Two Hand Gesture Based 3D Navigation in Virtual Environments

I. Rehman*, S. Ullah, M. Raees

University of Malakand, Department of CS & IT, Chakdara (Pakistan)

Received 30 April 2018 | Accepted 22 June 2018 | Published 6 July 2018

Abstract

Natural interaction is gaining popularity due to its simple, attractive, and realistic nature, which realizes direct Human Computer Interaction (HCI). In this paper, we presented a novel two hand gesture based interaction technique for 3 dimensional (3D) navigation in Virtual Environments (VEs). The system used computer vision techniques for the detection of hand gestures (colored thumbs) from real scene and performed different navigation (forward, backward, up, down, left, and right) tasks in the VE. The proposed technique also allows users to efficiently control speed during navigation. The proposed technique is implemented via a VE for experimental purposes. Forty (40) participants performed the experimental study. Experiments revealed that the proposed technique is feasible, easy to learn and use, having less cognitive load on users. Finally gesture recognition engines were used to assess the accuracy and performance of the proposed gestures. kNN achieved high accuracy rates (95.7%) as compared to SVM (95.3%). kNN also has high performance rates in terms of training time (3.16 secs) and prediction speed (6600 obs/sec) as compared to SVM with 6.40 secs and 2900 obs/sec.

Keywords

Human Computer Interaction, Virtual Reality, Augmented Reality, Gesture Recognition, 3D Navigation.

I. Introduction

The effective and realistic interaction is a prerequisite to attain/explore physical/behavioral properties of a Virtual Environment (VE) and its constituent objects. The interaction in VEs ranges from keyboard and mouse where button clicks (key/mouse) carry no information [1], to touch based, which uses 2D gestures with finger/hand actions via touchscreens/interactive panels, and recently to touchless (natural) interfaces which involves 3D gestures in midair [1]. Hand gesture recognition plays a vital role in various applications such as sign language recognition [2-5], robotics [11-13], physical (health) sciences [14-16], natural sciences [17], computer engineering [18, 19], and industrial areas [20, 21] etc.

Gesture based interaction is the most attractive form of natural interaction due to the exclusion of physical contact with hardware where input/interaction devices need to be permanently connected (via some physical means) with computer. These physical media (cables) are used for delivery of input to the computer. These media lead to extra burden, need space, extra cost and complexity. Gesture based interaction offers more natural and intuitive HCI with various multimodal forms [22-26].

Different sensors have been used in computer vision and image processing for recognition purposes, such as Koller et al. [27], who used monocular camera for tracking in augmented reality applications. Jalal et al., presented different systems for Human Activity Recognition (HAR) based on depth video [28, 29] and depth imaging [30, 31]. Some authors proposed, Depth Images-based Human Detection and Activity Recognition [32], and Human Pose Estimation and Recognition from RGB-D Video [33].

Feature selection has a vital role in any recognition system. For face recognition different methods have been used. In Holistic Matching method [34, 35], the complete face area is taken as input to the recognition system. In Feature based method, the position and statics of nose, eyes and mouth are considered as input to the system. The Hybrid method uses the combination of both Holistic and feature based methods [36].

For action recognition, Lowe [37] introduced Scale Invariant Feature Transform (SIFT). Dense Sampling Scale Invariant Feature Transform (DSIFT) was used by different authors for action recognition [38-48]. Histogram of oriented gradient (HOG) was used by [41, 47-55]. Shape context (SC) was proposed by Belongie and Malik [56] for feature extraction and also was used by Wang et al. [57], Gupta et al. [51], and Yao and Fei-Fei [58]. For recognition of action from still images, GIST was proposed by Oliva and Torralba [59] and used by Gupta et al. [51], Prest et al. [60], and Li and Ma [48]. The Speeded Up Robust Features (SURF), is proposed by Bay et al. [61] and used by Ikizler et al. [62] to represent the human silhouettes for action recognition.

