System Based on Free Hand Gestures and a Touch-Based Virtual Interface

SHUN ZHANG AND SHIZHOU ZHANG
School of Electronic and Information, Northwestern Polytechnical University, Xi'an 710072, China

Corresponding author: Shun Zhang (szhang@nwpu.edu.cn)

This work was supported in part by the Youth Program of National Natural Science Foundation of China under Grant 61703344, in part by the Fundamental Research Funds for the Central Universities of China under Grant 3102017OQD021, and in part by the Top International University Visiting Program for Outstanding Young scholars of Northwestern Polytechnical University.

ABSTRACT The input method based on free-hand gestures has gradually become a hot research direction in the field of human-computer interaction. Hand gestures like sign languages, however, demand quite a lot of knowledge and practice for interaction, and air writing methods require their users to hold the arm and hand in mid-air for a period of time. These methods limit the user experience and get severer when a large number of gestures are required. To address the problem, this paper presents a novel human-3DTV interaction system based on a set of simple free-hand gestures for direct-touch interaction with a virtual interface to facilitate human-3DTV interaction. Specifically, our system projects a virtual interface in front of the user who wears the 3D shutter glass, and the user just stretches the arm and touches the virtual interface like performing on a smart phone with a touch screen, using gestures such as Click, Slide, Hold, Drag and Zoom In/Out. Our system is able to recognize the user's gesture fast and accurately, as the system only needs to search for a small region neighboring the virtual interface for a small set of gesture types. Because we adopt the key gestures using on smart phones, our free-hand gestures can be easily used by anyone with only a brief training. The users feel more comfortable than traditional gesture input methods and can effectively interact with 3DTV using our system. We report a comprehensive user study on accuracy and speed to validate the advantages of the proposed human-3DTV interaction system.

INDEX TERMS Human–computer interaction, virtual interface, free hand gestures, 3D interaction.

I. INTRODUCTION

The fast growing technologies in digital Television (TV) has dramatically increased the richness of multimedia content. With the launch of smart television projects by many important companies including Google, Apple, Samsung, Xiaomi etc., modern TV set allows the user to surf Internet, install a wide variety of TV apps, play on-line games, upload/share videos from external sources, enjoy double-screen experiences and so on. In addition to have the above functions, 3D Television (3DTV) can significantly enhance the visual experience of viewers by employing techniques like an active shutter 3D system or a polarized 3D system, and it has attracted much attention in both the cinema and televisor industries. Although the users are now given various selections of 3DTV content, the human-3DTV interaction methods have been left behind. Most 3DTV sets still use the conventional interaction method, i.e. , a physical remote controller for users to input commands. For example, to enter the lengthy website link or WiFi password, the user of Xiaomi TV has to perform multiple operations with the physical remote controller which only contains **Left/Right/Up/Down/Ok** buttons. Consequently, given the diversity and complexity of the 3DTV contents and functions, the conventional human-3DTV interaction strategy strongly limit the user's experience.

To fulfil requirements of new Human-3DTV interaction, researchers have experimented various natural and human-centered approaches. Some interaction designs are the extensions of traditional remote controller, such as joystick [1] that pivots on a base and reports its angle or direction, laser-pointer style direct-pointing device [2], wireless keyboard [3] and touch-based interfaces on smart phones [4]. While these interaction paradigms are well-established, their

The associate editor coordinating the review of this manuscript and approving it for publication was Waleed Alsabhan.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

FIGURE 1. The proposed human-3DTV interaction system enables users sitting in front of the 3DTV to remotely control the television by directly touching the virtual interface with simple free-hand gestures.

drawbacks are obvious: (1) the physical device may be lost, broken or unavailable; (2) the user's visual attention would be distracted from the TV screen by these devices; and (3) there is a possibility to disseminate infectious diseases due to dirt, dust, and moisture getting into the spaces between buttons. Some other interaction methods are based on speech input [5] making use of the advantages of natural language. The recognition accuracy, however, would get worse if people are far from the TV and there exist ambient environment noise, and some speech commands like continuous scrolling channel list or adjusting the TV volume are not well suited for typical Human-3DTV interactions.

Compared to these above-mentioned methods, the input method based on free-hand gesture recognition [6]–[9] provides a very friendly, intuitive and free-hand way for users, and it has become a hot research direction in the field of human-computer interaction. Most conventional gesture recognition algorithms [10], [11] are based on RGB color images. Recently, with the development of inexpensive depth cameras like Microsoft Kinect [12], the availability of capturing the depth information along with the color information improves the gesture recognition performance [13], [14] and makes gesture recognition based on Kinect easier and popular in Human-computer interaction. Existing free-hand gesture recognition approaches, however, still suffers from several issues. Hand gestures like sign languages [11], [15] often require quite a lot of knowledge and practice for interaction, which results in restricting the number of their users. Air writing methods [8], [9] require their users to hold the arm and hand in mid-air, which makes people uncomfortable and tiring. These gesture-based paradigms limit the user experience and get severer when a large number of gestures are required.

In other side, touch screen technology [16], [17] has been widely used in a myriad of devices such as smart phones, supermarket checkouts, restaurant ordering systems, medical devices, and self serve kiosks. Touch screen interaction shows the advantages: (1) Because a user operates an electronic device by directly touching the images on the display he is seeing, the operation will be intuitive, thus anyone can operate it from first use. (2) There is a lot of flexibility in design because people can make and modify various input interfaces creatively by software. (3) Various operations/inputs are possible including single-touch gestures (e.g. swipe, flick, drag, and tap) and multi-touch function (e.g. zoom-in/zoom-out, rotation).

This has motivated us to present a new human-3DTV interaction system that combines the benefits of touch-based virtual interface and free-hand gesture control. Our proposed Human-3DTV system (as shown in Figure 1) projects a virtual interface in front of the user who wears the 3D shutter glasses. Only in the user's viewing angle, the virtual interface consisting of several elements/buttons is visible, and the user just need to stretch the arm and hand and then directly touch the virtual interface like performing on a smart phone with a touch screen. The user can perform a set of simple free-hand gestures including primitive gesture types, **Click**, **Hold** and **Slide**, and three ''Combo'' gestures, **Double-Click**, **Drag** and **Multi-touch Zoom**. Our system applies computer-vision based algorithms to recognize the user's gesture and then control the TV set. With the presented virtual interface, our system has three advantages over existing approaches. (1) because all the user need to do is directly touching the elements on the virtual interface that are visible in front of eyes, the user has the similar easy and intuitive experience with performing on touch screen devices. (2) As the virtual interface and its user elements are projected at known world coordinates, users' gestures have smaller variance on the amplitude and direction of motion than traditional gesture input methods. Besides, our system only needs to search for a small region neighboring the virtual interface for a small set of gesture types. Thus Our system is able to recognize the user's gesture fast and accurately in such controlled settings. (3) As there is a lot of flexibility in the design of various interfaces, the virtual interface combined with the small set of simple gestures can fulfil the requirement of the diversity and complexity of the 3DTV contents. We report a comprehensive user study on accuracy and speed to validate the advantages of the proposed human-3DTV interaction system with the comparison of traditional gesture interaction methods.

This work is an extension of our preliminary work published in [18]. Here, we present more details of the gesture recognition algorithm and present the gesture recognition method based on Hidden Markov Model (HMM) with the orientation features, conduct more extensive and comprehensive experiments including influence of the distance between users and 3DTV, influence of button sizes and spacing distance of the virtual interface, confusion matrix on the gesture recognition accuracy, visualization of gesture hip map and the questionnaire. Besides, we add an expanded literature review of the related human-computer interaction approaches.

The rest of the paper is organized as follows: Section II reviews the related works with this paper. In Section III-A we give an overview of our system implementation. In Section III-B and III-C we describe the interface module and the gesture recognition module. Section V introduces comprehensive experimental evaluation results. Last section draws the conclusion.

II. RELATED WORK

In this section, we are going to review three related human-computer interaction approaches: glove-based interaction method, free-hand gesture input for interaction, and interaction with virtual interface.

The data glove-based interaction method [19]–[22] is extensively studied at the beginning. This method first measures various types of information of the hand through the glove, and then uses the measured information for the modeling and recognition of the gesture. For example, Huang *et al.* [22] design and fabricate a textile glove by sewing reduced graphene oxide-coated fiber to monitor the motion of ten finger joints from one hand for gesture recognition. KITTY [19] is a glove-type input device that covers parts of the hand with electronic contacts. When touched by another contact, it can trigger a touch event with location information. Fang *et al.* [21] present a data glove based on inertial and magnetic measurement units for gesture capturing and recognition. The data glove is made up of three-axis gyroscopes, three-axis accelerometers and three-axis magnetometers. The data glove-based gesture recognition method has high recognition rate and high speed, but this method requires artificial wear data gloves and placement of position trackers. In addition, the purchase price of data gloves is high and does not apply to natural human-computer interaction.

