



VEN-3DVE: vision based egocentric navigation for 3D virtual environments

Muhammad Raees¹ · Sehat Ullah¹ · Sami Ur Rahman¹

Received: 21 March 2018 / Accepted: 23 April 2018 / Published online: 5 May 2018
© Springer-Verlag France SAS, part of Springer Nature 2018

Abstract

Navigation is moving inside a virtual environment to reach to a desired object or location. This paper presents a new egocentric navigation technique for exploring a virtual scene by tracking the 3DOF moves of index finger along x, y and z axis. With image processing at backend, the finger's position in 2D image frame is traced dynamically to travel inside the designed frontend 3D scene. Mimicking a camera in hand, the system can also support Panning. Dynamic mapping is performed to interactively follow the trajectory of index finger. For either of the interactions, movement of index finger controls orientation and position of virtual camera. Linking the libraries of OpenCV and OpenGL, the technique is implemented in a visual studio project navigation by index tracing (NBIT). With NBIT, parallel processing is performed for backend index tracking and front-end rendering of virtual scene. The system was evaluated by fourteen participants for a total of 336 trails whereas applicability of the approach in virtual reality based computer aided design was assessed by five professional designers. Un-biased statistical evaluation of the system reveals that the approach suits well for interactive virtual and augmented reality applications.

Keywords Gesture based interaction · Navigation in virtual environment · Design engineering by Finger's movement

1 Introduction

Akin to the real world, interaction brings realism in Virtual Environments (VE). Out of the four interaction tasks as categorized by Gabbard [1], navigation is more important because most often than not, it is the first step to perform any selection and/or manipulation. The far-off spreading of virtual scenes cannot be viewed from a single static point of view. Therefore user should have the ability to navigate in the VEs and to explore different parts and portion it encloses. Furthermore, to manipulate an object of a 3D world, we have to select the object and before selection we have to access the object. Thus, navigation becomes a preliminary task even if it is not desired. Although, a number

of navigation techniques have been proposed so far to make this frequently used interaction errors-free and cost-effective, but naturalism and intuition are still the challenges to be covered. Low-cost vision based tracking on the other hand provides an applicable platform to devise a flexible interface for Human Computer Interactions (HCI). This research work is an attempt to make navigation simple and realistic by bridging real and virtual world while keeping accuracy and cost in checks. VEN-3DVE tracks real world movement of index finger using an ordinary camera to walk inside the designed synthetic world. At the back-end, from a scanned dynamic image, the system traces position of index finger. Area of the tip-of-index finger is calculated to predict position of hand on z-axis. The position and area of index finger are then forwarded to Front-end of the system for actual operation. Front-end of the system renders the scene accordingly with an avatar representing user's position in the VE. A fingertip-thimble made of simple piece of paper is used to segment out index from the rest of a scanned image frame. The thimble is marked green for unambiguous and hasty segmentation. Forward and backward gesture of index finger about the z-axis performs navigation while horizontal and vertical movement of the finger performs panning about

✉ Muhammad Raees
visitrais@yahoo.com

Sehat Ullah
sehatullah@hotmail.com

Sami Ur Rahman
softrays@hotmail.com

¹ Department of Computer Science and IT, University of Malakand, Chakdara, Pakistan

x-axis and y-axis respectively. The system can be used for all the three sub-navigation tasks; exploration, searching and inspection as discussed by Tan et al. [2].

VR based Computer Aided Design (VR-CAD) is on the rise to make designing process more simple and natural. Navigation and Panning are used in VR-CAD particularly in the designing of large Digital Mock-Up (DMU) [3]. The proposed approach can be easily extended to gesture based VR-CAD without using any extra tool or toolkit [4]. Furthermore, the issue of pointing imprecision [5] can be avoided to a satisfactory extent.

This research paper is structured into six sections. Section 2 is about related work and Sect. 3 explains details of the proposed system. Section 4 illustrates implementation and evaluation details while Sect. 5 is about applicability of the approach in VR-CAD. The last section is about conclusion and future work.

