IEC
Institute of Electronics and Computer

Journal of Artificial Intelligence and Systems, 2019, 1, 125-142
https://iecscience.org/journals/AIS
ISSN Online: 2642-2859

GIST: Gesture-free Interaction by the Status of Thumb; an interaction technique for Virtual Environments

Muhammad Raees[1, *], Sehat Ullah[2], Sami Ur Rahman[3]

Department of Computer Science and IT, University of Malakand, Pakistan
[1.] visitrais@yahoo.com, [2.] sehatullah@uom.edu.pk, [3.] Softrays@hotmail.com
*Corresponding Author: Muhammad Raees, Email: visitrais@yahoo.com

How to cite this paper: Muhammad Raees; Sehat Ullah; Sami Ur Rahman (2019) GIST: Gesture-free Interaction by the Status of Thumb; an interaction technique for Virtual Environments. Journal of Artificial Intelligence and Systems, 1, 125–142. https://doi.org/10.33969/AIS.2019.11008

Received: October 16, 2019
Accepted: November18, 2019
Published: November 22, 2019

Abstract

User interface has special importance in immersive virtual environments. Interactions based on the simple and conceivable gestures of a hand may enhance immersivity of a Virtual Environment (VE). However, due to the structural issues like small size and complex shape of human hand, recognition of hand gestures are more challenging. This work introduces a novel interaction technique to perform the basic interaction tasks by the simple movement of hand instead of distinct gestures. With an ordinary camera, the fist posture of hand is segmented out from the image stream using the optimal segmentation model. Like pressing a button with a thumb, the status of thumb is traced for the activation or deactivation of the interactions. After the activation of interaction, the trajectory of hand is followed to manipulate a virtual object about an arbitrary axis. Without training and comparison of gestures, the basic interactions required in a VE are performed by the perceptive movement of a hand. By incorporating image processing in the realm of VE, the technique is implemented in a case-study project; FIRST (Feasible Interaction by Recognizing the Status of Thumb). A group of 12 users evaluated the system in a moderate lighting condition. Outcomes of the evaluation revealed that the technique is suitable for Virtual Reality (VR) applications.

Keywords

3D interactions, Gestural interfaces, Virtual Reality, Finger Recognition, Computer Vision

1. Introduction

There is an increasing interest in making the interface of VE simpler and engrossing. Besides enabling a user to perform different tasks in a VE, interaction enhances immersivity of a VR system [1]. By now, It has been proved that the use of hand gestures are suitable for natural interactions [2-3]. Various gesture-based techniques have been proposed for 3D interactions using the magnetic [4] and mechanical [5] sensors. However, such systems are inadequate due to their intrinsic limitations and cumbersome setup. Similarly, the tracking of gestures by the Nintendo Wii-Mote [6] needs Oculus Touch,

HTC Vive Controller and HoloLens. Such a complex setup undermines the naturalness of a VE. Fiducial markers have also been proposed for the VR systems [7,8]. However, markers based systems are good for short-range applications [9]. Interaction with a bare hand is more appealing than wearables [10]. Several hand posture-based techniques have been proposed for VR interactions [11-13]. Most of the techniques in the literature [8,14-15] trigger predefined actions at the recognition of specific gestures. Some interaction methods necessitate the training of gestures [16-17]. As Kinect is good for human activity recognition, recent research is based on the Kinect sensor [18]. While Kinect can trace body gestures, it cannot distinguish individual fingers accurately [19].

Although the use of hand gestures is suitable for interactions, due to the structural issues, the recognition of hand gesture is more challenging. For instance, false-tracking [20] and user-side difficulties [21] are the key challenges of gestural interfaces. Accurate recognition of gestures is difficult because of the small and complex structure of hand [22]. Moreover, some people have difficulties in posing specific hand gestures [23]. A suitable solution to overcome these challenges is to use the least possible gestures/postures for interactions. As interaction by the use of thumb is more engaging [24] and can be used for a more appealing 3D interaction system therefore, the GIST technique is proposed with the following key contributions:

1. Keeping a database of gestures, extraction of features, and comparison of gestures with stored templates are computationally costlier [25]. Moreover, the possibility of false-recognition of whole-hand gestures is high. The proposed technique introduces a gesture-free interaction system that works on the status of thumb instead of distinct whole-hand gestures.
2. With the contemporary gesture-based systems one needs to remember a number of hand gestures to be posed at run time. Cognitive load is involved in recalling and posing the exact gestures [26-28]. We believe that perceptive hand movements are suitable for interaction as no cognitive load is involved in recalling the gestures. For example, with less or no cognitive load a user may use forward hand movement for going inside a VE and vice versa.
3. Most of the contemporary gesture-based interaction systems utilize Machine Learning (ML) [29-30]. However, ML classifiers are data-hungry [30-31] and necessitate training of voluminous gestural dataset. Inspired from the pressing-of-button metaphor [24], we introduce thumb status based interaction. Without training of gestures and extraction of features, the technique simply counts thumb-up to activate an interaction at runtime.

The GIST technique privileges VR users to perform 3D interactions in a VE without caring for the predefined set of gestures. The basic interactions are performed in distinct states, where the status of thumb (thumb-up or thumb-down) is traced for the activation or deactivation of a state. In an attained state, interaction about an arbitrary axis is performed by the perceptive movement of a hand. Analogous to the button press metaphor, an interaction state is activated by the thumb-down posture. Interaction in an attained state is

performed by the horizontal, vertical or diagonal movement of a hand. The technique is implemented in a VS project; FIRST. The VE of FIRST is designed in OpenGL whereas the OpenCV library is used for the backend image processing. The system is evaluated by 12 users in a moderate lighting condition. Besides achieving the satisfactory accuracy; 87.5%, the evaluation revealed that the technique is suitable for interactions in VE.

2. Methodology

A potential solution to the challenges of gestural interfaces is to simply examine the status of thumb instead of complex gestures. With this research work, we introduce a gesture-independent interaction technique based on the perceptive movement of the hand in conjunction with the status of thumb. An Input image (I_{img}) scanned with an ordinary camera, is transformed to the YCrCb color space (Y_{img}). To get a compact skin mask image (M_{img}), the thresholding and closing operations are performed dynamically. With our designed *Sliding Scan* (SC) algorithm, the Tip-of-thumb Area (TA) is extracted from the initial M_{img}. The difference in M_{img} in the succeeding frames F_i and F_{i+1} is compared with the TA to get the status of thumb. The list of abbreviation in ascending with description is shown in Table 1.

Pre-processing is performed to target a hand in each image frame. Without training gestures or searching for specific features, the status of thumb is examined to activate/deactivate an interaction state. The algorithm; *Thumb Status based Switching* (TSS) performs state to state transition by detecting thumb-up or thumb-down in the succeeding input frames. Once an interaction state is activated, the *Centroid of Hand* (CH) is traced to perform interaction by the free 2D movement of hand.