Recently, more gesture input methods use a Webcam, Kinect, or Leap Motion sensor [6], [8], [11] to track the trajectory of hand in the air, and then apply computer vision algorithms for image analysis to recognize the gestures. More analysis about gesture control prototypes can be found in a technical report [23]. These methods does not require wearing any devices, which makes the human-computer interaction more natural. Besides, the sensors, especially the webcam, are not expensive, which makes the cost of the system lower. Google made a customized chip in their Soli [24] system that uses 60 GHz wireless signal with mm-level wavelength to track small movement of a hand/finger based on the fact that wireless signal changes as a hand/finger moves. Li *et al.* [7] present a dynamic gesture recognition method which defines ten types of static gestures and ten types of motion trajectories such as up, down, left, right, clockwise, inverse time, top left, bottom left, right up, and right down. The defined static gestures and gesture motion trajectories are combined to form a combined gesture. Some other researchers [25] take human poses/actions as inputs for interaction. For example, Ghojogh *et al.* [25] utilize the temporal position of skeletal joints obtained by Kinect sensor to recognize sequences of several pre-defined poses such as hands together front, left/right hand up, duck, kick, bend and sit. Borghi *et al.* [26] recognize dynamic body gestures based on Multiple Stream Discrete Hidden Markov Models and 3D skeleton joint data for explicit Human-Computer Interaction. These gesture input methods need a lot of knowledge and practise, which results in restricting the number of their users.

Free-hand gestural control methods of TV have been investigated for many years. An early prototype by Freeman and Weissman [27] employs a video camera to detect the location and posture of the viewer's hand and also to detect click-like actions for controlling a TV. This enables the user to move an on-screen pointer and control graphical elements. Dawar and Kehtarnavaz [28] present a real-time detection and recognition approach to identify actions of interest involved in the smart TV application from continuous action streams via simultaneous utilization of a depth camera and a wearable inertial sensor. Li *et al.* [29] propose the design principles of multichannel interaction fusion on Smart TV based on user experience research of far-field speech and mid-air gesture interaction. These works require the user to hold his arm and hand in mid-air for a certain period of time each time, which people find somewhat uncomfortable and tiring.

Human-computer interaction methods using virtual interface have been reported recently. Chan *et al.* [30] propose the called Virtual Panel that uses a Fresnel lens to display an intangible planar interface in front of user's eyes. HoloDesk [31] uses a mirror as beam splitter to reflect light towards the user from a LCD display mounted above the mirror. It allows the user to directly interact with spatially aligned 3D graphics using their hands. Benko et al. introduce MirageTable [32] where the user can interact with virtual objects spatially co-located with real objects on the tabletop. MirageTable provides an immersive experience where the only limitations are due to the use of uncommon stereo projector and curved screen. Mujibiya *et al.* [33] propose a touch typing enabled virtual keyboard system that uses depth sensing on arbitrary flat surface to robustly detect hand postures. Shin and Kim [34] present a character input system based on hand tapping gestures for Japanese hiragana and English characters. The system places a virtual keypad at the calculated position on the screen for character input if the user raises the hand with some fingers stretched. The virtual keypad spread in the air is determined according to the number of stretched fingers of hand. Lam *et al.* [35] investigate the effect of the visual feedback of interaction techniques in a virtual reality system for selecting and manipulating a virtual object using a virtual hand.

The proposed system applies the characteristic of 3DTV that conveys depth perception to the viewer and project a virtual interface in front of the user who wears the 3D shutter glasses. The user just need to stretches the arm and touches the virtual interface like performing on a smart phone with a touch screen, using a small set of simple gestures such as Click, Slide, Hold, Drag, Zoom In/Out and so on. Our system is able to recognize the user's gesture fast and accurately, as the system only needs to search for a small region neighboring the virtual interface for a small set of gesture types. Because we adopt the key gestures using on smart phones, our free-hand gestures can be easily used by anyone with only a brief description. The users feel more comfortable than traditional gesture input methods and can effectively interact with 3DTV using our system.

III. SYSTEM DESIGN

In this section, we present our proposed human-3DTV interaction paradigm. Firstly, we give a system overview of the presented paradigm. Then, we describe the detailed modules including the virtual interface display module and the finger gesture recognition module.

A. OVERVIEW

The proposed system consists of a 3DTV to convey depth perception to the viewer, 3D shutter glasses for the user to see stereoscopic 3D images, a Kinect sensor placed below or on top of the 3DTV for capturing depth and RGB information, and a laptop for processing data from Kinect and performing virtual interface display and gesture recognition computation. The laptop and the 3DTV set are connected with an HDMI cable and they display the same image.

Our control software consists of two major modules: the virtual interface display module and the finger gesture recognition module. Specifically, the virtual interface display module detects the user's wearing glasses and display the left and right images so that the user wearing 3D shutter glasses can see a virtual interface appearing at a defined location in front of eyes. The virtual interface contains different elements that are adaptive to the TV content. The finger gesture recognition module is to process both depth and RGB color data from the Kinect sensor and analyze how the viewer's finger interacts with the virtual interface and then deliver the recognized commands to control the 3DTV set. The complete setup of our prototype system is shown in Figure 2.

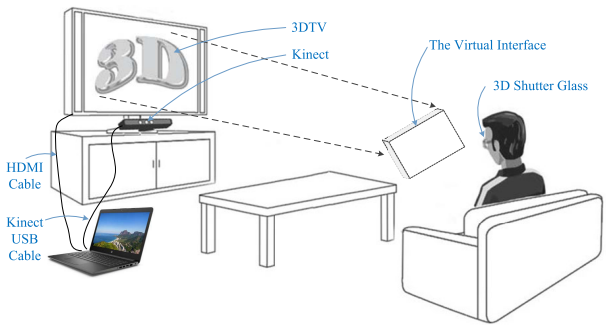


FIGURE 2. The proposed human-3DTV interaction system consisting of a 3DTV set to project stereoscopic virtual interface perceivable with 3D shutter glasses, and a Kinect sensor for user's finger gesture recognition.

Our system works as follows (Figure 3): When the system is turned on, the Kinect sensor first scans the entire area in front of the 3DTV to detect users. After the user is detected, the virtual interface display module starts to locate the user's eyes and then display a pre-designed virtual interface at a fixed location in front of the user. The viewer can interact with this virtual interface, such as click a button, drag an object, or Zoom In/Out a picture. During the interaction process, the Kinect sensor keeps monitoring the scene and capturing the 3D information of the user. The finger recognition module will recognize those gestures near to the region of virtual interface, and those recognized gestures are interpreted as commands to the 3DTV set. The following sections explain our system in details.

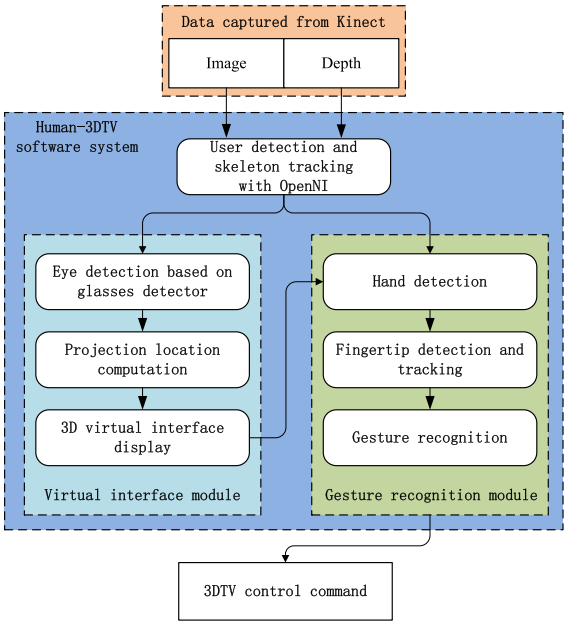


FIGURE 3. The working flow of the presented human-3DTV system.

B. VIRTUAL INTERFACE DISPLAY

The virtual interface display module aims to generate stereoscopic virtual interface with predefined items at specified 3D location in front of the viewer's eyes (in our implementation, we empirically set the distance d_1 between the virtual interface to the viewer's eyes as $d_1 = 0.5m$).

We first establish two coordinate systems: (1) the three-dimensional (3D) world coordinate system (WCS) with the origin O_w at the center of the Kinect's optical lens and the Z-axis facing the viewer; and (2) the 2D image coordinate system (ICS) with the origin O_{tv} at the top-left corner of the TV screen, as illustrated in Figure 4. The WCS for the top-left corner of the TV screen is denoted as $T_{tv} = [x_{tv}, y_{tv}, z_{tv}]^T$. WCS is expressed in millimeter, while ICS is expressed in pixel. With the two coordinate systems, in order to display the interface at correct 3D world location, next we estimate the 3D coordinate of the user's eye.