2 Related work

The continual advancement in storage and processing of computer system aids in the emerging virtual reality applications. However, the improvement made so far in 3D interaction fails to follow the ceaseless progress. Being the most oftenly used interaction task, navigation is in the focus of different research works to make exploration of VEs feasible and flexible. Most of the techniques for direct navigation make use of traditional mouse/keyboard or HMD [2]. Where the former lacks intuition in interaction and engineering design [7], the later remains a second choice because of high cost. The Multi-Finger gestural navigation of Malik et al. [8] works on a constrained tabletop surface. Moreover, the system is bi-manual and is therefore not suitable for hand-held devices. The NuNav3D approach of navigation necessitates whole body pose estimation before hand gesture recognition [9], furthermore the system is 79% slower than joy-pad based navigation. The technique of Lee et al. [10] consumes a large amount of processing for finger action recognition as the system ought to pass from three hefty stages; skin-color detection, k-cosine based angle detection and contour's analysis each for finger's state, position and direction. Furthermore, due to variable finger's thickness, accuracy of the system varies from user to user. In the Drag'n Go [11] the screen cursors position casts ray to the target for navigation. As a straight path is to be followed, therefore the technique suites well for navigation inside a large empty space but unadoptable for zigzag navigation. Similarly in the system of Tan et al. [12], user have to drag a mouse in a particular direction to move virtual camera. For a long travel user have to repeat the practice over and over again. Furthermore, if the ray collides with an object in the way, the system mistakenly inspects the collided object

instead of proceeding navigation. With head-directed navigation approach [13], navigation speed and direction are calculated from head pose. The estimation may mistakenly interpret casual head's movement. The MC (Management Cabin) in FmF (Follow my Finger) model [14] projects view of 3D world on 2D table-top device suffers from disorientation. Hand gesture recognition using different trackers [15,16], are the navigation techniques where virtual scene is treated as a big object and are grabbed/moved with both hands. The techniques are interesting but the hand-worn overload and the complex nature of use make them a rare option. The walking in place approach [17] relates user walking pace with navigation speed in virtual environment. The approach cannot be used while setting on chair. The system needs a large number of sensors and is implementable only inside a dedicated lab. Similarly, the foot-based interface for navigation suggested by [18] requires a cumbersome setup of waist-based magnetic trackers with conveyer belt for navigation. Fiducial markers have also been utilized effectively for navigation. The DeskCube [19] as a passive input device is used where different markers on different faces of a cube are glued. The system fails to cope with occlusion and blurring of markers with gentle movement of markers cube. The leap motion based technique [20] is not fully immersive as users have to activate commands by clicking buttons within a limited space. Fingers based locomotion; FWIP (Finger Walk In Place) [21] for mobile based VEs necessitates touch of the fingers on display screen for navigation. The pointing technique of Radkowsk et al. [22] uses two fingers; one for viewing and the other for direction. Though it successfully avoids the mistakes of gaze-directed navigation, speed controlling is its main challenge.

3 VEN-3DVE

The steering based navigation metaphor [23,24] is considered a standard for its close resemblance to real world navigation. Keeping the insight of steering, the proposed system follows index finger movement to interact dynamically with the designed VE. Being a dominant finger, algorithm of the system works on dynamic movement of index finger. To interact with VE using dynamic movement of the finger, it is necessary to precisely extract the part and portion of hand on which interaction routine depends. A green color finger-cap made of paper is used for this purpose to flawlessly detect the index finger. Furthermore, lest the system wrongly suppose a green color object, first the detection of skin color is made to extract segment of scanned image containing the finger's pose. A range of Hue, Saturation and Values (*HSV*) space for green color is set to detect index in a balanced lighting condition. At the detection of index-cap, first a Central Zone (*CZ*) is specified from 2D position of index finger in initial