Table 1. The list of abbreviation in ascending with description

Acronym	Word/Phrase
FIRST	Feasible Interaction by Recognizing the Status of Thumb
I_{img}	Input Image
CH	Centroid of Hand
TSS	Thumb Status based Switching
ROI	Region Of Interest
TT_{img}	Tip of Thumb image
TA	Tip-of-thumb Area
DA	Dynamic Area
TSS	Thumb Status based Switching
M_{img}	Mask Image
d_{img}	Difference image

A *Virtual Hand* (VH) represents the position of the user inside the VE. Coordinates mapping is performed dynamically to change the position of the VH by the CH dynamically. Framework of the proposed technique is shown in Fig. 1.

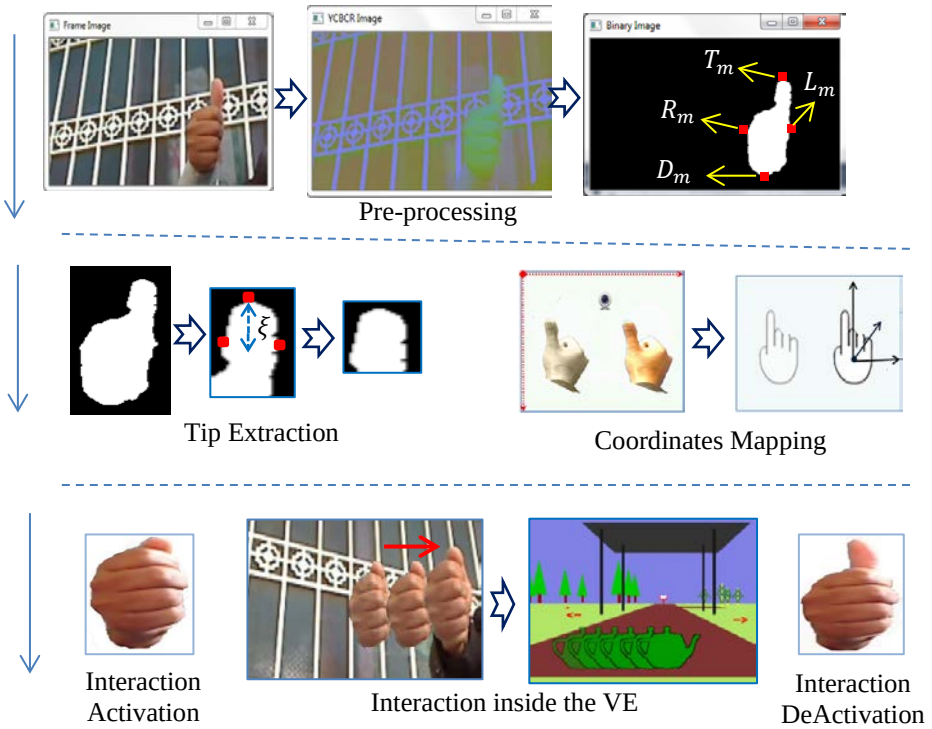


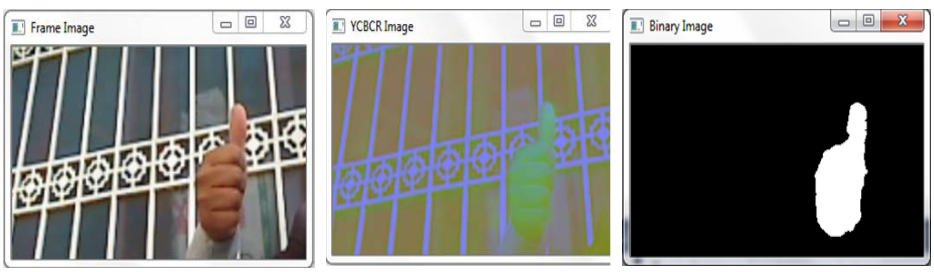
Figure 1. Framework of the GIST technique

2.1. Pre-processing

An input stream image scanned with an ordinary camera is converted into binary. As the YCbCr is the efficient colour model to distinguish the skin color from non-skin colours [32], therefore the YCbCr model is pursued for segmentation. The YCrCb representation (Y_{img}) of an input RGB image (I_{img}) is obtained with the most optimal range.

$$Y_{img}\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \frac{1}{256}\begin{bmatrix} 65.6 & 129 & 24.8 \\ -37.8 & -74.5 & 112.3 \\ 111.9 & -93.6 & -18.5 \end{bmatrix}\begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

The Y_{img} is then thresholded with the chrominance range suitable for skin colour [33] to get the skin mask image in binary (M_{img}), see Fig. 2.



(a) (b) (c)

Figure 2. The (a) I_{img} in RGB (b) the Y_{img} and (c) the M_{img}

$$M_{img} = \begin{cases} 1, & if\ 77 < Y_{img}.Cb < 127 \\ & \Lambda\ 133 < Y_{img}.Cr < 173 \\ 0, & Otherwise \end{cases}$$

For the sake of compactness, the morphological operation [34] by a structuring element $S_{(5\times5)}$ is performed to eliminate the unwanted white regions in the M_{img}.

$$M_{img} \circ S = (M_{img} ! S) \oplus M_{img} \quad (2)$$

The Region of Interest image (ROI) tightly enclosing the hand (a fist posture) is extracted from the M_{img} using our designed sliding scan algorithm [35] as,

$$ROI(m,n) = \begin{pmatrix} \cup_{r=D_m}^{T_m}(M_{img}), \\ \cup_{c=L_m}^{R_m}(M_{img}) \end{pmatrix} \quad (3)$$

where m and n represent the rows and columns of the ROI. The L_m, R_m, T_m and D_m represents the Left most, Right-most, Top-most and Down-most skin pixels of the M_{img}, see Fig. 3. The CH point is calculated as,

$$CH(x,y) = ((L_{m.x} + R_{m.x})/2\ , (T_{m.y} + D_{m.y})/2) \quad (4)$$

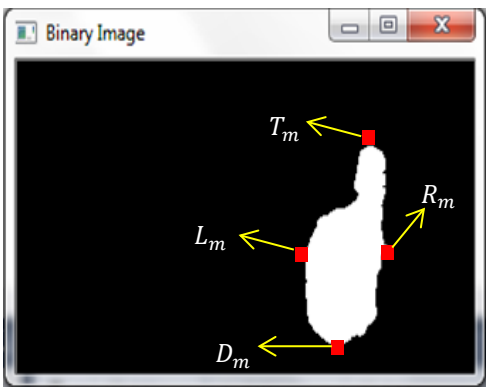


Figure 3. The M_{img} with the boundary pixels to extract ROI

2.2. Extracting the Tip of Thumb

The sliding scan algorithm [35] is followed to extract the Tip of Thumb image (TT_{img}). The TA is then computed once the TT_{img} is extracted in an initial frame. As the algorithm traces the TT_{img}, therefore the thumb status should be up (Thumb-up) in the initial frame. The sliding scan is performed from the top-left to the bottom-right to trace a T_m pixel. The empirical constant ξ is added with the position to have enough region beneath the T_m pixel.

$$D_m = T_m + \xi \quad (5)$$

The region (white/skin) enclosed by the L_m, R_m, T_m and D_m pixels, as shown in Fig. 4, is treated as TT_{img} for onward processing.