1) 3D EYE COORDINATE ESTIMATION

A simple way to estimate 3D coordinate of the user's eyes is to directly adopt the well known OpenNI [36] framework, which can track the skeleton joints of a body. The 3D coordinates of the user's eyes can be roughly estimated from the coordinate of the head joint. The method, however, is not robust or accurate since the skeleton tracking is not steady and may fail. In this paper, we propose to incorporate the concurrently complementary information of the RGB and depth data captured from the Kinect sensor to obtain more precise eye coordinate. Specifically, we first crop the head RGB image from the head joint, and then detect the user's 3D

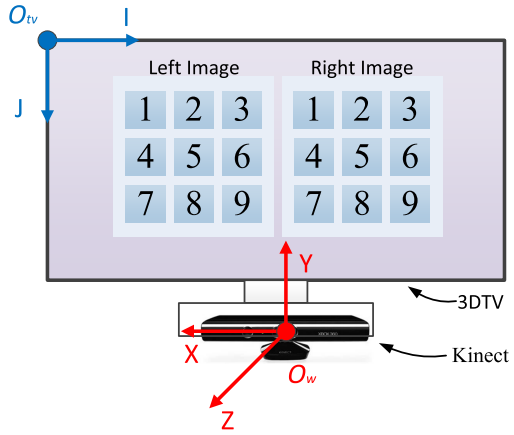


FIGURE 4. Illustration of world coordinate and image coordinate systems. Red: World Coordinate System (WCS); Blue: Image Coordinate System (ICS). For simplicity, we require the Z-axis in the WCS to be perpendicular to the TV screen.

shutter glasses in the cropped head Image. We correspond the glasses in the RGB image with the depth data in order to get the center of detected shutter glasses, and the center can be take as the midpoint of the line of two eyes. Glass detection in RGB images is a standard object detection problem. In our implementation, we adopt a cascaded AdaBoost detector [37] using Haar-like features to identify the position of the shutter glass object. As we detect glasses in a small head image instead of the whole image, our glass detection method have faster processing speed and higher accuracy.

Let $G = [x_g, y_g, z_g]^T$ be the center of detected shutter glasses. Assume that the two eyes have the same height, and are both within a plane parallel to the TV screen. The 3D coordinate of the left eye $E_l = [x_l, y_l, z_l]^T$ and right eye $E_r = [x_r, y_r, z_r]^T$ can be calculated in a simple form:

$$E = G + B, \tag{1}$$

where for the left eye, $E = E_l$ and $B = [b/2, 0, 0]^T$; for the right eye $E = E_r$, $B = [-b/2, 0, 0]^T$. The parameter b is the distance between two eyes and is empirically set to 65 millimeters.

2) 3D DISPLAY

Given the center G of detected shutter glasses, the center Q of the virtual interface can be $Q = G + [0, 0, -d_1]^T$. As shown in Figure 5, d_1 is the distance between the center Q of the virtual interface and the eyes, and $d_2 = z_g - 0.5m$ is the distance between Q and the TV plane. To display an item at given location $Q = [x_q, y_q, z_q]^T$, the points of two offset image in the TV screen can be drawn at the 3D location F_l and F_r as:

$$F = \frac{d_2}{d_1} \cdot (Q - E) + Q, \tag{2}$$

where for the left offset image, $F = F_l$ and $E = E_l$; and for the right offset image, $F = F_r$ and $E = E_r$. Finally, to convert the 3D location F into the image coordinate system for rendering on TV, the corresponding 2D location $[u, v]^T$ can be computed by:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = s_0 \cdot (R_{tv} \cdot F + T_{tv}), \tag{3}$$

where R_{tv} and T_{tv} are the rotation matrix and offset matrix between the image coordinate system and the world coordinate system, s_0 is a scaling factor which converts the millimeter unit in the world coordinate system to the pixel unit in the image coordinate system. For an 46 inch 3DTV set of 1280×720 pixels in resolution and $1018mm \times 572mm$ in screen dimension, $s_0 = 1280/1018 \approx 1.257$.

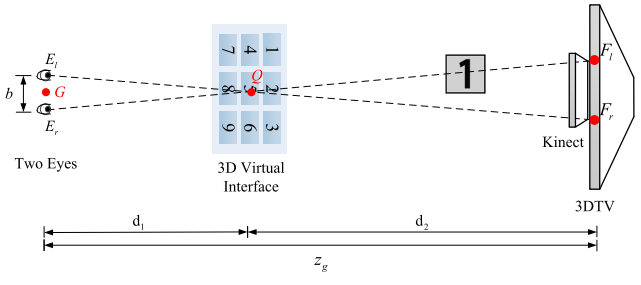


FIGURE 5. The pinhole camera model.

C. FINGER GESTURE RECOGNITION

The finger gesture recognition module aims to analyze the depth map captured by the Kinect sensor according to the principle of structural light and recognize the user's finger gestures. Combined with the virtual interface, the user's gestures can be interpreted as control commands.

1) GESTURE DEFINITION

Our system supports similar gestures with those for hand-held devices, but there are some different definitions since our gestures are applied in 3D space. Table 1 lists the detailed definition of the total six gestures. The set of finger gestures our system recognizes can be decomposed into three primitive gesture types: **Click** (touching an UI element in the virtual interface and then release), **Hold** (touching an UI element and

TABLE 1. Gesture definition in our human-3DTV interaction paradigm.

Gestures	Definition
Click	Position one finger at a 3D location, keep its X, Y coordinates in WCS relatively stable while moving forward and then backward along Z-axis.
Hold	Position one finger at a 3D location, keep its X, Y and Z coordinates relatively stable for more than 1 second.
Slide	Move one finger along X or Y-axis while keeping its Z-coordinate relatively stable.
Double-Click	Continuously "Click" the same position twice in 1 second.
Drag	"Hold" the finger at a 3D position to select the target, and then "Slide" to move the target to a new location.
Multi-touch Zoom	Both of two hands stretched out one finger, "Hold" the two buttons on the virtual interface, and "Slide" them towards each other or away from each other.

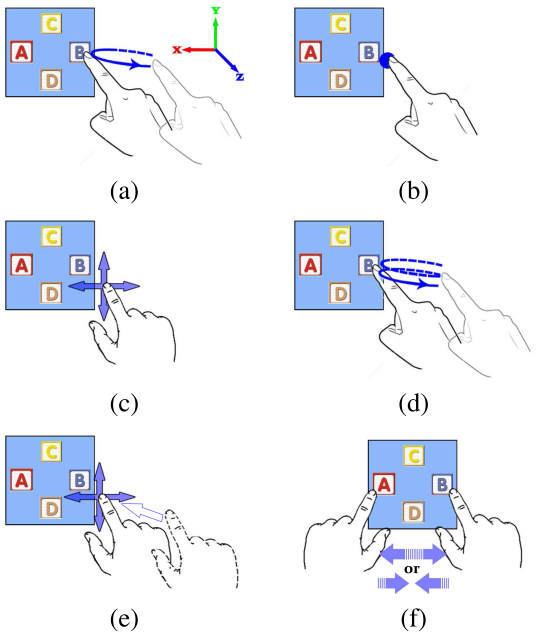


FIGURE 6. The simple set of finger gestures that can be used to interact with the virtual interface. There are three primitive gestures: (a) Click - moving forward and backward along Z-axis; (b) Hold - keeping the finger still in the space; (c) Slide - with the finger flicking the interface; and three "Combo" gestures: (d) Double-Click - do "Click" gesture twice in 1 second; (e) Drag - "Hold" the target and then "Slide" it to the object location, (f) Multi-touch Zoom - moving two hands apart/together to scale up/down the image.

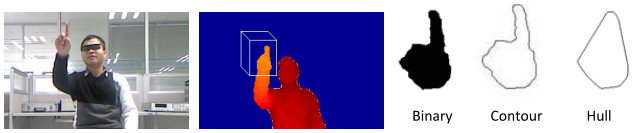


FIGURE 7. Fingertip detection: (a) (b) Hand object detection and (c) Finger segmentation and fingertip detection.

then keeping still for a minimal duration) and **Slide** (swiping an UI element in the virtual interface horizontally or vertically). Based on the combination of the above three primitive gesture types, many "Combo" finger gestures can be further recognized by our system. In our prototype, we found the following three "Combo" gestures: **Double-Click**, **Drag** and **Multi-touch Zoom**. Together with the primitive gestures, the "Combo" gestures are to be sufficient for a rich set of control commands suitable for interacting with 3DTV.