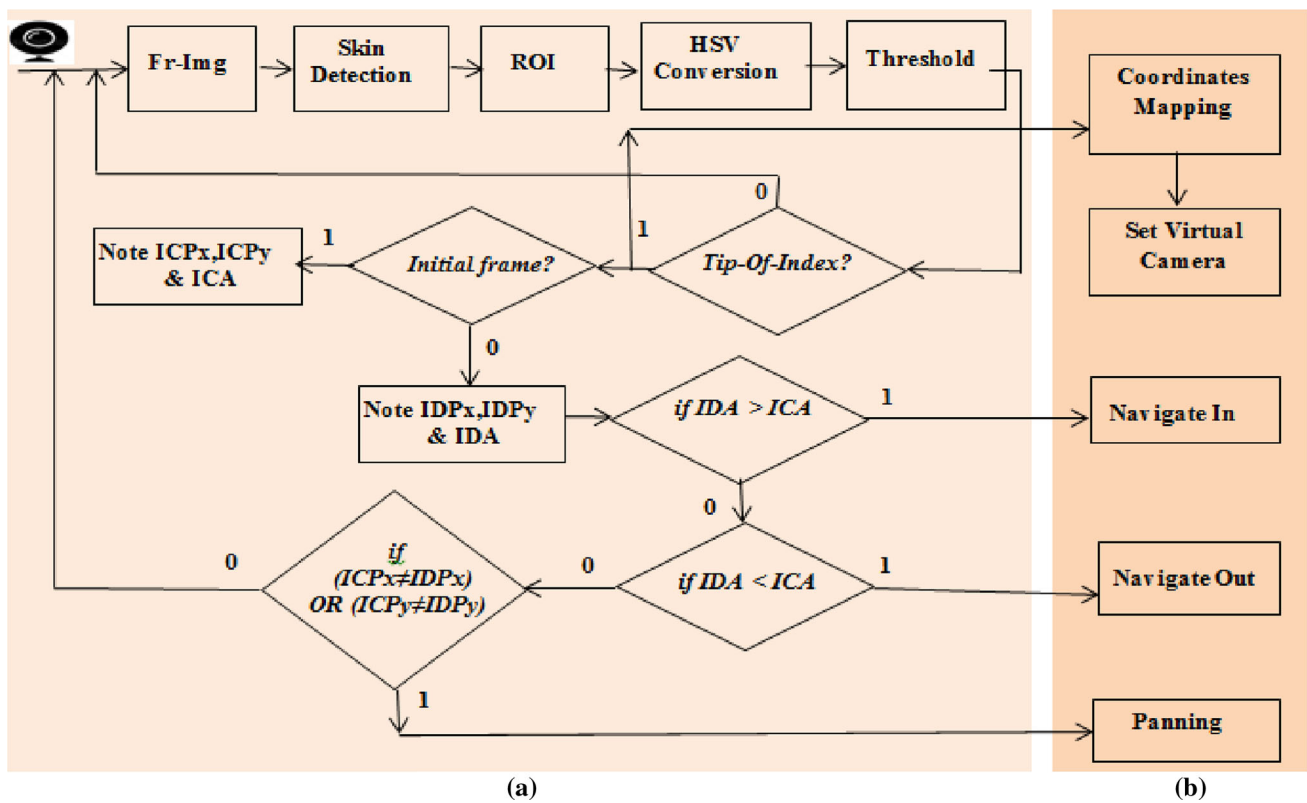


Fig. 1 Schematic of the system a backend (image processing) and b frontend (VE)

frames. Panning in either direction is performed until the finger’s movement is inside the CZ. Like a real camera in hand, navigation is performed by moving the finger along z-axis beyond the CZ.

3.1 Architecture design

The system starts with a virtual environment comprises different 3D objects. The real world scanned images are thresholded dynamically for a broader range of green color at the back-end. Once thimble of index finger is traced, coordinates mapping, for the index-tip is performed to locate virtual camera and users position in the virtual environment. As long as the index cap is visible, the virtual camera can move freely with index finger movement to travel or pan. As conceivable, outside the CZ, forward movement of index finger navigates user inside the look vector, while the backward movement is to zoom out. Horizontal and vertical finger’s move changes the eye coordinates of virtual camera accordingly. Schematic of the proposed system is shown in Fig. 1.

3.2 Details of the algorithm

Based on skin color, a ROI-Img (Region Of Interest Image) is extracted from the whole scanned Frame Image (Fr-Img).