Different techniques are proposed by various authors for gait recognition. Ahmed et al. [63], used horizontal and vertical distances of selected joint pairs. Andersson et al. [64], calculated mean and standard deviation in the signals of lower joint angles. Ball et al. [65] used mean, standard deviation and maximum of the signals of lower joint angles. Dikovski et al. [66] proposed a set of seven different features, such as joint angles and inter-joint distances aggregated within a gait cycle, body parameters, along with various statistics. Kwolek et al. [67], used static body parameter, bone rotations, and the person’s height. Preis et al. [68], presented 13 pose attributes. Sinha et al. [8], used multiple gait features, upper and lower body area, inter-joint distances and other

* Corresponding author.
E-mail address: inam.btk@gmail.com
Different approaches are proposed for hand recognition such as mount based sensors [71], multi-touch screen sensors [72, 73] and vision based sensors [74-76]. The depth based hand gesture recognition have three types i.e. static hand gesture recognition [77], hand trajectory gesture recognition [78, 79] and continuous hand gesture recognition [80, 81]. Most of the authors used computer vision and image processing methods [82-85], along with some newly introduced input devices such as Leap Motion [86-88] and Kinect [89]. For a natural interaction with AR environments, fiducial markers are used with fingers [90, 91]. Different computer vision techniques are used for detection of hand and fingertips for AR interaction [92-94]. These systems are commercially limited due to problems such as skin color and precise depth sense [94]. Different glove based techniques have been used for accurate interaction [95-97] but limited due to its cumbersome nature.

For gesture based navigation, different systems have been proposed so far, but have limited commercial application due to cumbersome or inaccurate nature, cost, or dependency/need for special devices and their limited range. Recent research mostly stresses to deal with these problems, but simple and intuitive interaction is still the major area than needs to be improved. The previous gesture based navigation techniques, mostly rely on the coarse/unrealistic alteration/shape of hands and fingers layout for transition among different gestures which results in an increased physical and mental load on the user.

In this paper, we propose a novel two hand gesture based 3D navigation technique for VEs with the objective of providing intuitive and easy navigation. Navigation includes 3D movement, i.e. forward, backward, up, down, left, and right along with an effective speed control mechanism. Computer vision techniques are used for detection of gestures (colored thumb, fingers) from the real scene while OpenGl is used as a front end for navigation in the VE. Machine learning tools such as SVM and kNN are used to assess the accuracy and performance of the proposed gestures.

The rest of the paper is organized as follows: section 2 presents related work, section 3 describes the proposed system, section 4 consists of experiments and evaluation and finally section 5 is related to conclusion and future work.

II. RELATED WORK

In daily life communication, hand gestures cover the gap of merely verbal information, and so is a necessary part of effective and meaningful communication with the receiver. In HCI, hand gesture based interaction is more valuable due its natural and attractive nature. Hand gestures refer to meaningful movement of the hand and fingers [86], which entails most valuable information [98].

In the past different navigation techniques in VEs have been proposed [99, 100]. Different types of sensing techniques have been used for recognition of patterns in gestures [101]. Different sensing systems have been proposed so far, such as glove based, vision based, along with some newly introduced devices such Leap Motion [102] and Kinect [89].

Glove based devices use movement based approach with high performance in some applications such as recognition of sign language [103]. CyberGlove, a type of data gloves is used for tracking hand gestures [104]. Cooper et al. [105] used color coded glove for tracking of hand movement, but the system needs wearable gloves which decrease user’s experience in the environment. Kim et al. [106], used a Cyber Glove, a wearable device for recognition of hand gestures and performed different navigation tasks as shown in Fig. 1.

Although wearable devices were used mostly for gesture based interaction in the past, the cumbersome and costly nature of gloves, limit its widespread use in HCI [101].