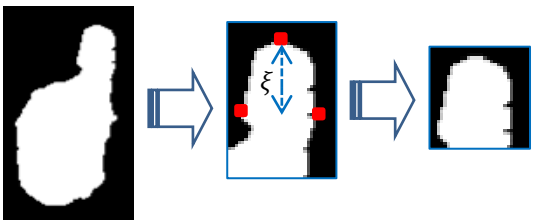


Figure 4. The extraction of the TT_{img} from the ROI

With the zeroth moment, the area (TA) of the TT_{img} is calculated as,

$$TA = \sum_{x=0}^{TT_{img.rows}} \sum_{y=0}^{TT_{img.columns}} TT_{img(x,y)} \quad (6)$$

Where x is the row and y the column position of a skin pixel in the TT_{img}.

2.3. Coordinates Mapping

Coordinates mapping between the CH in an image frame and the OpenGL window is performed for the seamless movement of the VH. The OpenCV coordinate system is different from the OpenGL coordinate system. The OpenCV frame starts with $O(0,0)$ from top left, whereas in OpenGL the origin $O(0,0,0)$ lies at the centre of the VE, see Fig. 5.

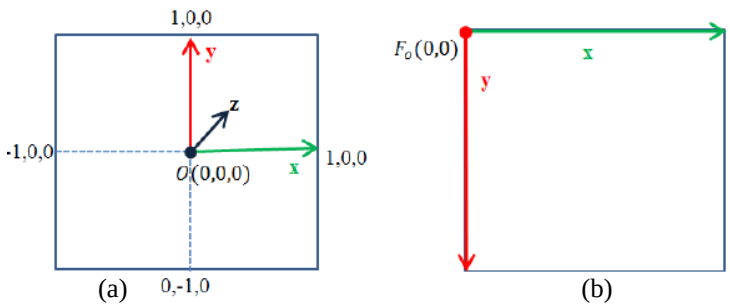


Figure 5. The coordinates of (a) OpenGL and (b) OpenCV

To harmonize the dissimilar coordinate systems, we devised our own mapping function, ω [36]. A scanned image frame is split into four regions R_1 to R_4 as shown in Fig. 6, where mapping is made by the corresponding function taking 'x' and 'y' of a pixel of R_n as independent variables. The point; CH is used to locate and move the VH in the designed VE.

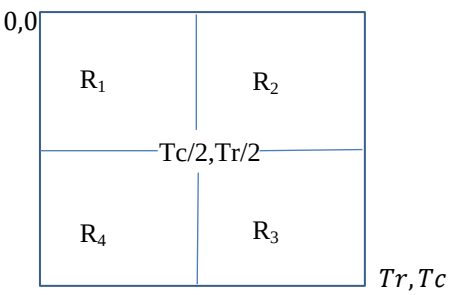


Figure 6. The virtual division of an image frame

If $CH_i \in \mathbb{R}^2$ represents the CH in a preceding frame and $CH_f \in \mathbb{R}^2$ represents the position of the CH in the following frame, then the virtual hand position; VH: $VH \in \mathbb{R}^3$ is calculated as,

$$VH(x,y) = \omega(CHi, CHf) \quad (7)$$

$$\omega(x,y) = ((\Delta Px/Tc)k, (\Delta Px/Tr)c) \quad (8)$$

$$\Delta Px = (CHd.x - CHi.x) \quad (9)$$

$$\Delta Py = (CHd.y - CHi.y) \quad (10)$$

Where *c* is the speed constant; greater the value of c speedier will be the movements of the VH. We kept a moderate value of c in the FIRST project. The value of *'Tc'* and *'Tr'* represents the total number of columns and rows respectively. The mapping process is shown in Fig. 7.

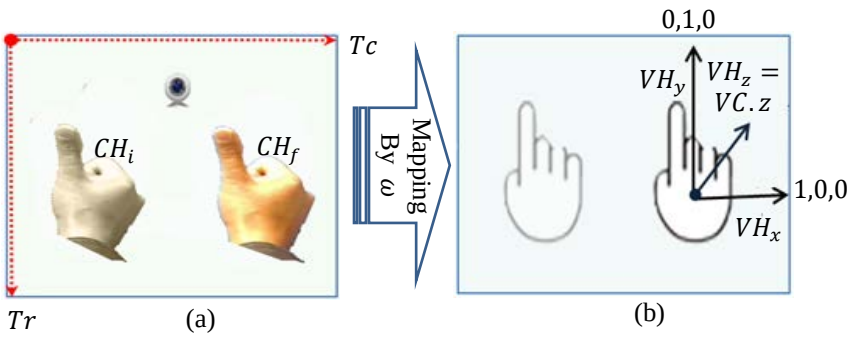


Figure 7. The mapping between (a) the CH and (b) the VH

2.3. States Switching

Instead of posing a distinct gesture for an interaction, the status of thumb (Thumb-down) is traced at the backend to activate a desired interaction state. At a time, one of the interaction state S_i gets activated where $i = \{0,1,2,3\}$. The up or down status of thumb is check by comparing the TA with the Dynamic Area (DA). The DA is the area of the difference image (d_{img}) obtained by subtracting the succeeding mask images. Let $Mp \in \mathbb{R}^2$ be the mask images in a preceding frame and $Mf \in \mathbb{R}^2$ the mask image in the following frame, then d_{img} is given as,

$$d_{img}(x,y) = Mf(x,y) - Mp(x,y) \quad (11)$$

As the normal *Frame Per Second* (FPS) rate is 24-30 frames, therefore, the d_{img} is computed after every 1/24 sec. The DA is obtained by adding the white pixels of the d_{img}.

$$DA = \sum_{x=0}^{w} \sum_{y=0}^{h} d_{img}(x,y) \quad (12)$$

The *w* and *h* represents the width and height of the d_{img} respectively.

On the basis of thumb Triggering (T_i), transition to a state S_i is performed where $i = \{0,1,2,3\}$. Besides changing the position of the VH, navigation is performed in the default state (S_0). To ensure whether thumb is up in an initial frame, the TA is checked to be within an empirical range $\beta_1 = \{100,800\}$. On the successful matching, the Mf is set as Mp. If the absolute difference between the TA and DA is in the second empirical range $\beta_2 = \{30,50\}$, it is assumed that the status of thumb has changed. A count variable;

C is incremented with each change in the status of thumb. An interaction state is activated with thumb down therefore, the odd value of C is used for the activation of an interaction. If C is even; the system switches back to the default state; S_0.