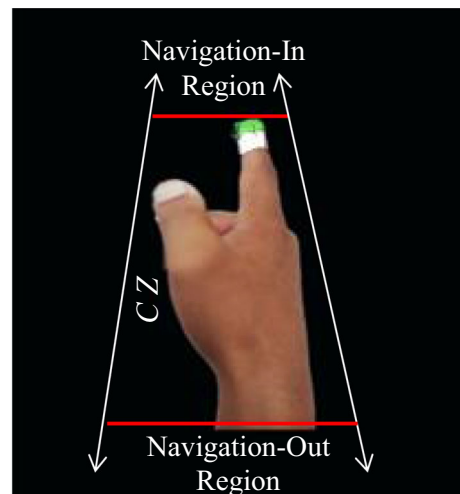


Fig. 2 Gesture with CZ and navigation regions

ROI-Img is then thresholded for green color to trace tip of the index finger. From the 2D position of the finger in ROI-Img, Index Central Points ($ICPx$, $ICPy$) and ICA (Index Central Area) are saved while CZ is made set from $ICPx$, $ICPy$ and ICA. CZ is a limited distance over the look-vector but covering the entire horizontal and vertical range of real camera, see Fig. 2. Navigation is performed only behind and beyond

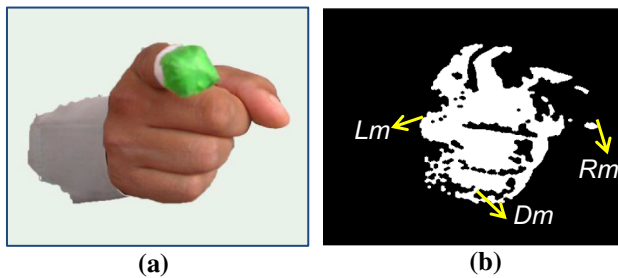


Fig. 3 Fr_Img in **a** RGB and in **b** binary after conversion into $YCbCr$

the CZ . Within CZ , finger movement on xy -plane is reserved for horizontal and vertical panning.

3.2.1 Image segmentation

To avoid possibility of false detection of the background green color, first ROI_Img is extracted. ROI_Img which is the most probable area of Fr_Img to have index finger, is segmented out on the basis of skin color. As the $YCbCr$ space provides best discrimination between skin and non-skin colors [25], therefore skin color is extracted by using the $YCbCr$ model. The binary Fr_Img from scanned RGB image frame is obtained as,

$$Fr_Img \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.1 & 128 & 24.8 \\ -37.3 & -74 & 110 \\ 110 & -93.2 & -18.2 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

After getting the binary image of Fr_Img , see Fig. 3, ROI_Img with rows ' m ' and column ' n ' is extracted from Fr_Img using our designed algorithm [26] as,

$$ROI_Img(m, n) = \left(\bigcup_{r=Dm}^{Fr_Img.Row(0)} (Fr_Img), \bigcup_{c=Lm}^{Fr_Img.Column(Rm)} (Fr_Img) \right) \quad (2)$$

Where Lm , Rm and Dm represents *Left-most*, *Right-most* and *Down-most* skin pixels

The segmented image ROI_Img is then thresholded for greecolor using HSV color space as,

$$ROI_Img(x, y) = \begin{cases} 1 & \text{if } ROI_Img.H(x, y) \geq 47 \wedge ROI_Img.H(x, y) \leq 94 \\ 1 & \text{if } ROI_Img.H(x, y) \geq 100 \wedge ROI_Img.H(x, y) \leq 187 \\ 1 & \text{if } ROI_Img.H(x, y) \geq 102 \wedge ROI_Img.H(x, y) \leq 255 \\ 0, & \text{Otherwise} \end{cases}$$