Chen et al. [107], used computer vision for the detection of hand gestures from the video taken by a webcam. Two types of hand gestures were proposed i.e. appearance based which used bare hand and marker based (with colored markers on a black glove) 3D hand model. For moving a virtual car, different gestures were proposed as shown in Fig. 2. The system is unable to provide complete 3D movement and speed control. The system has less degree of correspondence with real world navigation.

Krum et al. [108], presented a navigation interface (earth 3D visualization) using verbal and hand gestures. The system used image processing techniques for detection of hand gestures taken from a Gesture Pendant video camera. Multiple infrared emitting LEDs were used for illuminating the hand gestures in front of camera. Different types of hand gestures were used for navigation.

Shao Lin [109], used Leap Motion controller for detection of hand gestures. The proposed technique used both hands for three different types of gestures as shown in Fig. 3. Similarly clockwise and anti-clockwise rotation of hands produced the same rotation in the VE but the technique has no mechanism for up, down movements and speed control.

Kerefeyn et al. [110], used different gestures for controlling and manipulation of virtual objects in a VE using Leap Motion controller. The system proposed five different gestures for interaction with virtual objects using right hand (see Fig. 4). The system does not provide complete 3D navigation i.e. forward, backward, up, down, left, and right as well as there is no mechanism for speed control.

Khundam et al. [111], proposed single hand gestures via Leap Motion controller for navigation in VE as shown in Fig. 5. Forward movement is done by raising the hand with straight palm. While backward movement is achieved via turning palm direction or reversing hand facing. Pushing the palm to left causes step movement to left side while its movement to right results in step right. Grasping or moving out of display area causes hold position. Forward speed depends on advance movement of palm while pulling of palm to body controls backward speed. More or less movement towards left or right side causes more or less speed. The proposed gestures are hard to learn.
Liang et al. [113] presented a system using Leap Motion controller for the detection of hand gestures. There were different types of modules for navigation such as single hand gestures, designed for children which causes avatar in VE to fly left, right, up, and down as shown in Fig. 7.

<table>
<thead>
<tr>
<th>Hand gesture</th>
<th>Movement</th>
<th>Target action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move right</td>
<td>Fly to the right</td>
<td></td>
</tr>
<tr>
<td>Move left</td>
<td>Fly to the left</td>
<td></td>
</tr>
<tr>
<td>Move down</td>
<td>Fly down</td>
<td></td>
</tr>
<tr>
<td>Move up</td>
<td>Fly up</td>
<td></td>
</tr>
<tr>
<td>Stretch</td>
<td>Hover</td>
<td></td>
</tr>
<tr>
<td>Stretch to grip</td>
<td>Grasp stick</td>
<td></td>
</tr>
<tr>
<td>Grip to stretch</td>
<td>Drop stick</td>
<td></td>
</tr>
</tbody>
</table>

Leap Motion controller is unable to detect all fingers of hand, specially middle and pinky finger, moreover, it has a limited working space [109]. Other problems include misdetection during overlapping of hands [114], crossing of field boundaries, and varying lighting conditions [112]. Leap Motion gives inaccurate but comprehensive data while Leap Motion gives comparatively accurate results. Nabiyouni et al. [114] stated that Leap Motion fails to recognize cross over fingers or if they are next to each other. Rotation of palm or fingers more than 80 degrees causes failure in tracking. Moreover it produces significant fatigue.

Kinect is used for detection of full human body gestures to interact with virtual objects [117]. Kumerburg et al. [116] used Microsoft Kinect as input device for navigation in a VE. Different gestures for navigation were proposed such as raising both arms upside for fly and forward movement as shown in Fig. 8. The proposed navigation gestures were hard to learn and use.
Special Issue on Artificial Intelligence Applications

<table>
<thead>
<tr>
<th>Gesture</th>
<th>Command</th>
<th>Characteristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tilt up</td>
<td>Bend backward with hands down</td>
<td></td>
</tr>
<tr>
<td>Turn left</td>
<td>Raise the left hand above shoulder sideways</td>
<td></td>
</tr>
<tr>
<td>Pan left</td>
<td>Move left hand across the body and right hand down</td>
<td></td>
</tr>
<tr>
<td>Fly, move forward</td>
<td>Raise both arms up</td>
<td></td>
</tr>
<tr>
<td>Fly pan right</td>
<td>Raise both arms and move right hand across head</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 8. Gestures proposed by Kumerburg et al. [116] using Microsoft Kinect.