2) FINGER DETECTION, SEGMENTATION AND TRACKING

Given the coordinate of the displayed virtual interface, the finger detection process identifies whether there is a hand object exist in a nearby region of the interface, in stead of scanning the entire space. Once a hand is detected, we apply the method [38] to obtain a bounding box of the hand, as illustrated in Figure 8(b). In order to locate the precise position of the fingertip, the finger detection process performs hand segmentation, hand contour detection and convex hull

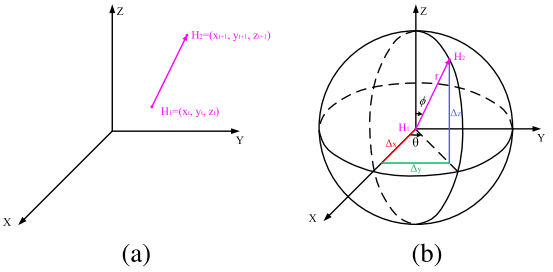


FIGURE 8. Orientation of two consecutive points: (a) in the Cartesian coordinate system and (b) in the spherical coordinate system.

extraction of the hand shape (Figure 8(c)) successively to obtain the up-most point of the hull as the observed fingertip location. The observation is finally sent to a Kalman Filter for smooth tracking of the fingertip location. This step has assumed that the user keeps his hand at up-straight position when performing finger gesture, which holds in general as observed throughout our experiment.

3) GESTURE RECOGNITION BASED ON HMM

Selecting good features to recognize hand gestures play significant role in our Human-3DTV system. There are three basic features: location, orientation and velocity. The previous research [39], [40] showed that the orientation feature is the best in term of accuracy results. Therefore, we will rely upon the orientation feature as the main feature in our system. A gesture path is spatio-temporal pattern which consists of finger point (x, y, z). For two consecutive points $H_1 = (x_t, y_t, z_t)$ and $H_2 = (x_{t+1}, y_{t+1}, z_{t+1})$ from hand gesture path, the orientation is a vector $(\Delta_x, \Delta_y, \Delta_z)$ which can be represented as (r, θ, ϕ) in the Spherical coordinate system, where

$$r = \sqrt{\Delta_x^2+\Delta_y^2+\Delta_z^2}, \quad \theta = arccos\frac{\Delta_z}{r}, \ \phi = arctan\frac{\Delta_y}{\Delta_x}. \quad (4)$$

Here, $\delta - x = x_{t+1} - x_t$, $\delta_y = y_{t+1} - y_t$, and $\delta_z = z_{t+1} - z_t$. The orientation θ and ϕ is quantized by dividing it by 45° to generate the codewords from 1 to 6 (as shown in Table 2). The discrete vector is determined and then is used as input to HMM.

TABLE 2. Coding of the direction angle.

Range of θ	Range of ϕ	Coding Number
$[0°, 360°)$	$[0°, 45°)$	1
$[0°, 360°)$	$[135°, 180°)$	2
$[0°, 45°), [315°, 360°]$	$[45°, 135°)$	3
$[45°, 135°)$	$[45°, 135°)$	4
$[135°, 225°)$	$[45°, 135°)$	5
$[225°, 315°)$	$[45°, 135°)$	6

The Hidden Morkov Model (HMM) [41] has been proven to be a powerful stochastic approach and found robust in speech and text (printed and hand written) recognition. We apply HMM for gesture recognition due to its ability

to cope with variable-length observation sequences obtained from images. Generally, the HMM contains a fixed number of hidden states and it follows the first-order Markov assumption where each state q_t at time t depends only on the state q_{t-1} at time $t-1$. Assume the number of hidden states is N and the number of observations is M, for a observation sequence $\mathcal{O} = (o_1, o_2, \ldots, o_T)$ of length T, its state sequence is $\mathcal{Q} = (q_1, q_2, \ldots, q_T)$. The HMM includes three parameters $\lambda = (\pi; A; B)$ [4], where $\pi = \{\pi_1, \pi_2, \ldots, \pi_N\}$ represents an initial state probability vector and $\sum_{i=1}^{N} \pi_i = 1$, $A_{N\times N} = \{a_{ij} | a_{ij} = P(q_{t+1} = j | q_t = i)\}$ is a state transition probability matrix and $\sum_{i=1}^{N} a_{i,j} = 1$, and $B_{N\times M} = \{b_j(k) | b_j(k) = P(o_t = k | q_t = j)\}$ refers to an observation probability matrix and $\sum_{j=1}^{N} b_j(k) = 1$. Here, π_i is the initial probability of state i, $a_{i,j}$ is the transition probability from state i to state j, and $b_j(k)$ is the state observation likelihood of the observation $o_t = k$ being generated from state j. For a HMM model $\lambda = (\pi; A; B)$ and an observation sequence $\mathcal{O}$, the likelihood $P(\mathcal{O}|\lambda)$ is calculated by:

$$P(\mathcal{O}|\lambda) = \sum_{\mathcal{Q}} \pi_{q_1} b_{q_1}(o_1) \prod_{T} a_{q_{T-1},q_T} b_{q_T}(o_T). \tag{5}$$

We apply the Viterbi decoding algorithm to search the subsequence of an observation that matches best to a given HMM. For 6 gestures in our system, we generate 6 HMMs $\lambda^c = (\pi^c; A^c; B^c)$, $c = 1, 2, \ldots, C$. Given a sequence of unknown category $\mathcal{O}$, we calculate the likelihood $P(\lambda^i|\mathcal{O})$ for each HMM and select the best matched HMM model $\lambda^{\hat{c}}$ where:

$$\hat{c} = \underset{i}{\operatorname{argmax}}\, P(\lambda^i|\mathcal{O}). \tag{6}$$

The parameters of the HMM of each gesture, are learned using the Baum-Welch algorithm [42] which is a special case of the Expectation-Maximization algorithm. The algorithm will let us train both the transition probabilities A and the observation probabilities B of the HMM, by computing an initial estimate for the probabilities, then using those estimates to computing a better estimate, and so on, iteratively improving the probabilities $P(\mathcal{O}|\lambda)$ that it learns. In our implementation, we use 60 training data per gesture to train 6 HMMs. For classifying an observed sequence $\mathcal{O}$, the classifier choose the model whose likelihood is highest as the recognition result with (6). The whole gesture recognition algorithm is shown in Figure 9.

IV. USER STUDY

To demonstrate and evaluate the feasibility of our proposed human-3DTV interaction method with the touch-based virtual interface, we conduct a comprehensive user study to demonstrate the key performance characteristics of our system. Besides, we select a commercial product, the Samsung UA46ES7000 smart TV which supports free hand gestures input, as a baseline and make a comparison with our system.

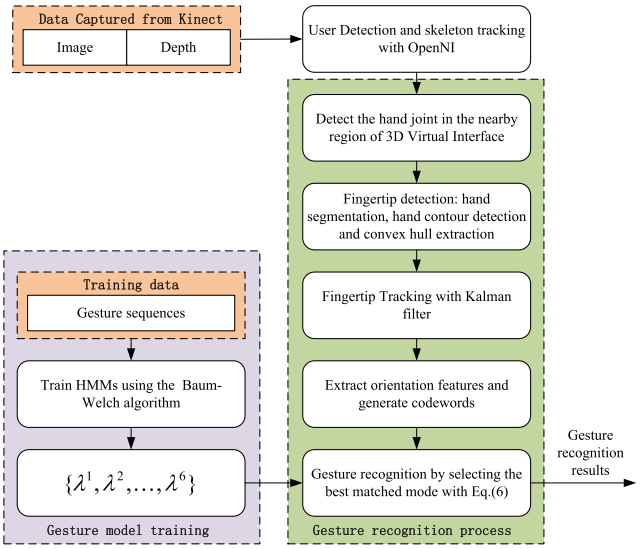


FIGURE 9. Gesture recognition diagram.

A. APPARATUS

In our developed prototype system, we select a Samsung UA46ES7000 smart 3DTV as the 3D display device, a Microsoft Kinect sensor for capturing the RGB and depth information, and a Dell 5560 laptop to run our system. The laptop has a Core i7-6820HQ cpu and 16GB DDR3 memory. The users use 3D shutter glasses to perceive the 3D effect, and the shutter glasses are the only piece of device the users needed to wear. We use the OpenNI SDK [36] to extract RGB and depth images of the same size 640 × 480 at 30 fps, and perform skeleton tracking. The depth resolution of the Kinect sensor increases quadratically from a few millimeters at 0.5 m distance to about 4 cm at the maximum range of 5 meters [43].