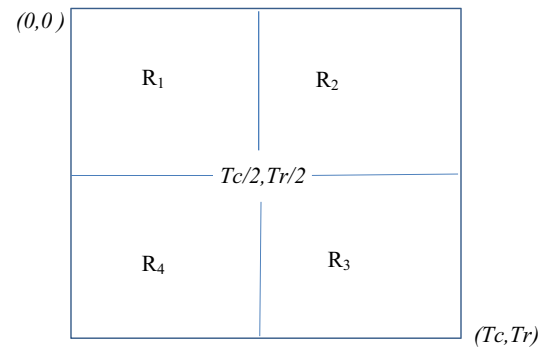


Fig. 4 Virtual division of the image frame

3.2.2 Coordinates mapping

One of the tedious challenges in the implementation was to harmonize image pixels representing tip-of-index to locate virtual camera in the VE. As in OpenCV, image frame starts with $(0,0)$ at top left while in OpenGL, $(0,0)$ lies at the center of virtual environment hence the coordinates are entirely different. To synchronize the dissimilar coordinate systems, we devise our four mapping functions m_1, m_2, m_3, m_4 . The image frame is virtually split into four regions R_1 to R_4 as shown in Fig. 4, where mapping for a region R_n is made by the corresponding m_n taking x and y of a pixel of R_n as independent variables.

$$m_1(x, y) = ((Px - (Tc/2) / (Tc/2)), (Tr/2 - Py) / (Tr/2)) \quad (3)$$

$$m_2(x, y) = ((Px/Tc), (Tr/2 - Py) / (Tr/2)) \quad (4)$$

$$m_3(x, y) = ((Px - (Tc/2) / (Tc/2)), (Py/Tr)) \quad (5)$$

$$m_4(x, y) = ((Px - (Tc/2) / Tc/2), (Py/Tr)) \quad (6)$$

In the above functions, Px and Py represent ' x ' and ' y ' positions of the traced pixels in image frame. ' Tc ' and ' Tr ' represent *Total columns* and *Total rows* respectively. Rendering of the virtual scene after the mapping is shown in Fig. 5.

Position of the finger at the time of initial detection is supposed to be in appropriate range from the real camera. To make $ICPx$ and $ICPy$ precise, arithmetic mean of ' x ' and ' y ' positions of the pixels representing center of finger-tip are calculated from the first five images. Hence, the zone; CZ is set as soon as index-tip is recognized by the system. CZ represents all positions of the index until the Index Dynamic Area (IDA) doesn't exceed the ICA .

$$CZ = \left(\bigcup_{i=0}^{Tr} x_i, \bigcup_{j=0}^{Tc} y_j \right) \leftrightarrow IDA \leq ICA \quad (7)$$