Vulture et al. [118], also used Microsoft Kinect and proposed gesture for navigation using both arms as shown in Fig. 9.

<table>
<thead>
<tr>
<th>Gesture</th>
<th>Characteristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foward</td>
<td>Forward navigation</td>
</tr>
<tr>
<td>Backward</td>
<td>Backward navigation</td>
</tr>
<tr>
<td>Right</td>
<td>Navigation to the right</td>
</tr>
<tr>
<td>Left</td>
<td>Navigation to the left</td>
</tr>
<tr>
<td>Up</td>
<td>Up to a higher level</td>
</tr>
<tr>
<td>Down</td>
<td>Down to a lower level</td>
</tr>
</tbody>
</table>

Fig. 9. Gestures proposed by Vulture et al. [118].

The previous techniques for gesture based navigation mostly depend on variant shapes/layouts of hands, fingers, and arms which lead to extra mental and physical load on user in learning and usage. They are limited in use due to less realistic nature.

### III. Proposed System

We propose a new, two hand gesture based navigation technique for VEs with a close resemblance to car steering driving. The relative position of both thumbs determines various gestures. These gestures are used for 3D navigation in the VE. Navigation includes Forward, Backward, Up, Down, and Left and Right side movement. The green and yellow color caps made of paper/rubber are used with thumbs. The VE consists of different 3D objects as shown in Fig. 10. Identification of different gestures leads the virtual object/camera to navigate accordingly.

![Fig. 10. Scenario of VE (from different perspectives). (a). forward movement, (b). backward movement, (c). Upward movement, (d). Downward movement, (e). Right turn, (f). Left turn.](image)

**A. System Architecture**

The proposed system uses OpenCV as backend and OpenGL as frontend tool. OpenCV is used for image processing, consists of different phases such as image acquisition, conversion of image to HSV (Hue, Saturation, Value), thresholding for green and yellow colors, and finally the calculation of 2D position \((x, y)\) of the specified colors as shown in Fig. 11. OpenGL is used for designing and interaction of the VE. OpenGL, based on position of colors (Green and Yellow), identifies various gestures that leads to different navigation tasks in the VE.

![Fig. 11. (a) Backend (Image processing), (b) Frontend (VE).](image)
First of all OpenCV performs image acquisition via a camera. The image is then converted to HSV for realistic performance which is then thresholded for Green and Yellow colors. In the last stage, position of both colors is calculated dynamically from the image. OpenGl receives the positions of both thumbs colors. On the basis of these positions, different gestures are identified which lead to 3D navigation in the VE.

OpenCV performs image acquisition via ordinary webcam. Finger caps of green and yellow colors are used for left and right thumbs of both hands. Skin color is omitted to achieve best results as it varies from person to person. A range of Hue, Saturation and Values are selected for green and yellow colors to detect thumbs in stable lighting conditions. First the region of interest of the image (RI_img) is extracted from the Frame Image (F_img) to avoid false detection of the background green and yellow colors. The IR_img is then segmented from the F_img based on the skin area which is the most probable area to get the thumb fingers. YCbCr model has the capability to distinguish between skin and non-skin colors [119].

\[
\begin{bmatrix}
    F - Img - YCbCr \\
    (Gb) \\
    (Cr)
\end{bmatrix} =
\begin{bmatrix}
    16 & 128 & 24.8 \\
    −37.3 & −110 & 110 \\
    110 & −93.2 & −18.2
\end{bmatrix} R
\]