B. PARTICIPANTS

20 participants take part in the user study. The participants include 13 male and 7 female, aging between 20 to 50. Among of the 20 participants, 5 are research engineers, 2 are university professors, 3 are lab technicians, 6 are graduate students and 4 are under-graduates. All participants have used 3DTV or watched 3D movies before. All but 2 have experience with touch screen devices such as smart-phone or tablet.

C. GESTURES AND THE VIRTUAL INTERFACE

The virtual interface in our prototype system can be adaptive to the 3DTV content. Here, we design different virtual interfaces to test three primitive gestures as well as three "Combo" gestures.

1) CLICK AND Hold

The two primitive gestures are tested with a 9-key numeric keyboard that contain 9 numbered buttons. Each button is in a rectangular shape, with the button size and spacing distance between button adjustable (in the default setting, the button

size is $A_{Large_1} = (29.2mm, 26.3mm)$ and the spacing distance is 20mm), as shown in Figure 10(a). If any buttons are being clicked or hold, the system makes a beep sound and highlights the button with a color border, e.g., with a yellow border for gesture "Click" and a red border for gesture "Hold". The color border disappears in 0.3 second.

2) SLIDE

The gesture is tested using a virtual interface with 4 buttons in a rhombus shape, two in horizontal direction and two in vertical direction (as shown in Figure 10(b)). The two horizontal buttons are used to perform left/right sliding, where for sliding left, the user could draw a line from the left button to the right button, and visa verse. Similarly, the two vertical ones are used for testing up/down sliding. Once the gesture is recognized by the system, a pink colored arrow pointing the sliding direction would be superimposed on the virtual interface, and fades away after 0.3 seconds.

3) DOUBLE-CLICK

This "combo" finger gestures is tested under the same 9-key numeric keyboard for testing "Click". The only difference is that, if an "Double-Click" is recognized, the system would highlight the clicked button with a green border.

4) DRAG

The gesture is recognized when the user tries to move an item like the cursor or an object image to a specified place in the screen. The task is very similar to the drag and drop action on a smart phone. To test this gesture, the virtual interface is an object image floating over a black background, as shown in Figure 10(d).

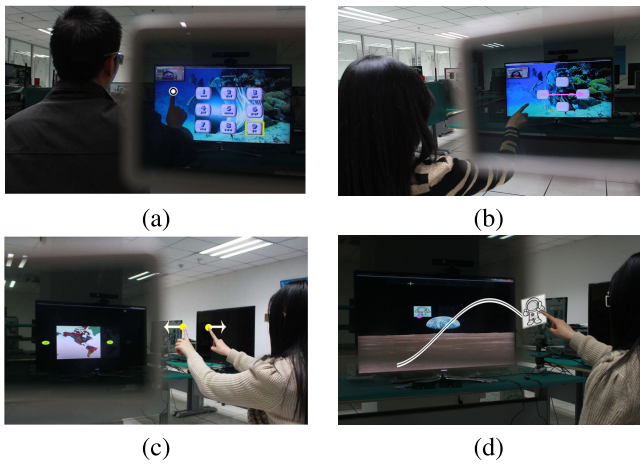


FIGURE 10. Illustration of human-3DTV interaction with various gesture inputs on the virtual interface: (a) "Click", "Hold" or "Double-click", (b) "Slide", (c) "Multi-touch zoom", and (d) "Drag".

5) MULTI-TOUCH ZOOM

The "Multi-touch Zoom" gesture is the only gesture that requires two-hand operation. The virtual interface for testing "Multi-touch Zoom" contains two given points. The participant first holds the two points using two fingers, and then slides away/together to zoom in/out an image, as shown in Figure 10(c).

D. PROCEDURE

Before the evaluation starts, we give a brief demonstration of each gesture to each participant. Then the participant sits in front of the 3DTV (2 meters away), put on the shutter glasses, and start to watch the 3D scene. Once he/she is ready, we start to display a virtual interface. As our virtual interface is set to be 0.5m in front of the user, some participants may have difficulty perceiving our interface object immediately. To address the issue, we simulate an "fly-in" effect, i.e. our interface "flies" from the TV screen towards the participant and stops at 0.5m in front of the user, to help the participant perceive the virtual interface. The whole flying process takes 10 seconds, and only needs to be performed once. We give each participant a maximum of 10 minutes to warm up. They can practice with items in the virtual interface until they feel confident to test. During the warm-up period, the system automatically runs skeleton calibration, shutter glass detection and eye coordinate estimation.

The user data is collected in the following way. For "Click" and "Hold", each participant repeats 3 times on each button of the 9-key numeric keyboard, and this procedure produces 27 trials (9 buttons $\times$ 3 trials) for each gesture. For the gesture "Slide", participants are asked to touch one of 4 buttons in the rhombus interface and perform four sliding directions: sliding left, sliding right, sliding up and sliding down. Each sliding direction is repeated 5 times. For three combo gestures, "Double Click" is taken with the same setting with "Click" and "Hold", i.e. each numbered button is repeated 3 times. Both "Drag" and "Multi-touch" gestures are repeated 10 times for each participant. We record the outputs of the recognition error rate (or the recognition accuracy rate), the interaction speed, and the location of touching points.

To test the influence of the distance between the user and 3DTV, participants interact with our human-3DTV system with three primitive gestures at three specified distances: Close (1.5m), Average (2m) and Far (2.5m). Besides, we quantify the influence of different button sizes and spacing distances of the projected virtual interface on the accuracy rate. More details are shown in Subsections V-C and V-D.

After finishing the above test, participants are asked to complete a questionnaire containing 14 questions. For example, some questions are asking participants if the primitive and "Combo" gestures are important for human-3DTV interaction. This procedure takes about 3 minutes.

V. RESULT AND DISCUSSION

This section discusses the results in the terms of gesture recognition accuracy and input speed, in comparison with the Samsung UA46ES7000 smart 3DTV and the work in [7]. The Samsung UA46ES7000 smart 3DTV tracks the user's

open-palm hand and displays a curser in the screen with a vision camera. It supports several hand gesture recognition such as ''Click'' that is recognized when the user first makes a fist and then unclench the fist. ''Hold'' is recognized when the user holds his fist for a certain duration. ''Slide'' is recognized when the user slides the palm left/right/up/down. ''Double-Click'' is performing ''Click'' twice. ''Drag'' is available when dragging the volume bar to a desired value. The Samsung tv does not contain any ''Multi-touch'' functions, hence we don't compare with it.

We give the detailed steps to control the gesture system of Samsung UA46ES7000 smart 3DTV: (1) Confirm that the gesture controls are enabled in the TV and that the camera is set for the viewing area. (2) Raise the hand and hold it up for about 2 seconds. (3) Wave at the TV. (4) If gesture controls have been activated a pointer will appear on the TV and a gesture control command bar will appear at the bottom of the TV display with available commands. If the command bar and pointer do not appear then return to Step 1. (6) Use the hand to guide the pointer around the TV. (6) With the pointer hovering over the object that the user would like to select, close the hand to select.

A. GESTURES RECOGNITION ACCURACY

The gesture recognition error rate is measured by the percentage of incorrectly recognized gestures. For ''Click'', ''Hold'' and ''Double-Click'', incorrect recognition means that either the gesture is not detected or not recognized as the correct number. For the ''Slide'' gesture, incorrect recognition denotes the gestures that are not detected or not recognized as the correct sliding direction. For ''Drag'', the gestures may be not detected or putting the object image in an incorrect place. For Multi-touch Zoom, incorrect recognition is counted when the target doesn't zoom accordingly.

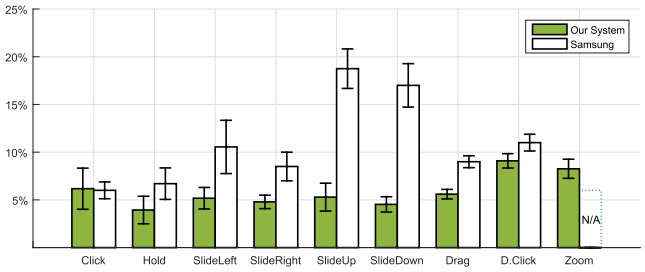


FIGURE 11. The comparison of gestures recognition error rate between the presented human-3DTV system and Samsung smart TV system. The error bar means the standard deviation of results.