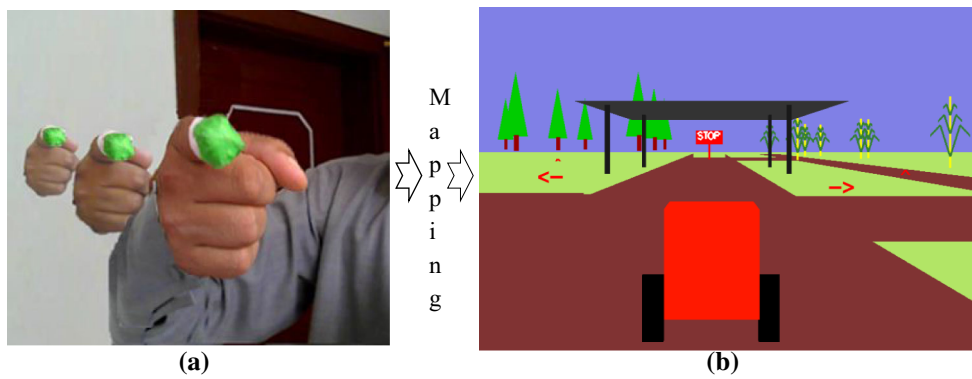


Fig. 5 a The finger move in image frame and b Avatar’s moving inside the VE

Fig. 6 The forward move of finger beyond CZ for navigation-in

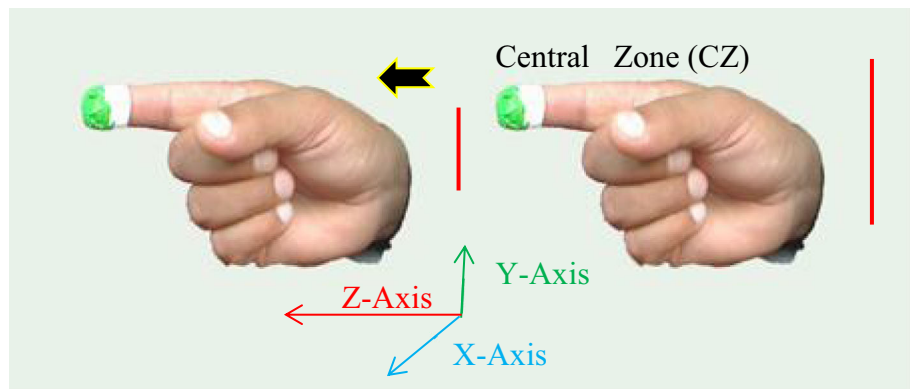


Fig. 7 a The index-tip ICA in an initial image frame and b the IDA after forward hand movement

3.3 Navigation

Navigation in 3D VE is the moving of virtual camera towards or away from a look-at point. In the proposed system, forward movement, beyond the CZ shifts virtual camera towards the look-at point as shown in Fig. 6. Over the look vector, moving the finger behind the CZ, zoom out the virtual scene.

Since the image frames scanned are in 2D, forward or backward hand movement on z-axis is deduced from variation in area of the index-tip. IDA of the index-tip is calculated on the fly for each thresholded frame and is compared with ICA. Beyond the forward CZ limit, IDA is increased by k, see Fig. 7b.

$$IDA = (k) (ICA) \tag{8}$$

where $k > 1$

The unintentional increase/decrease possibility is avoided by checking dynamic positions of index finger on x-axis (IDx) and y-axis (IDy) with ICPx and ICPy along-with ICA and IDA. This is clear from the following pseudo-code where a range of ten pixels is set to accurately deduce navigation.

```
if (IDA > ICA) AND (IDx ≤ ICPx + 10 AND IDx ≥ ICPx - 10)
    AND (IDy ≤ ICPy + 10 AND IDy ≥ ICPy - 10)
```

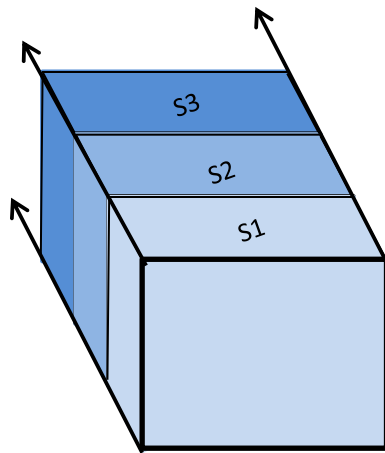


Fig. 8 First three speed sectors of forward navigation

Forward Navigation

if($IDA < ICA$) AND ($IDx \leq ICPx + 10$ AND $IDx \geq ICPx - 10$)
 AND ($IDy \leq ICPy + 10$ AND $IDy \geq ICPy - 10$)

Backward Navigation

3.3.1 Speed control

Depending on the quality of camera and the position of user hand from it, different finite speed sectors S_n can be set where,

$$n = 1, 2, 3, \dots, k$$

Navigation speed in sector S_{n+1} is double of the speed in S_n . Constant Speed SP_n in a particular sector S_n is retained till entrance into next sector S_{n+1} , where SP_n is calculated as,