To perform state-to-state transition is a cyclic way, the entire process is repeated after the last state (at $T_i = 3$). The states of thumb are shown in Fig. 8 while schematic of the algorithm is shown in Fig. 9.

(a) (b)

Figure 8. The fist with (a) thumb-down and (b) thumb-up

2.4. Interaction Support

The basic 3D interactions; navigation, translation, scaling and rotation are performed by the free movement of hand. After switching into a particular state $S_i | S_i \neq S_0$, interactions are performed by tracing the 2D position of the CH. Each time, the position of the CH in a preceding frame (CH_P) is checked against the CH in the following frame (CH_f).

Tracing a change between the CH_p and CH_f about an axis, appropriate interaction is performed along that particular axis. An object is selected for manipulation by hovering the VH over an object at the time of thumb-down (at the time of initiating an interaction).

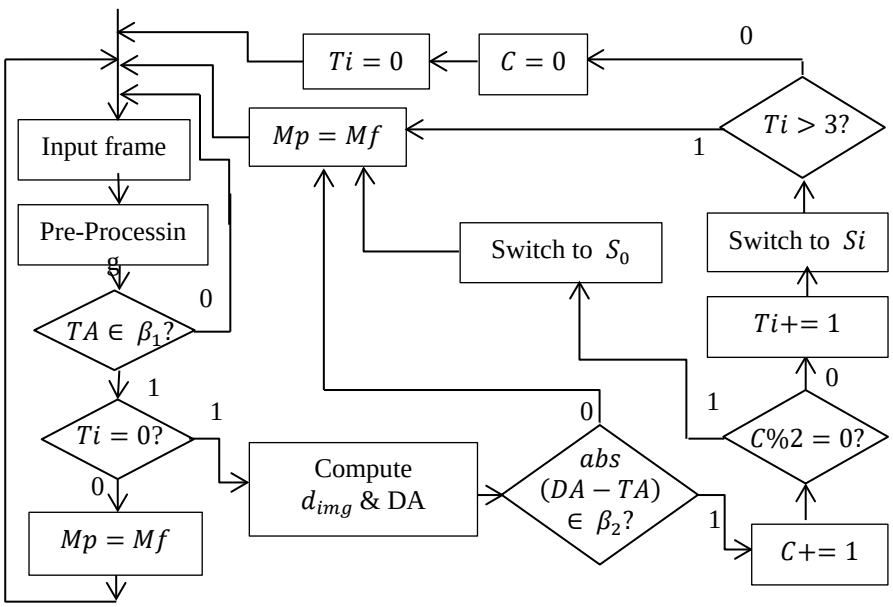


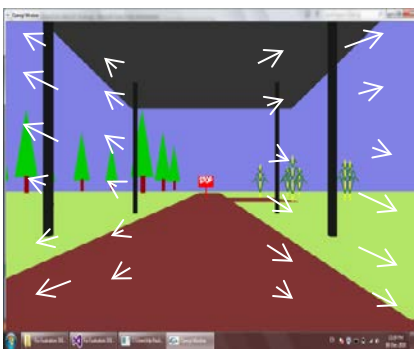
Figure 9. Schematic of the TSS algorithm

2.4.1 Navigation

Navigation is to explore a 3D VE. In the proposed technique, the diagonal movement of the hand in the upper-right (R_2) and lower-left (R_4) regions of the image frame perform navigation. As conceivable, the forward navigation is carried out by the upward-diagonal hand movement in the navigation region (R_2 and R_4) of an image frame, see Fig. 10. Navigation in the backward direction is performed by the downward-diagonal movement of the hand. Let CH_E be the position of CH at the time it enters into a navigation region (R_2 or R_4) and CH_d be the dynamic position of CH in the regions. The diagonal movements of hand are detected by the pseudo-code given as,

if $(CH_{d.x} > CH_{E.x}$ AND $CH_{d.y} <$
$CH_{E.y})$ AND !$\begin{pmatrix} (CH_{d.x} > CH_{E.x} \text{ AND } \text{abs}(CH_{d.y} - CH_{E.y}) \leq 5) \\ \text{OR}(|CH_{d.x} - CH_{E.x}| \leq 5 \text{ AND } CH_{d.y} < CH_{i.y}) \end{pmatrix}$
Forward Navigation
if $(CH_{d.x} < CH_{E.x}$ AND $CH_{d.y} > CH_{E.y})$ AND
!$\Big((CH_{d.x} > CH_{E.x}$ AND $(|CH_{d.y} - CH_{E.y}| \leq 5))OR\big((|CH_{d.x} - CH_{E.x} \leq$
$5|)$ AND $CH_{d.y} < CH_{E.y}\big)\Big)$

(a)



(b)

Figure 10. The (a) upward diagonal movement of hand for (b) forward navigation

2.4.2 Translation

Translation is to change the position of a virtual object along an arbitrary axis in a VE. With the first thumb-down; $C = 1$, $T_i = 1$, transition to S_1 is performed. In the state; S_1 an object is translated by the free movement of hand. With the horizontal and vertical movements of hand, the object is translated along the x and y axis (see Fig. 11). The upward and downward diagonal movements of hand translate the object along the –ve and +ve z-axis respectively. The pseudo-code used for the detection of the axis of translation in the S_1 is given as,

if $(\Delta(CH_{f.x}, CH_{p.x}) > 0$ AND $\Delta(CH_{f.y}, CH_{p.y}) = 0)$
Translate along x-axis
if $(\Delta(CH_{f.y}, CH_{p.y}) > 0$ AND $\Delta(CH_{f.x}, CH_{p.x}) = 0)$
Translate along y-axis
if $(CH_{f.x} > CH_{p.x}$ AND $CH_{f.y} < CH_{p.y})$ AND

$$!\Big((\mathrm{CH_{f.x}} > \mathrm{CH_{p.x}}\ \mathrm{AND}\ \mathrm{CH_{f.y}} = \mathrm{CH_{p.y}})\mathrm{OR}(\mathrm{CH_{f.x}} = \mathrm{CH_{p.x}}\ \mathrm{AND}\ \mathrm{CH_{f.y}} < \mathrm{CH_{p.y}})\Big)$$