Figure 11 illustrates the comparison of gestures recognition error rate between our system and the Samsung system. For the ''Click'' gesture, both systems get almost the same error rate. However, the standard variance of our system is a bit higher than the Samsung system. The reason is that the Samsung system moves the projected cursor to the button in the screen and then makes a fist and unclenches, while our system does not have the projected cursor to help indicate if the fingertip is right at the position of the virtual interface. Except ''Click'', all other gestures in our interaction system get lower error rate than the benchmark Samsung system. For the ''Slide'' gesture, we can see that the Samsung system gets lower performance especially on performing SlideUp and SlideDown gestures which have 15% error rate. In contrast, the error rate of our system is only 5%. Because in the Samsung system the user has to slide the arm in an open world without any ''seeable'' object for reference, participants' gestures have large variance on the amplitude and direction of motion. In contrast, our system provides an intuitive and easy way to touch an ''seeable'' virtual interface with its items projected at fixed world coordinates, so our system is able to recognize the user's gesture accurately. The ''Double-Click'' gesture in our system gets higher error rate than the ''Click'' gesture, as ''Double-Click'' needs to perform ''Click'' twice. Overall, the average error rate of all gesture types with our presented Human-3DTV system is about 5% which is much lower than about 10% average error rate with the Samsung 3DTV.

It is also observed that ''Hold'' gets the lowest error rate among of all gestures. The reason is that ''Hold'' needs to keep the finger still for a duration, which is much easier to be extracted and recognized by the gesture recognition algorithm based on the HMM. . Another discovery is that the average error of ''Click'' is slightly higher than ''Slide'', although ''Click'' and ''Slide'' both refer to the linear motion: the former is in the direction of Z-axis and the latter is in the direction of X-axis or Y-axis. The reason is that, for the RGB and depth data captured from the Kinect sensor, it is easier to distinguish the motion in the X-Y plane than that in the direction of Z-axis.

The recognition confusion matrix for the gesture recognition based on HMM is shown in Table 3. Note that the values in the reported confusion matrix just compute the gesture recognition rate of all recognized gestures and do not consider if gestures hit the correct button in the virtual interface. As can be seen from the table, the ''Slide'' and ''Multi-touch Zoom'' gestures have no ambiguity with the other gestures, and misclassifications frequently occur among these gestures of ''Click'' and ''Double Click''.

TABLE 3. Confusion matrix of hand gesture recognition using Hidden Markov Model in our presented human-3DTV system. ''D.Click'' denotes the ''Double Click'' gesture and ''M.Zoom'' denotes the ''Multi-touch Zoom'' gesture.

Accuracy(%)	Click	Hold	Slide	D.Click	Drag	M. Zoom
Click	**97.8**	0.4	-	1.8	-	-
Hold	1.2	**98.4**	-	-	0.4	
Slide	-	-	**100**	-	-	-
D.Click	3.4	-	-	**96.6**	-	-
Drag	0.4	1.0	0.6	-	**98.0**	-
M. Zoom	-	-	-	-	-	**100**

B. GESTURES INPUT SPEED

The input speed is based on the duration from when the participant start to perform the gesture until the correct visual

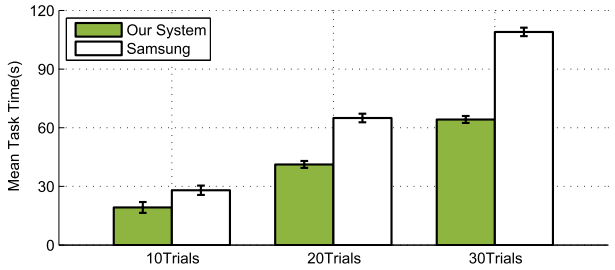


FIGURE 12. Comparison of input speed.

response is shown on the screen. The participant might repeat his gesture several times for correct recognition if the gesture recognition in our system fails. Figure 12 illustrates the input speed comparison between the presented Human-3DTV system and the Samsung UA46ES7000 3DTV. We list the comparisons with 10 trials, 20 trials and 30 trials of the "Click" gesture. The results show that the presented Human-3DTV system achieve obviously fast speed. To move an on-screen cursor in the Samsung TV, the user needs to perform at least three steps: firstly raise the palm and find the cursor in the screen, then move the open-palm hand to fine tune the cursor on the correct button, and finally perform the "Click" gesture by making a fist and then unclenching it. Obviously, this process is cumbersome and easily makes the user tiring and uncomfortable. In contrast, with the help of the virtual interface, our method requires much less time to finish each trial, as the user just needs to stretch the arm and hand and touch what they see, which is intuitive and easy to use.

C. INFLUENCE OF THE DISTANCE BETWEEN USERS AND 3DTV

To test the influence of the distance between the user and 3DTV, participants interact with our Human-3DTV system with three primitive gestures at three specified distances: Close (1.5m), Average (2m) and Far (2.5m). Table 4 illustrates the error rate of 3 different distances. As can be seen, "Far" distance leads to the lowest accuracy rate because the Kinect sensor produces larger depth measurement error at the range of 2.5 meters. On the other side, although at the "Close" distance the participants could achieve slightly higher accuracy rate, participants may feel uncomfortable when eyes are too close to TV. Hence, the "Average" distance is a more balanced setting for 3DTV users.

TABLE 4. Accuracy rate of different distances between the user to 3DTV. "Close", "Average" and "Far" denote that the user is 1.5 meters, 2 meters and 2.5 meters away from the 3DTV, respectively. We report the mean and stand deviation values of the gesture accuracy rate of all 20 participants. The best values are shown in bold.

Distances	Click		Hold		Slide	
	Mean(%)	Std. Dev.	Mean(%)	Std. Dev.	Mean(%)	Std. Dev.
Close	**94.6**	**0.03**	**97.2**	**0.03**	**96.3**	**0.02**
Average	93.8	0.04	96.1	**0.03**	95.1	**0.02**
Far	84.2	0.06	88.1	0.07	85.2	0.06

D. INFLUENCE OF BUTTON SIZES AND SPACING DISTANCE ON THE VIRTUAL INTERFACE

In this subsection, we quantify the influence of different button sizes and spacing distances of the projected virtual interface on the accuracy rate, to find the optimal setting that minimizes gesture recognition error for the three primitive gestures.

We test three primitive gestures on 5 different button sizes on the 9-key numeric keyboard: A_{Small_1}, A_{Small_2}, A_{Medium}, A_{Large_1} and A_{Large_2}. The (Length, width) dimensions (in millimeters) of their buttons are $A_{Small_1} = (17.5, 15.8)$, $A_{Small2} = (20.9, 18.8)$, $A_{Medium} = (25.1, 22.6)$, $A_{Large_1} = (29.2, 26.3)$ and $A_{Large_2} = (32.2, 29.3)$, respectively. For all settings, the spacing distance between two buttons is fixed to be 20mm. Table 5 lists the accuracy rate of different button sizes on the 9-key numeric keyboard. Overall our system with the A_{Large_2} setting achieves the best gesture accuracy rate. Along with the button size decreases, the gesture accuracy rate goes down. The results show that the accuracy rate drops dramatically when the button size in the virtual interface decreases to A_{Small_1}. For example, the accuracy rate of "Click" goes down from $A_{Medium} - 90.3\%$ to $A_{Small_1} - 78.5\%$, and the accuracy rate of "Click" decreases from $A_{Medium} - 91.1\%$ to $A_{Small_1} - 79.9\%$. Table 5 shows that the presented Human-3DTV system achieves satisfying gestures recognition rate if the button size in the virtual interface is larger than $A_{Medium} = (25.1mm, 22.6mm)$.

TABLE 5. The accuracy rate of different button sizes on the 9-key numeric keyboard. The best values are shown in bold.

Button sizes	Click		Hold		Slide	
	Mean(%)	Std. Dev.	Mean(%)	Std. Dev.	Mean(%)	Std. Dev.
A_{Large_2}	**94.1**	**0.03**	**96.5**	**0.03**	**96.0**	**0.02**
A_{Large_1}	93.8	0.04	96.1	**0.03**	95.1	**0.02**
A_{Medium}	90.3	0.05	91.1	0.04	94.2	**0.02**
A_{Small_1}	78.5	0.05	79.9	0.04	90.6	**0.02**
A_{Small_2}	71.1	0.05	72.4	0.04	88.2	0.03

We test three primitive gestures on 3 different spacing distance between buttons on the virtual keyboard: $d_{Small} = 10mm$, $d_{Medium} = 20mm$ and $d_{Large} = 30mm$. For all settings, the (Length, Width) of buttons is set to $A_{Medium} = (25.1mm, 22.6mm)$. For all settings, the spacing distance between two buttons is fixed to be 20mm. As shown in Table 6, overall our system with the d_{Large} setting achieves

TABLE 6. The accuracy rate of different spacing distances between buttons on the virtual keyboard. The best values are shown in bold.

Spacing distances	Click		Hold		Slide	
	Mean(%)	Std. Dev.	Mean(%)	Std. Dev.	Mean(%)	Std. Dev.
d_{Large}	**94.7**	**0.03**	**96.5**	**0.03**	**95.2**	**0.02**
d_{Medium}	93.8	0.04	96.1	0.03	95.1	**0.02**
d_{Small}	82.7	0.05	84.7	0.04	91.6	**0.02**

TABLE 7. The questionnaire of the presented human-3DTV system. The statistics data is also reported after each option.