$$SP_n = \sum_{k=2}^n SP_{(k-1)} \tag{9}$$

Fig. 9 CZ for panning and navigation area over the look-vector

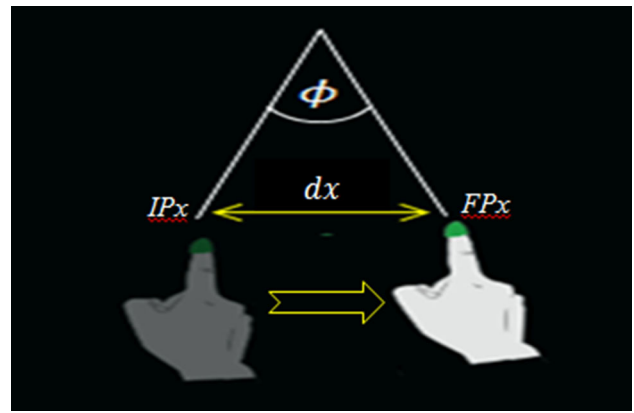
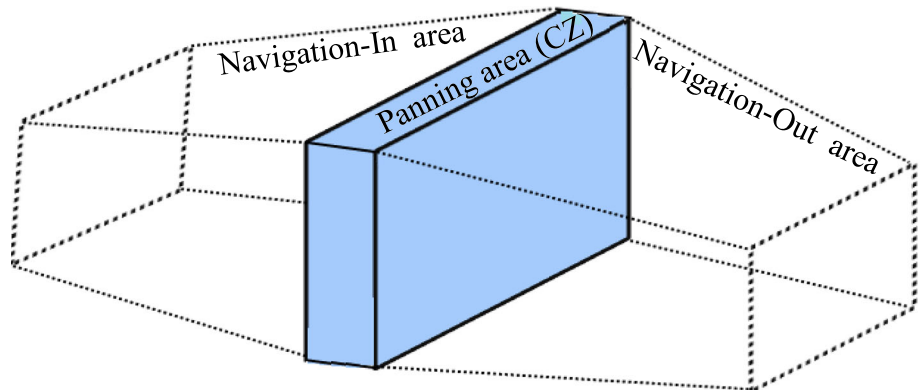


Fig. 10 The finger's move over x-axis for right-panning

To determine the entrance of index-tip in a particular sector S_n , shown in Fig. 8, the following condition is checked on the fly,

$$IDA \geq (n) ICA$$

3.4 Panning

Panning is to translate horizontally or vertically the eye position and look-at point of the virtual camera. To prevent any possibility of disorientation and confusion, panning is enabled only inside CZ, see Fig. 9. Panning over x-axis is performed by horizontal hand movement, while along y-axis by vertical movement. Like the camera-in-hand metaphor, the virtual camera follows index finger or hand movement. Movement along the +ve x-axis shifts the x-coordinate of camera on +ve axis while the point of view (POV) towards -ve x-axis accordingly. Similarly, vertical hand movement changes the y-coordinate of virtual camera with look-at point on the opposite y-axis (Fig. 10).

If $IP_{(x,y)}$ and $FP_{(x,y)}$ represents Initial Point and Final Point of the index-tip in panning area then, horizontal change dx and vertical change dy are calculated as;

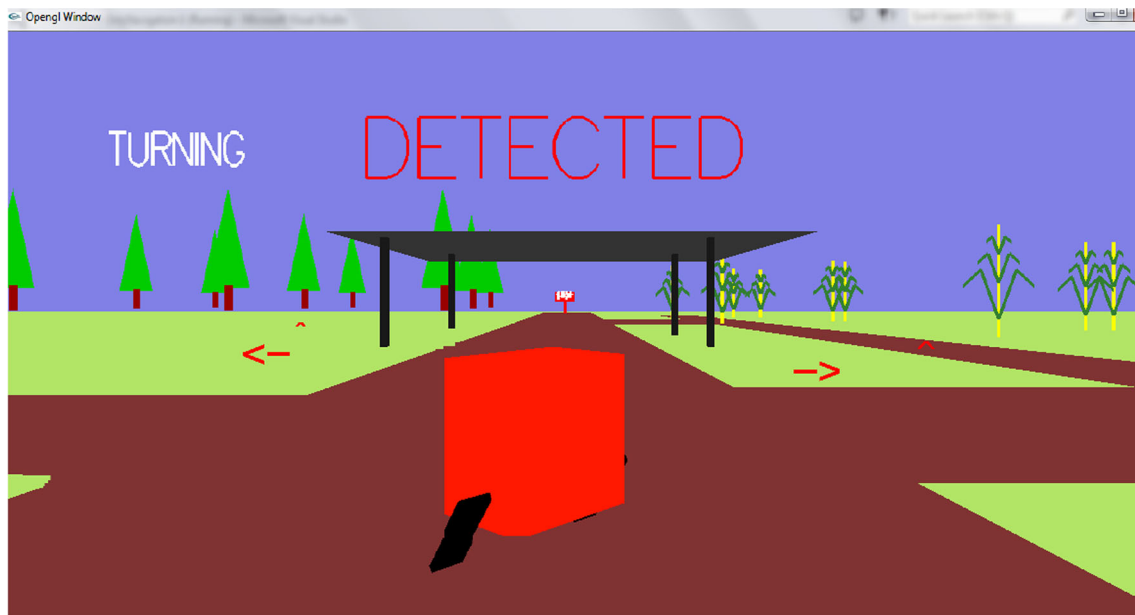


Fig. 11 Virtual scene after left panning activation

$$dx = \sqrt{(FP_x - IP_x)^2} \quad (10)$$

$$dy = \sqrt{(FP_y - IP_y)^2} \quad (11)$$

The following algorithm is followed for horizontal and vertical panning.