(1)

After getting the binary image of F-Img (see Fig. 12), RI-Img with rows ‘m’ and columns ‘n’is extracted from F-Img using algorithm [120] as,

\[
RI_{Img}(m, n) = \left\{ \begin{array}{ll}
F_{Img ROW(m)}(F_{Img}) & \text{if } Lml \\
F_{Img COLUMN(n)}(F_{Img}) & \text{if } Rml
\end{array} \right.
\]

(2)

Where \( Lml \), \( Rml \) and \( Dml \) represent Left-most, Right-most and Down-most skin pixels of the left hand.

\[
RI_{Img}(m, n) = \left\{ \begin{array}{ll}
F_{Img ROW(m)}(F_{Img}) & \text{if } Lmr \\
F_{Img COLUMN(n)}(F_{Img}) & \text{if } Rmr
\end{array} \right.
\]

(3)

Where \( Lmr \), \( Rmr \) and \( Dmr \) represent Left-most, Right-most and Down-most skin pixels of the right hand.

The segmented image RI-Img is then thresholded for green and yellow color simultaneously, using HSV color space (see Fig. 13).

\[
RI_{Img}(x,y) =
\begin{cases}
1 & \text{if } RI_{Img}.H(x,y) \geq 63 \land RI_{Img}.H(x,y) \leq 98 \\
1 & \text{if } RI_{Img}.S(x,y) \geq 98 \land RI_{Img}.S(x,y) \leq 255 \\
1 & \text{if } RI_{Img}.V(x,y) \geq 37 \land RI_{Img}.V(x,y) \leq 255 \\
0 & \text{Otherwise}
\end{cases}
\]

(4)

\[
RI_{Img}(x,y) =
\begin{cases}
1 & \text{if } RI_{Img}.H(x,y) \geq 65 \land RI_{Img}.H(x,y) \leq 80 \\
1 & \text{if } RI_{Img}.S(x,y) \geq 110 \land RI_{Img}.S(x,y) \leq 255 \\
1 & \text{if } RI_{Img}.V(x,y) \geq 145 \land RI_{Img}.V(x,y) \leq 255 \\
0 & \text{Otherwise}
\end{cases}
\]

(5)

\[Fig. 12. Binary images of both hands after conversion to YCbCr.\]

\[Fig. 13. Detected thumbs of both hands.\]

\[B. Navigation\]

Navigation is the movement towards the desired position in a VE. OpenCV computes positions of both thumbs in 2D. The z-axis movement is deduced from the area variation of the detected thumbs. As the area of the thumbs increases with inward and decreases with outward movement in z-axis (towards the camera eye) as shown in Fig. 14 (a, b). So increase from a predefined (threshold) area KA results in forward navigation while decrease results in backward navigation. The value of KA is half of the fully detected thumb area (near to camera eye) which divides the navigation space (z-axis) into two zones i.e. Forward and Backward zone, as shown in Fig. 15. For accurate navigation in the VE, it is necessary that both thumbs should be visible to camera. Movement of both thumbs forward (where detected thumb area >KA) towards camera eye results in forward navigation while moving thumbs backward (where detected thumb area <KA) produces backward navigation.

LPx and LPy represent position of left hand thumb (with green cap) in x and y-axis, RPx and RPy are position of right hand thumb (with yellow color) in x and y-axis, UZ is the upper zone, LZ is the lower zone, LA and RA represent the detected thumb areas.

The algorithm for forward and backward movement is given below:

\[\text{If}(LA>KA \text{ AND } RP>KA)\]

Forward navigation

\[\text{If}(LA<KA \text{ AND } RA<KA)\]

Backward navigation

\[\text{Fig. 14. (a) Detected (Left and Right hand) areas in forward navigation.}\]

\[\text{Fig. 14. (b) Detected (Left and Right hand) areas in backward movement.}\]
For navigation in y-axis, the thumbs of both hands use UZ or LZ. If both thumbs simultaneously move to upper zone (UZ), upward navigation (along y-axis) is produced, while at lower zone (LZ), it leads to downward navigation as shown in Fig. 16.