Index	Question and Answer Options				
Q1	**Please select the age range that you are in:**				
	<18-21 (20%)	22-25 (20%)	26-30 (35%)	31-40 (15%)	>40 (10%)
Q2	**Please select your gender:**				
	Male (65%)		Female (35%)		
Q3	**How often do you watch a 3D movie with the 3DTV set or in a cinema?**				
	Very often (10%)	Often (75%)	Sometimes (15%)	Rarely (0%)	Never (0%)
Q4	**How often do you use the gesture inputs (*e.g.* the Microsoft Kinect games)?**				
	Very often (0%)	Often (0%)	Sometimes (60%)	Rarely (25%)	Never (15%)
Q5	**Are you easy to see the virtual interface flying to you?**				
	Very difficult (0%)	Difficult (0%)	Neutral (20%)	Easy (65%)	Very easy (15%)
Q6	**Do you think the "Click" gesture is useful for Human-3DTV interaction?**				
	Not at all (0%)	Slightly useful (0%)	Neutral (0%)	Very useful (10%)	Extremely useful (90%)
Q7	**Do you think the "Hold" gesture is useful for Human-3DTV interaction?**				
	Not at all (0%)	Slightly useful (0%)	Neutral (0%)	Very useful (5%)	Extremely useful (95%)
Q8	**Do you think the "Slide" gesture is useful for Human-3DTV interaction?**				
	Not all important (0%)	Slightly useful (0%)	Neutral (15%)	Very useful (35%)	Extremely useful (50%)
Q9	**Do you think the "Double Click" gesture is useful for Human-3DTV interaction?**				
	Not at all (0%)	Slightly useful (15%)	Neutral (45%)	Very useful (30%)	Extremely useful (10%)
Q10	**Do you think the "Drag" gesture is useful for Human-3DTV interaction?**				
	Not at all (5%)	Slightly useful (10%)	Neutral (40%)	Very useful (35%)	Extremely useful (10%)
Q11	**Do you think the "Multi-touch Zoom" gesture is useful for Human-3DTV interaction?**				
	Not at all (0%)	Slightly useful (10%)	Neutral (20%)	Very useful (50%)	Extremely useful (20%)
Q12	**Are you satisfied with the interaction speed?**				
	Not at all satisfied (0%)	Slightly satisfied (0%)	Unsure (0%)	Very satisfied (10%)	Extremely satisfied (90%)
Q13	**To enter WiFi password, would you prefer our system rather than a conventional remote controller?**				
	Not a priority (0%)	Low priority (0%)	Neutral (10%)	High Priority (80%)	Essential Priority (10%)
Q14	**Would you prefer our system rather than the method of gesture input?**				
	Not a priority (0%)	Low priority (0%)	Neutral (0%)	High Priority (75%)	Essential Priority (25%)

the best gesture accuracy rate. If the buttons are too close, the gesture accuracy rate goes down as users are easier to touch incorrect buttons when they move their fingers in the region of the virtual interface.

E. VISUALIZATION OF GESTURE HIT MAP

We plot in Figure 13 a hit map of the "Hold" gesture performed on the 9-key virtual keyboard. In the figure, nine rectangles represent the relative locations of the 9 numeric keys, and each red dot corresponds to the touch point location of a "Click" gesture. Intuitively we expect the distribution of the red dot to be Gaussian centralized at each button. The figure shows that all the touch points are clearly clustered into nine groups, each of which is within the neighborhood of a key. Additionally, we discover that there was a small systematic offset between where the user clicked and where the system believed the user clicked, in agreement with previous findings in the touchscreen literature [44]. We can simply apply a global X/Y offset to compensate for the system inaccuracy.

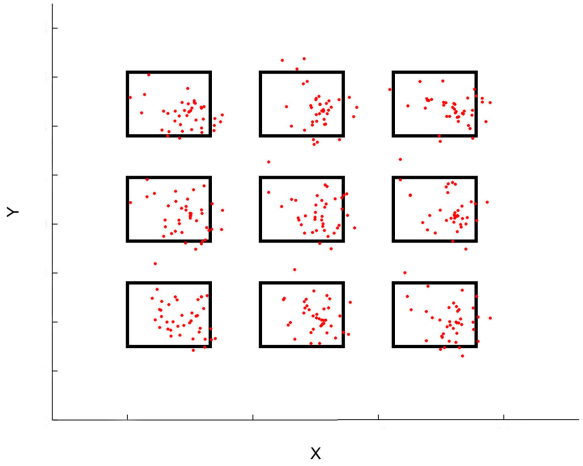


FIGURE 13. The plot of the touch points of the recognized "Click" gestures in the subspace x-y in WCS.

F. QUESTIONNAIRE FOR THE PRESENTED HUMAN-3DTV SYSTEM

Table 7 shows the designed questionnaire for the user survey of the presented Human-3DTV system. Our questionnaire is based on the System Usability Scale (SUS) [45], which is composed of 10 statements that are scored on a 5-point scale of strength of agreement. Final scores for the SUS can range from 0 to 100, where higher scores indicate better usability. In our questionnaire, each question in Q5-Q14 contains five response options for participants, from Strongly disagree (left) to Strongly agree (right). The mean SUS score of all participants is 79.5 which means our system gets a Good rating. All participants have the experience of watching 3D movie, 85% have tried the gesture input method, and 80% feel easy to see the projected virtual interface. Among of all 6 gestures, both gestures of "Click" and "Hold" get more than 90% of votes on "Extremely useful", and "Slide" get 50% of votes. In contrast, three "Combo" gestures get

less positive votes. Besides, 90% think our system has good experience on the interaction speed. More than 90% prefer our system rather than a conventional remote controller or gesture input.

VI. CONCLUSION

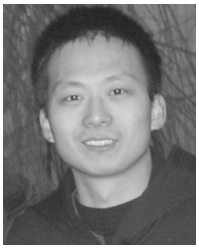
In this paper, we present a novel human-3DTV interaction system by utilizing a virtual interface to combine the advantages of free-hand gesture input and touch screen interaction. In our human-3DTV interacting system, the virtual interface display module generates a stereoscopic virtual interface with predefined contents at a specified location in front of the user, and the finger gesture recognition module searches the neighboring region of the virtual interface for gesture recognition. We define three primitive gesture types and three ''Combo'' gestures. The evaluation reveals that the proposed human-TV interaction system is effective and get higher accuracy rate with the help of the virtual interface. The future work of the research includes the following three aspects: First, we will plan to enrich the content of the virtual interface by designing more elements, so that our system would develop more applications of virtual reality. We will also add a function that the virtual interface can be closed, shrunk or moved to other positions for not blocking the direct line of sight after the user finishes the interaction. Second, we will combine three primitive gestures proposed in our work to perform more functions. Thirdly, as the current system cannot address the situation that the users roll, tilt or rotate their heads, we plan to develop an algorithm to detect the angles of the glasses and the user's face, so as to estimate the positions of the user's two eyes. Finally, we hope our work can be implemented in commercial 3DTV.