Translate along the –z-axis

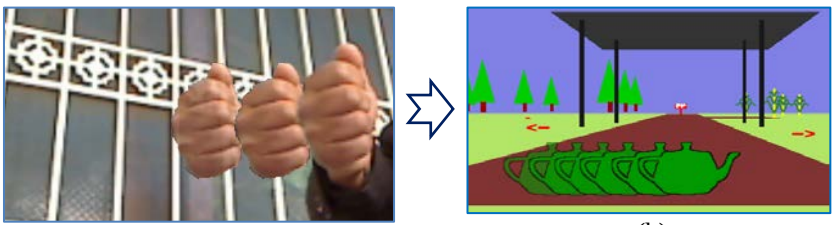
if $(\mathrm{CH_{f.x}} < \mathrm{CH_{p.x}}\ \mathrm{AND}\ \mathrm{CH_{f.y}} > \mathrm{CH_{p.y}})$ AND

$$!\Big((\mathrm{CH_{f.x}} > \mathrm{CH_{p.x}}\ \mathrm{AND}\ \mathrm{CH_{f.y}} = \mathrm{CH_{p.y}})\mathrm{OR}(\mathrm{CH_{f.x}} = \mathrm{CH_{p.x}}\ \mathrm{AND}\ \mathrm{CH_{f.y}} < \mathrm{CH_{p.y}})\Big)$$

Translate along the +z-axis

In the state (S_1), the VH is controlled by the CH whereas the position of the VH defines the dynamic positions of a selected object (Obj).

$$Obj(x,y,z) = \begin{bmatrix} 1 & 0 & VH.x \\ 0 & 1 & VH.y \\ 0 & 1 & VH.z \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} Objx \\ Objy \\ Objz \\ 1 \end{bmatrix} \quad (13)$$



(a) (b)

Figure 11. The (a) horizontal movement of hand to (b) translated the object along x-axis

2.4.3 Scaling

The size of a 3D object is increased (scale up) or decreased (scale down) along an arbitrary axis in the state; S_2. In the state (S_2), a selected object is scaled or scale-down by comparing the positions of the $\mathrm{CH_p}$ with the $\mathrm{CH_f}$. Scaling (up scaling) about the x or y axis is performed by tracing the hand movement along the *+ve* x or y axis. Similarly, downscaling is performed with the hand movement along the *–ve* x or y axis. The diagonal movements of hand scale an object up or down about the z-axis. The Euclidean distance (Ed) between the $\mathrm{CH_p}$ and $\mathrm{CH_f}$ is computed to find out the scaling factor.

$$Ed_x = \sqrt{(CH_{f.x} - CH_{p.x})^2} \quad (14)$$

$$Ed_y = \sqrt{(CH_{f.y} - CH_{p.y})^2} \quad (15)$$

$$Ed_z = \sqrt{(CH_{f.x} - CH_{p.x})^2 + (CH_{f.y} - CH_{p.y})^2} \quad (16)$$

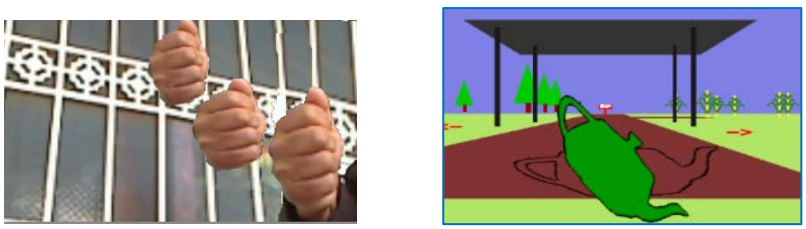
$$\begin{bmatrix} \boldsymbol{Objx} \\ \boldsymbol{Objy} \\ \boldsymbol{Objz} \\ \boldsymbol{1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Ed_x} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Ed_y} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Ed_z} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} \end{bmatrix} \times \begin{bmatrix} \boldsymbol{Objx} \\ \boldsymbol{Objy} \\ \boldsymbol{Objz} \\ \boldsymbol{1} \end{bmatrix} \quad (17)$$

2.4.3 Rotation

Rotation is to turn an object about an axis inside a VE. In the last state; S_3, the horizontal,

vertical and diagonal movements of the hand performs rotation along the y-axis, x-axis and z-axis respectively. The angle θ of rotation is computed from the dynamic distance (d) between the points CH_p and the CH_f. Greater the distance, larger will be the angle of rotation. The clockwise rotation is shown in Fig. 12.

$$d = \sqrt{(CH_{f.x} - CH_{p.x})^2 + (CH_{p.x} - CH_{f.y})^2} \quad (18)$$



(a) (b)

Figure 12. The (a) diagonal movement of hand to (b) rotate the object clockwise

The pseudo-code for the axis for rotation is given as follow,

if $(|CH_{f.y} - CH_{p.y}| \leq 5)$
    if $(CH_{f.x} > CH_{p.x})$
        Rotate along y-axis
        $\theta = +d$
    if $(CH_{f.x} < CH_{p.x})$
        Rotate along y-axis
        $\theta = -d$
if $(|CH_{f.x} - CH_{p.x}| \leq 5)$
    if $(CH_{f.y} > CH_{p.y})$
        Rotate along x-axis
        $\theta = +d$
if $(CH_{f.y} < CH_{p.y})$
        Rotate along x-axis
        $\theta = -d$
if $(CH_{f.x} > CH_{p.x}$ AND $CH_{f.y} < CH_{p.y})$ AND
    $!\big((CH_{f.x} > CH_{p.x}$ AND $CH_{f.y} = CH_{p.y})OR(CH_{f.x} = CH_{p.x}$ AND $CH_{f.y} < CH_{p.y})\big)$
        Clockwise rotation
if $(CH_{f.x} < CH_{p.x}$ AND $CH_{f.y} > CH_{p.y})$ AND $!\big((CH_{f.x} > CH_{p.x}$ AND $CH_{f.y} = CH_{p.y})OR(CH_{f.x} = CH_{p.x}$ AND $CH_{f.y} < CH_{p.y})\big)$
        Anticlockwise rotation

3. Implementation and Evaluation

The proposed technique is implemented in a case-study application. A Corei7 laptop with 3.10 GHz processor and 8GB RAM with HD graphics card was used for the implementation and evaluation. The OpenGL and OpenCV were used for the front-end VE and for the back end image processing. Offering the first person's view, the position of

a user in the VE is represented by the VH. The z-axis of the virtual camera is assigned to the VH.z in order to keep the VH visible everywhere in the VE. The system is activated by detecting the thumb area in the specified range (β_1) in a frame. With the detection of a fist with thumb up, the text *"DETECTED"* appears in the upper part of the scene. A user is constantly informed about the states transitions with the help of text and distinguishing audio signals (beeps). For easy noticing, end-point of the scene is marked by a board with text 'Stop'. To immerse users in the VE, different 3D objects are rendered at different points. The system resets by one time pressing of the Enter key and quits with the pressing of the Escape key. The designed VE is shown in Fig. 13.