Mathematically it can be written as:
\[ \text{Sp} \propto \frac{1}{\text{abs}(\text{RP}_x - \text{LP}_x)} \]  
(6)

### D. Turning

Left or right turn is taken by simply comparing the y-position of both thumbs. For left turn, the right hand thumb will be in upper zone (UZ) while left hand thumb will be in lower zone (LZ). For right turn, the right hand thumb will be in lower zone (LZ) while left hand thumb will be in upper zone (UZ) as shown in Fig. 16.

The algorithm for turning right and left is given below:

- If(\(\text{RP}_y > \text{LP}_y\))
  - Turn left (see Fig. 18.a)
- If(\(\text{RP}_y < \text{LP}_y\))
  - Turn right (see Fig. 18.b)

### E. System Implementation

The proposed system was implemented using Visual Studio 2013 with corei3 laptop having 1.7 GHz processor, 4GB RAM, and 640x480 resolution low cost built in camera. The OpenCV, as a backend tool, performs different image processing tasks such as acquisition of image from camera, identification of thumbs colors (green and yellow), and dynamic area and pose calculation of these colors. OpenGL, as a frontend tool is responsible for creation and interaction with VE as shown in Fig. 19. The system allows users to interact (navigate) with VE using his/her both hands thumbs having colored (green, yellow) caps. The left hand thumb uses green color while right hand thumb uses yellow color cap. The combination and relative position of both colored thumb makes different gestures which are used to perform 3D navigation in the VE.

### IV. Experiments and Evaluation

We performed objective and subjective evaluation to assess the accuracy and effectiveness of the proposed navigation technique in VEs. We also used machine learning models i.e. SVM and kNN to assess the accuracy and performance of the proposed gestures.
A. Protocol and Task

Forty (40) volunteer male students participated in the experimental study. Their ages were in the range of 22 to 35 years. All of them had gaming experience with keyboard, mouse, and touch screen but had no experience with gesture based VEs. In the training phase all the students were demonstrated about the use of proposed system and gestures. After that they performed different navigation tasks (Forward, Backward, Up, Down, and turning Left, and Right) using the specified gestures in the VE. Each student performed five pre-trials of each navigation task. After the training session, each participant performed four trails of three different navigation tasks in three different lighting conditions making a total of 1440 trails.

1) Interaction Routes

The experimental environment consisted of three different routes from start to stop position as shown in Fig. 20.

2) Interaction Task

The students need to perform four (4) trials of each of the following three tasks:

- Task 1. The first task was to follow the Route 1 which covers forward navigation (five times), one right turn and four left turns as shown in Fig. 20.
- Task 2. The second task was to follow the Route 2, which covers five forward movements, one left and four right turns.
- Task 3. The third task was to follow the Route 3 which covers one upward, two forward, one downward, and one backward movement.

3) Lighting conditions

The proposed system uses colored thumbs for interaction with the VE. The detection thumb is highly dependent on the surrounding light. So we performed all the tasks in three different lighting intensities.

- Low light intensity (3-7 Lux) Dark limit of civil twilight under a clear sky [128].
- Medium light intensity (50-70 Lux) Family living room lights (Australia, 1998) [129] to office building hallway/toilet lighting [130-131].
- High light intensity (200-500 Lux) Office lighting [129] [132-133], sunrise or sunset on a clear day.

After performing the tasks, task completion time (TCT) and errors for each task were recorded for objective analysis. Misdetection and deviation from the specified route were considered as errors. Finally, each participant filled a questionnaire for subjective analysis.

B. Result Analysis

In this section, we performed objective analysis (task completion time and errors) and subjective analysis (questionnaire) to assess the accuracy and effectiveness of the proposed navigation technique in VEs.