REFERENCES

[1] Y. Rabhi, M. Mrabet, and F. Fnaiech, ''Intelligent control wheelchair using a new visual joystick,'' *J. healthcare Eng.*, vol. 2018, Feb. 2018, Art. no. 6083565.
[2] M. Long and C. Gutwin, ''Effects of local latency on game pointing devices and game pointing tasks,'' in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2019, p. 208.
[3] A. Lorenz, C. F. De Castro, and E. Rukzio, ''Using handheld devices for mobile interaction with displays in home environments,'' in *Proc. 11th Int. Conf. Hum.-Comput. Interact. Mobile Devices Services*, 2009, Art. no. 18.
[4] S. C. Lee, M. C. Cha, and Y. G. Ji, ''Investigating smartphone touch area with one-handed interaction: Effects of target distance and direction on touch behaviors,'' *Int. J. Hum.-Comput. Interact.*, vol. 35, no. 16, pp. 1532–1543, 2018.
[5] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, ''State-of-the-art speech recognition with sequence-to-sequence models,'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4774–4778.
[6] W. Zeng, C. Wang, and Q. Wang, ''Hand gesture recognition using leap motion via deterministic learning,'' *Multimedia Tools Appl.*, vol. 77, no. 21, pp. 28185–28206, 2018.
[7] G. Li, H. Wu, G. Jiang, S. Xu, and H. Liu, ''Dynamic gesture recognition in the Internet of Things,'' *IEEE Access*, vol. 7, pp. 23713–23724, 2018.
[8] V. Joseph, A. Talpade, N. Suvarna, and Z. Mendonca, ''Visual gesture recognition for text writing in air,'' in *Proc. 2nd Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, Jun. 2018, pp. 23–26.
[9] S. Mohammadi and R. Maleki, ''Air-writing recognition system for persian numbers with a novel classifier,'' *Vis. Comput.*, pp. 1–15, Jun. 2019.
[10] J. Singha, A. Roy, and R. H. Laskar, ''Dynamic hand gesture recognition using vision-based approach for human–computer interaction,'' *Neural Comput. Appl.*, vol. 29, no. 4, pp. 1129–1141, 2016.
[11] P. Kumar, R. Saini, S. K. Behera, D. P. Dogra, and P. P. Roy, ''Real-time recognition of sign language gestures and air-writing using leap motion,'' in *Proc. 15th IAPR Int. Conf. Mach. Vis. Appl. (MVA)*, May 2017, pp. 157–160.
[12] (2010). *Microsoft Kinect for xbox360*. [Online]. Available: http://www.xbox.com/en-US/kinect
[13] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, ''Real-time human pose recognition in parts from single depth images,'' in *Proc. CVPR*, Jun. 2011, pp. 1297–1304.
[14] X. Ma and J. Peng, ''Kinect sensor-based long-distance hand gesture recognition and fingertip detection with depth information,'' *J. Sensors*, vol. 2018, Mar. 2018, Art. no. 5809769.
[15] M. J. Cheok, Z. Omar, and M. H. Jaward, ''A review of hand gesture and sign language recognition techniques,'' *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 1, pp. 131–153, Jan. 2017.
[16] E. Oliemat, F. Ihmeideh, and M. Alkhawaldeh, ''The use of touch-screen tablets in early childhood: Children's knowledge, skills, and attitudes towards tablet technology,'' *Children Youth Services Rev.*, vol. 88, pp. 591–597, May 2018.
[17] D. M. Twomey, C. Wrigley, C. Ahearne, R. Murphy, M. De Haan, N. Marlow, and D. M. Murray, ''Feasibility of using touch screen technology for early cognitive assessment in children,'' *Arch. Disease Childhood*, vol. 103, no. 9, pp. 853–858, 2018.
[18] S. Zhang, J. Wang, Y. Gong, and S. Zhang, ''Free-hand gesture control with 'touchable' virtual interface for human-3DTV interaction,'' in *Proc. 3DTV-Conf., True Vis.-Capture, Transmiss. Display 3D Video (3DTV-CON)*, Jul. 2015, pp. 1–4.
[19] F. Kuester, M. Chen, M. E. Phair, and C. Mehring, ''Towards keyboard independent touch typing in VR,'' in *Proc. VRST*, 2005, pp. 86–95.
[20] B. Li, Y. Sun, G. Li, J. Kong, G. Jiang, D. Jiang, B. Tao, S. Xu, and H. Liu, ''Gesture recognition based on modified adaptive orthogonal matching pursuit algorithm,'' *Cluster Comput.*, vol. 22, pp. 503–512, Jan. 2019.
[21] B. Fang, F. Sun, H. Liu, and C. Liu, ''3D human gesture capturing and recognition by the IMMU-based data glove,'' *Neurocomputing*, vol. 277, pp. 198–207, Feb. 2018.
[22] X. Huang, Q. Wang, S. Zang, J. Wan, G. Yang, Y. Huang, and X. Ren, ''Tracing the motion of finger joints for gesture recognition via sewing RGO-coated fibers onto a textile glove,'' *IEEE Sensors J.*, vol. 19, no. 20, pp. 9504–9511, Oct. 2019.
[23] *Gesture Control Technology: An Investigation on the Potential Use in Higher Education*. Accessed: Nov. 11, 2019. [Online]. Available: https://intranet.birmingham.ac.uk/it/innovation/documents/public/Gesture-Control-Technology.pdf
[24] *Soli*. Accessed: Nov. 11, 2019. [Online]. Available: https://atap.google.com/soli/
[25] B. Ghojogh, H. Mohammadzade, and M. Mokari, ''Fisherposes for human action recognition using Kinect sensor data,'' *IEEE Sensors J.*, vol. 18, no. 4, pp. 1612–1627, Feb. 2017.
[26] G. Borghi, R. Vezzani, and R. Cucchiara, ''Fast gesture recognition with multiple stream discrete HMMs on 3D skeletons,'' in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 997–1002.
[27] W. T. Freeman and C. D. Weissman, ''Television control by hand gestures,'' in *Proc. AFGRW*, 1995, pp. 179–183.
[28] N. Dawar and N. Kehtarnavaz, ''Real-time continuous detection and recognition of subject-specific smart TV gestures via fusion of depth and inertial sensing,'' *IEEE Access*, vol. 6, pp. 7019–7028, 2018.
[29] X. Li, D. Guan, J. Zhang, X. Liu, S. Li, and H. Tong, ''Exploration of ideal interaction scheme on smart TV: Based on user experience research of far-field speech and mid-air gesture interaction,'' in *Proc. Int. Conf. Hum.-Comput. Interact.* Cham, Switzerland: Springer, 2019, pp. 144–162.
[30] L.-W. Chan, H.-S. Kao, M. Y. Chen, M.-S. Lee, J. Hsu, and Y.-P. Hung, ''Touching the void: Direct-touch interaction for intangible displays,'' in *Proc. CHI*, 2010, pp. 2625–2634.
[31] O. Hilliges, D. Kim, S. Izadi, M. Weiss, and A. Wilson, ''HoloDesk: Direct 3D interactions with a situated see-through display,'' in *Proc. CHI*, 2012, pp. 2421–2430.
[32] H. Benko, R. Jota, and A. Wilson, ''MirageTable: Freehand interaction on a projected augmented reality tabletop,'' in *Proc. CHI*, 2012, pp. 199–208.
[33] A. Mujibiya, T. Miyaki, and J. Rekimoto, ''Anywhere touchtyping: Text input on arbitrary surface using depth sensing,'' in *Proc. UIST*, 2010, pp. 443–444.

[34] J. Shin and C. M. Kim, "Non-touch character input system based on hand tapping gestures using Kinect sensor," *IEEE Access*, vol. 5, pp. 10496–10505, 2017.
[35] M. C. Lam, H. Arshad, A. S. Prabuwono, S. Y. Tan, and S. M. M. Kahaki, "Interaction techniques in desktop virtual environment: The study of visual feedback and precise manipulation method," *Multimedia Tools Appl.*, vol. 77, no. 13, pp. 16367–16398, 2018.
[36] *OpenNI Platform*. Accessed: Jan. 5, 2011. [Online]. Available: http://www.openni.org/
[37] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. CVPR*, Dec. 2001, pp. 511–518.
[38] V. Frati and D. Prattichizzo, "Using Kinect for hand tracking and rendering in wearable haptics," in *Proc. WHC*, Jun. 2011, pp. 317–321.
[39] M. Elmezain, A. Al-Hamadi, and B. Michaelis, "Real-time capable system for hand gesture recognition using hidden Markov models in stereo color image sequences," *J. WSCG*, vol. 16, nos. 1–3, pp. 65–72, 2008.
[40] M. Elmezain, A. Al-Hamadi, J. Appenrodt, and B. Michaelis, "A hidden Markov model-based continuous gesture recognition system for hand motion trajectory," in *Proc. 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.
[41] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, vol. 3. Upper Saddle River, NJ, USA: Pearson, 2014.
[42] L. E. Baum, "An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process," *Inequalities*, vol. 3, pp. 1–8, Jan. 1972.
[43] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of Kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, Feb. 2012.
[44] C. Harrison, H. Benko, and A. D. Wilson, "OmniTouch: Wearable multitouch interaction everywhere," in *Proc. UIST*, 2011, pp. 441–450.
[45] A. Bangor, P. T. Kortum, and J. T. Miller, "An empirical evaluation of the system usability scale," *Int. J. Human–Comput. Interact.*, vol. 24, no. 6, pp. 574–594, 2008.

SHUN ZHANG received the B.S. and Ph.D. degrees in electronic engineering from Xi'an Jiaotong University, Xi'an, China, in 2009 and 2016, respectively. He is currently an Assistant Professor with the School of Electronic and Information, Northwestern Polytechnical University, Xi'an. His research interests include machine learning, computer vision, and human–computer interaction, with a focus on visual tracking, object detection, image classification, feature extraction, and sparse representation.



SHIZHOU ZHANG received the B.E. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 2010 and 2017, respectively. He is currently an Assistant Professor with Northwestern Polytechnical University. His research interests include content-based image analysis, pattern recognition, and machine learning, specifically in the areas of deep learning, image classification, and image parsing.