Figure 13. The virtual scene for evaluation of the technique

3.1. Evaluation Setup

The evaluation was performed in the University IT lab in a moderate lighting condition. Twelve participants, all male, ages 22-47 ($\text{mean} = 31.8, \text{SD} = 5.0$), 11 right-handed and 1 left-handed performed five tasks, where each task evaluated the basic interactions. Participants were introduced to the system and were guided on how to perform the predefined tasks in the designed 3D VE. All the participants performed pre-trials of the tasks before the actual evaluation. The tasks were to evaluate the basic interactions; Navigation, Translation, Scaling and Rotation. A 3D object (*Teapot*) is rendered in the mid of the scene for selection and manipulation. Each participant performed two trials of the following five tasks.

Task-1: Perform forward navigate till the end point.
Task-2: Perform reverse/backward navigation till the starting point.
Task-3: Translate the teapot and place it on a table.
Task-4: Scale (scale up) the teapot about an axis and then scale it down.
Task-5: Rotate the teapot about an axis.

In a single trial, navigation is assessed two times while translation, rotation and scaling one time each. With the mentioned setup, an overall accuracy rate achieved for the 120 interaction attempts is shown in Fig. 14, is 87.5%. Inappropriate interaction after a required posture and false detection were counted as errors.
$Accuracy =$

$$(No.of\ Interaction\ Performed/Total\ No.of\ Interaction\ attempts) \times 100 \quad (19)$$

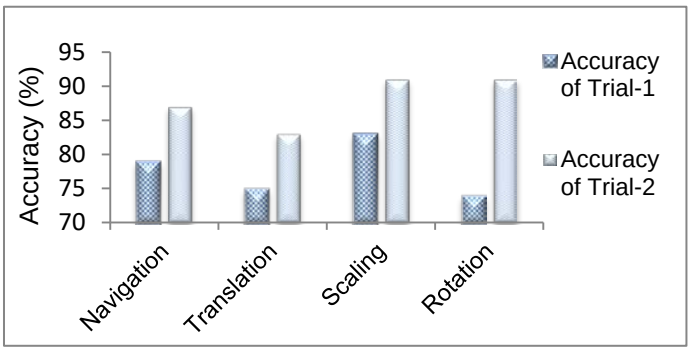


Figure 14. The mean accuracy (%) of the interactions performed

3.2. Learning Effect

The learning effect was measured from the errors occurrence rate. The paired two-sample T-test was used to analyze the differences in means of the two trails. With null hypothesis (H_0) we assumed that mean difference (μ_d) is 0. The hypothesis was rejected as there was a significant difference between the outcomes of Trail-1 ($\mu_1 = 82.2, SD = 1.8$) and Trail-2 (μ_2 =92.7, SD= 4.5) conditions; ($t(6) = 4.3, p = 0.0049$).

$$\mu_i = \sum_{j=1}^{n} x_j / n \quad (20)$$

$$SD = \sqrt{\frac{\sum_{j=1}^{n}(x_j - \mu_i)^2}{n-1}} \quad (21)$$

Where $i = \{1,2\}$ and n represents the total number of interaction (x_j).

The graph indicating this vivid decrease in error is shown in Fig. 15. While performing navigation, some of the users moved their hand faster. In such cases the position of the MH was wrongly traced by the camera, hence, comparatively more errors were counted during navigation.

3.3. Subjective Analysis

At the end of the evaluation session, the users were asked to fill up a questionnaire gauging the three factors; *Ease of Use*, *Fatigue* and *Suitability* in a VE.

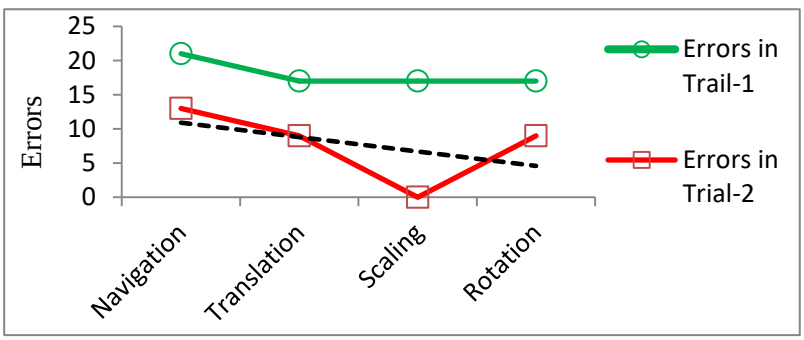


Figure 15. The number of errors (%) occurred in Trial-1 and Tiral-2

Most of the participants were in favor of the GIST technique; strongly agreed= 77%, agreed=19.6%. The participant's response to the three factors as calculated by the following formula is shown in Fig. 16.

$$Option\ Response(\%) = (No.of\ Users\ Opted\ for\ the\ option/Total\ No.of\ Users) \times 100 \quad (22)$$

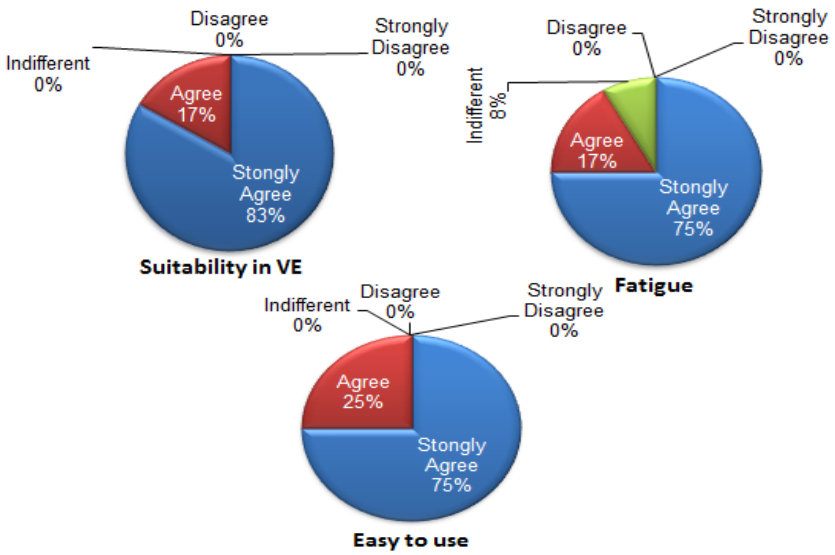


Figure 16. Outcomes of the subjective analysis about the three factors

4. Discussion

Although, gesture-based interactions are suitable for the interaction in a VE [37], such systems are difficult to design [37]. For a novice VR user, it is often difficult to learn and remember different gestures for interactions [21,38]. Considerable research has been carried out to use such gestures which are feasible to pose and reliable to recognize [26]. Games like StrikeAPose [39] have also been designed to explore feasible gestures for interactions. Despite the fact that accuracy can be improved by user-independent gestures [40,41], there is no agreed-upon standard list of gestures for 3D interactions. Moreover, the hand size and finger length vary from individual to individual. To overcome most of the gesture-related challenges, a suitable solution is to let users perform interactions by the simple movement of a hand. This will avoid the user's side difficulty to learn and remember the gestures set by others. Like the pressing of a button, in the proposed technique an interaction is activated by the thumb down. In an attained state, interaction along the x, y and z-axis is performed by the perceptive horizontal, vertical and diagonal movements of a hand. Switching to the default state, an interaction is deactivated by the raising of the thumb. This research work is a step forward to make 3D interaction simpler and feasible. Unlike the costlier and complex setup of data gloves [42] and Myo armbands [43], an ordinary camera is used for the detection of hand and thumb. The status of thumb is detected by the cost-effective method of frame-to-frame variation [44]. Outcomes of the technique

support applicability the technique in the VR applications. During the evaluation, it was observed that most of the errors were due to the quicker movements of the user's hand. The accuracy can be raised with high quality camera with a high frame rate.