Objective Analysis

1) Task Completion Time and Errors

The Mean and SD of time and errors for task 1 is shown in Fig. 21 and 22. The Mean time and SD is (89.01, 20.02), (82.30, 19.19), and (61.98, 15.16) for low, medium, and high lighting intensity respectively. The Mean and SD of errors is (3.31, 1.49), (2.19, 1.37), (1.71, 0.97) for low, medium, and high light intensities. The Mean for both time and errors are minimum for high lighting intensity as compared to other. It means that the technique has good performance in higher lighting intensity.
Special Issue on Artificial Intelligence Applications

Fig. 24. Mean error and SD for Task 2.

The Mean and SD of time and errors for task 3 is shown in Fig. 25 and 26. The Mean and SD of time is (62.01, 14.11), (56.28, 13.58), and (50.24, 13.89) respectively for low (3-7 Lux), medium (50-70 Lux), and high (200-500 Lux) light intensity.

Fig. 25. Mean time and SD for Task 3.

Fig. 26. Mean error and SD for Task 3.

2) Task Learning

The mean task completion time of each task is shown in Fig. 27, 28, and 29. The results show that the mean time decreases with each successive trial which leads to improved task learning. So it means that task learning improves with experience. The system intrinsically works better in high lighting conditions, while the repetition of task improves learning of task performance. The enhanced learning effect is obvious due to simple and realistic nature of gestures.

Fig. 27. Mean time of each trial for Task 1.

Fig. 28. Mean time of each trial for Task 2.

Fig. 29. Mean time of each trial for Task 3.

Subjective Evaluation

For the subjective evaluation, each student filled a questionnaire. The questionnaire was consisted of different questions related to the following topics:
1. Cognitive load during interaction
2. Technique learning
3. Fatigue during interaction

Likert scale is used for scaling purposes where 1 is for lowest level and 5 for highest level.

1) Cognitive Load During Interaction

The main goal of the proposed technique is to use simple and realistic alignment of fingers or hands in order to get an easy to learn and use interaction. The responses for the question concerned to cognitive load on users during interaction is shown in Fig. 30. The results show that most of the students opted for lowest (52.5%) and low level (35%) of mental load created during interaction.
2) Technique Learning

The navigation technique is developed with the aim of ease in learning. Most of the students selected the proposed technique as easy to learn as shown in Fig. 31. This figure shows that 35% and 40% students opted the highest and high level learning. We used real word phenomena/experience for the proposed navigation technique.

3) Fatigue

The students responses concerning the created fatigue during interaction is shown in Fig. 32. The results show that there is a considerable fatigue (40% students opted for high level and 10% for highest) developed during interaction. So we can conclude that fatigue (physical) increases gradually with time in restless operations while the main reason for creation of fatigue is due to baseless/midair operation of hands gestures for long times.

Performance Evaluation of Gesture Recognition

Various authors used different gesture recognition engines for recognition such as HMM [121-124], Advanced HMM [125], Combined HMM and SVM [126], and kNN [127]. We used SVM and kNN models to assess the accuracy of the proposed gestures. To evaluate the performance of the proposed system, 20 participants were selected to perform 10 trials of each of the six (6) different gestures i.e. Forward, backward, upward, downward, left and right. It makes a total of 1200 gestures.

The confusion matrix for SVM (see Table I) shows 95.3% mean accuracy of the proposed system with 97% (maximum) accuracy for right gesture and 93% (minimum) accuracy for left gesture.

<table>
<thead>
<tr>
<th>TABLE I. Confusion Matrix For SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backward</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>98%</td>
</tr>
<tr>
<td>98%</td>
</tr>
<tr>
<td>98%</td>
</tr>
<tr>
<td>98%</td>
</tr>
<tr>
<td>98%</td>
</tr>
<tr>
<td>98%</td>
</tr>
</tbody>
</table>

The confusion matrix for the proposed system with kNN model shows (see Table II) mean accuracy of 95.7% with 99.0% (maximum) accuracy for upward gestures and 94% (minimum) accuracy for left and right gesture.