5. Conclusion and Future Work

To cope with the rampant pace of VR developments, a simple and natural interface is needed for intuitive 3D interactions. Though a gestural interface may enhance the realism of a VE, the interclass gesture dissimilarities are the main issue. It is rare that two individuals may make the same gesture in the same way. Moreover, it is not possible for an individual to pose a specific gesture twice with the same accuracy. With this contribution, we present a gesture independent interaction technique. Despite the extraction and comparison of gesture-specific features, interactions are performed in distinct states. A user may activate or deactivate an interaction state by posing a feasible thumb posture (thumb-down or thumb-up). In an interaction state, interaction tasks inside a VE are performed by the perceptive movements of the hand. The technique is implemented and evaluated in a case-study project where a satisfactory accuracy of 87.5% was achieved. The proposed technique is suitable in a wide spectrum of man-machine interactions particularly in virtual prototyping, 3D gaming, robotics and simulation. The work also covers the smooth integration of image processing and VE. With less effort, the technique can be implemented on other sensing platforms. In future, we are determined to enhance the system for collaborative VE.

Acknowledgements

We are thankful to the staff and students of the University of Malakand, Pakistan.

Conflicts of Interest

The authors declare that there is no conflict of interest.

References

[1] Holl M, Oberweger M, Arth C. and Lepetit V. (2018) Efficient Physics-Based Implementation for Realistic Hand-Object Interaction in Virtual Reality. *In IEEE Conference on Virtual Reality and 3D User Interfaces*, Reutlingen, Germany, 18-22 March 2018, 175-182. http://dx.doi.org/10.1109/VR.2018.8448284

[2] Ghotkar A.S, Kharate G.K. (2012) Hand segmentation techniques to hand gesture recognition for natural human computer interaction. *International Journal of Human Computer Interaction (IJHCI)*. 3(1),15. http://dx.doi.org/10.1109/ICIIP.2011.6108940

[3] Shamaie A and Sutherland A. (2013) Accurate recognition of large number of hand gestures. K.N. Toosi University of Technology, 308-317.

[4] Kiyokawa K, Takemura H, Katayama Y, Iwasa H, Yokoya and N. (1996) Vlego: A simple two-handed modeling environment based on toy blocks. Based on Toy Blocks, *In proceedings of the ACM Symposium on Virtual Reality Software and Technology*, 27-34. http://dx.doi.org/10.1145/3304181.3304189

[5] Hua J, Qin H. (2001) Haptic sculpting of volumetric implicit functions. In *Proceedings of the Ninth Pacific Conference on Computer Graphics and Applications*, Tokyo, Japan, 16-18 Oct. 2001, 254-264. http://dx.doi.org/10.1109/PCCGA.2001.962881

[6] Wingrave C.A, Williamson B, Varcholik P.D, Rose J, Miller A, Charbonneau E, Bott J, LaViola Jr J. (2010) The wiimote and beyond: Spatially convenient devices for 3d user interfaces. *IEEE Comp Graph and App*; 1(2), 71-85. https://doi.org/10.1109/MCG.2009.109

[7] Chun J, Lee B. (2010) Dynamic Manipulation of a Virtual Object in Marker-less AR system Based on Both Human Hands. *KSII Transactions on Internet & Information Systems*, 1; 4(4).

https://doi.org/10.3837/tiis.2010.08.010

[8] Buchmann V, Violich S, Billinghurst M, Cockburn A. (2004) FingARtips: gesture based direct manipulation in Augmented Reality. *In Proceedings of the 2nd international conference on Computer graphics and interactive techniques*, Australasia and South East Asia, Jun 15 2004, 212-221. https://doi.org/10.1145/988834.988871

[9] Song S, Goh P, Hutama, WB, Fu, W and Liu, X. (2012) A handle bar metaphor for virtual object manipulation with mid-air interaction. *In Proceedings of SIGCHI Conference on human factors in Computing Systems*, Austin, Texas, USA, 5-10 May 2012, pp. 1297–1306. http://dx.doi.org/ 10.1145/159544.159562

[10] Höll M, Oberweger M, Arth C and Lepetit V. (2018) Efficient Physics-Based Implementation for Realistic Hand-Object Interaction in Virtual Reality. *In IEEE Conference on Virtual Reality and 3D User Interfaces*, Reutlingen Germany, 18-22 March 2018, 175-182. https://doi.org/10.1145/988834.988871

[11] Guna J, Jakus G, Pogačnik M, Tomažič S and 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), 3702-20. https://doi.org/10.3390/s140203702

[12] Zimmermann C and Brox T. (2017) Learning to estimate 3d hand pose from single rgb images. *In IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4913-4921. https://doi.org/10.1109/ICCV.2017.525

[13] Oberweger M, Wohlhart P and Lepetit V. (2015) Hands deep in deep learning for hand pose estimation, arXiv preprint arXiv:1502.06807, 4903-4911. https://doi.org/10.1109/ ICCV.2015.100

[14] Moehring M and Froehlich B. (2011) Effective manipulation of virtual objects within arm's reach. *InVirtual Reality Conference (VR)*, Singapore, 19-23 March 2011, 131-138. http://dx.doi.org/10.1109/VR.2011.5759451

[15] Yim D, Loison GN, Fard FH, Chan E, McAllister A and Maurer F. (2016) Gesture-driven interactions on a virtual hologram in mixed reality. *In Proceedings of the ACM Companion on Interactive Surfaces and Spaces*, Niagara Falls, Ontario, Canada, November 06 - 09 2016, pp. 55-61. https://doi.org/10.1145/3009939.3009948

[16] Prachyabrued M and Borst C W. (2012) Virtual grasp release method and evaluation. *International Journal of Human-Computer Studies*, 70(11), 828-48. https://doi.org/10.1016/j.ijhcs.2012.06.002

[17] Rijpkema H and Girard M. (1991) Computer animation of knowledge-based human grasping. *In ACM Siggraph Computer Graphics*, 1991, 25(4), 339-348. https://doi.org/10.1145/122718.122754