<table>
<thead>
<tr>
<th>TABLE II. Confusion Matrix for KNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backward</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>99%</td>
</tr>
<tr>
<td>99%</td>
</tr>
<tr>
<td>99%</td>
</tr>
<tr>
<td>99%</td>
</tr>
<tr>
<td>99%</td>
</tr>
<tr>
<td>99%</td>
</tr>
</tbody>
</table>

1) Comparison of SVM and kNN

Comparison of both SVM and kNN in terms of accuracy and performance is shown in Table III. The results show that kNN has high recognition accuracy (95.7%) as compared to SVM (95.3%). The kNN also has high performance rates in terms of training time (3.16 secs) and prediction speed (6600 obs/sec) as compared to SVM with 6.40 secs and (2900 obs/sec).

<table>
<thead>
<tr>
<th>TABLE III. Comparison of SVM and KNN Based on Recognition Accuracy, Training Time, and Prediction Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>SVM</td>
</tr>
<tr>
<td>kNN</td>
</tr>
</tbody>
</table>
V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel two hand gesture based interaction technique for 3 dimensional (3D) navigation in virtual environments (VEs). The system uses computer vision techniques for the detection of hand gestures (colored thumbs) from real scene and performs different navigation (forward, backward, up, down, left, and right) tasks in the VE. The proposed technique also allows users to efficiently control speed during navigation. This is implemented via a VE for experimental purposes. Forty participants tested the proposed technique in different lighting scenarios. Experiments revealed that the technique is feasible in normal lighting conditions, easy to learn and use, having less cognitive load on users. Its performance is evaluated using gesture recognition engines i.e. SVM and kNN. kNN achieves high accuracy rates 95.7% as compared to SVM (95.3). kNN also has high performance rates in terms of training time (3.16 secs) and prediction speed (6600 obs/sec) as compared to SVM with 6.40 secs.

The proposed system is sensitive to lighting conditions.

In future we will compare our proposed camera based system with Motion leap in terms of accuracy, interaction space, update rate, and positional distortion.

REFERENCES


[109] Shao, L., Hand movement and gesture recognition using Leap Motion Controller.


[111] Khundam, C. First person movement control with palm normal and hand gesture interaction in virtual reality, in Computer Science and Software Engineering (JCSE), 2015 12th International Joint Conference on. 2015. IEEE.


and recognition in dynamic scenes using ridge data. in Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on. 2014. IEEE.


Mr. Rehman
He is currently pursuing Ph. D. degree in 3D Interaction from the University of Malakand, Pakistan. He received his M. Phil in Virtual and Augmented Reality and BS degree in Computer Science from University of Malakand in 2006 and 2016 respectively. His research interest includes 3D interaction, virtual and augmented reality, and machine learning.

Dr. Ullah
He is currently working as Associate Professor in the department of Computer Science & IT, University of Malakand, Pakistan. He received his PhD degree in Computer Science in specialty in Robotics from University of Evry Val d’Essonne, France in 2011. He also earned MS degree of Computer Science with specialty in Virtual Reality and Intelligent Systems from the same University in 2007. He received MSc degree of Computer Science and BSc with major in Mathematics and Computer Science from University of Peshawar in 2001 and 1998 respectively. Dr. Sehat Ullah has research interest in virtual and augmented reality systems, machine learning, image processing and haptic systems.

Mr. Raees
He is serving as assistant professor of computer science in the department of Higher Education, Khyber pakhtunkhwa, Pakistan. He received his M.Phil degree from University Of Malakand in 2015 and is currently pursuing Ph.D. in flexible 3D interaction from the same university. His research interest includes 3D interactions, computer vision and machine learning.