[18] Zhang Z. (2012) Microsoft kinect sensor and its effect. IEEE multimedia, 19(2), 4-10. https://doi.org/10.1109/MMUL.2012.24

[19] Frank W, Bachmann D, Rudak B and Fisseler D. (2013) Analysis of the accuracy and robustness of the leap motion controller, *Sensors*, 13(5), 6380–6393, 2013. http://dx.doi.org/10.3390/s130506380

[20] Benko H. (2009) Beyond flat surface computing: challenges of depth-aware and curved interfaces, *In Proceedings of the 17th ACM International Conference Multimedia*, Vancouver, British Columbia, Canada, October 19-24, 2009, 935–944. http://dx.doi.org/ 10.1145/1631272.1631462

[21] Vanacken D, Beznosyk A and Coninx K. (2014) Help systems for gestural interfaces and their effect on collaboration and communication, *In Workshop on gesture-based interaction design: communication and cognition*. http://dx.doi.org/ 10.1.1.710.8679

[22] Smedt Q. (2017) Dynamic hand gesture recognition-From traditional handcrafted to recent deep learning approaches, Doctoral dissertation, Université de Lille 1, Sciences et Technologies; CRIStAL UMR 9189, https://hal.archives-ouvertes.fr/tel-01691715/document

[23] Bejan A, Wieland M, Murko P and Kunze C. A (2018) Virtual Environment Gesture Interaction System for People with Dementia. *In Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, Hong Kong, June 9-13 2018, 225-230. http://dx.doi.org/10.1145/3197391.3205440

[24] Holzer A, Vozniuk A, Bendahan S and Gillet D. (2016) Rule of thumb: effect of social button icons on interaction. *In Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, 659-666. https://doi.org/10.1145/2957265.2961842

[25] Choi J H, Ko N Y and Ko D Y. (2001). Morphological gesture recognition algorithm. *In Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology, Cat. No. 01*, vol. 1, 291-296.

[26] Ackad C, Kay J and Tomitsch M. (2014). Towards learnable gestures for exploring hierarchical information spaces at a large public display, *In CHI Workshop on Gesture-based Interaction Design*, *57*. http://dx.doi.org/ 10.1145/2669485.2670531

[27] Card S K. (2014). A simple universal gesture scheme for user interfaces. *In Gesture-Based Interaction Design: Communication and Cognition, CHI 2014 Workshop*.

[28] Pham H A. (2018). The challenge of hand gesture interaction in the Virtual Reality Environment: evaluation of in-air hand gesture using the Leap Motion Controller.

[29] Dardas N H and Georganas N D. (2011). Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. *IEEE Transactions on Instrumentation and measurement*, 60 (11), 3592-3607. https://doi.org/10.1109/TIM.2011.2161140

[30] Zhang C, Yang X and Tian Y. (2013). Histogram of 3D facets: A characteristic descriptor for hand gesture recognition. *In 2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, 1-8. https://doi.org/10.1109/FG.2013.6553754

[31] Asadi-Aghbolaghi M, Clapes A, Bellantonio M, Escalante H J, Ponce-López V, Baró X and Escalera S. (2017). A survey on deep learning based approaches for action and gesture recognition in image sequences. *In 12th IEEE international conference on automatic face & gesture recognition, (FG 2017)*, 476-483. https://doi.org/10.1109/FG.2017.150

[32] Phung S L, Bouzerdoum A and Chai D. (2002) A novel skin color model in ycbcr color space and its application to human face detection, *In Proceedings of International Conference on Image*, 1,

I-I. https://doi.org/10.1109/ICIP.2002.1038016

[33] Maheswari, S and Korah, R. (2017) Enhanced skin tone detection using heuristic thresholding, *Biomedical Research*, 28(9), 29-35. https://doi.org/10.11648/j.ajai.20170101.14

[34] Sreedhar, K. and Panlal, B. (2012) Enhancement of images using morphological transformation. arXiv preprint arXiv:1203.2514. https://doi.org/10.5121/ijcsit.2012.4103

[35] Raees, M., Ullah, S, Rahman, S U and Rabbi, I. (2016) Image based recognition of Pakistan sign language, *Journal of Engineering Research* , 4(1), 1-21. https://doi.org/10.7603/s40632-016-0002-6

[36] Raees, M., Ullah, S., and Rahman, S. U. (2018). VEN-3DVE: vision based egocentric navigation for 3D virtual environments, *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 1-11. https://doi.org/10.1007/s12008-018-0481-9

[37] Tversky, B., Jamalian, A., Segal, A., Giardino, V. and Kang, S. M. (2014) Congruent gestures can promote thought, *In Gesture-Based Interaction Design: Communication and Cognition, CHI Workshop*. http://dx.doi.org/ 10.1186/s41235-016-0004-9

[38] Card, S. K. (2014) A simple universal gesture scheme for user interfaces, *In Gesture-Based Interaction Design: Communication and Cognition, CHI Workshop*. http://dx.doi.org/ 10.1007/s00779-013-0725-42013

[39] Walter, R., Bailly, G. and Müller, J. (2013) Strikeapose: revealing mid-air gestures on public displays, *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 841–850. http://dx.doi.org/10.1145/2470654.2470774

[40] Hespanhol, L., Tomitsch, M., Grace, K., Collins, A. and Kay, J. (2012) Investigating intuitiveness and effectiveness of gestures for free spatial interaction with large displays, In Proceedings of the International Symposium on Pervasive Displays, 6. http://dx.doi.org/10.1145/2307798.2307804

[41] Diman Zad, T., Sampson, A., Mytkowicz, T and McKinley, K S. (2017) High Five: Improving Gesture Recognition by Embracing Uncertainty, arXiv preprint arXiv:1710.09441. https://arxiv.org/pdf/1710.09441

[42] Kerber, F, Puhl, M and Kruger A. (2017) User-independent real-time hand gesture recognition based on surface electromyography," In Proceedings of the 9th Int. Conf. Human Comp. Inter. with Mobile Devices & Services, Vienna, Austria, September 04-07 2017, 36. http://dx.doi.org/ 10.1007/s10462-012-9356-9

[43] Weissmann, J and Salomon, R. (1999) Gesture recognition for virtual reality applications using data gloves and neural networks, *In International Joint Conference on Neural Networks (IJCNN)*, Washington, DC, USA, 10-16 July 1999, 2043–2046. http://dx.doi.org/ 10.1109/IJCNN.1999.832699

[44] Ullah A, Muhammad K., Del Ser, J, Baik S W and Albuquerque V. (2018) Activity recognition using temporal optical flow convolutional features and multi-layer LSTM. *IEEE Transactions on Industrial Electronics*. 9692-9702. http://dx.doi.org/ 10.1109/TIE.2018.2881943