If $dx > dy$

If $(IP(x) < FP(x))$

Panning along -ve x - axis

If $(IP(x) > FP(x))$

Panning along +ve x - axis

If $dy > dx$

If $(IP(y) < FP(y))$

Panning along -ve y - axis

If $(IP(y) > FP(y))$

Panning along +ve y - axis

If I_x and F_x are the pixels representing the horizontal initial and final position of index-tip then the virtual camera's x (C_x) and z (C_z) coordinates of for panning are calculated as,

$$dx = F_x - I_x \quad (12)$$

$$\theta = 2 * \sin^{-1} (dx/F_y) \quad (13)$$

$$C_x = \sin(\theta) \quad (14)$$

$$C_z = \cos(\theta) \quad (15)$$

4 System implementation and evaluation

The system was implemented in Visual Studio 2015 using a Corei3 laptop running with 2.30 GHz processor and 4GB RAM. Resolution of the built-in camera was set at 640x480. Tracing of the Index finger cap, its area calculation and detection of horizontal and vertical coordinates was carried out by the OpenCV routines at the backend. The interactive front-end virtual scene, was designed in OpenGL. The system remains active only when index cap is visible. About the system activation, user is constantly made informed by the text "Detected" displayed in upper center part of the scene. Similarly, with left or right panning user is notified with text "Turning" at the respective side of the scene as shown in Fig. 11.

The system was tested by fourteen male participants having ages between 25 and 40. Two trials were performed by each participant for each of the four pre-defined tasks. Before actual trials, participants were introduced to the system and pre-trials were performed by each tester for both navigation and panning.

4.1 Testing environment

The 3D environment designed for the evaluation of the algorithm contained four routes as shown in Fig. 12 where an avatar made of cubes represents the user's position in the environment. For easy noticing, end point of the scene is marked by a board with text "Stop". Each route leads to the one Stop-board. To immerse users and to have a perception of navigation and panning, the scene was designed with dif-

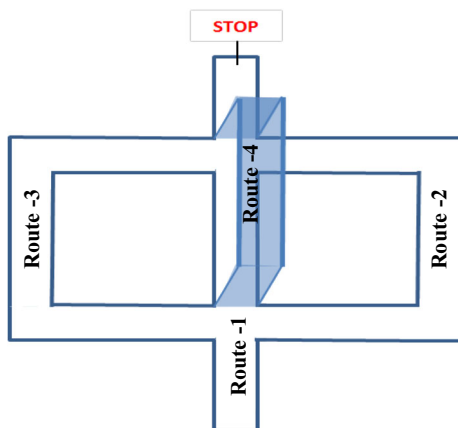


Fig. 12 2D model of the routes

ferent 3D objects at different positions. For each new trial, users were asked to press Enter key to reset the system.

Route-1: Straight pathway leading to Stop-board.

Route-2: Right-Straight-Left pathway.

Route-3: Left-Straight-Right pathway.

Route-4: Up-Straight-Down pathway to have a flying effect to navigate over the bandstand. Lest the scene became complicated, this route was intentionally not rendered in the VE.

4.2 Interaction tasks

Participants were asked to perform the following four interaction tasks in the designed 3D environment.

Task_1: Touching the Stop-board using Route-1 and then back to starting point following the same route.

Task_2: Touching the Stop-board using Rout-2.

Task_3: Touching the Stop-board using Rout-3.

Task_4: Touching the Stop-board using Route-4.

Navigation-In is the forward while Navigation-Out is the backward movements inside the VE. Left/Right panning is turning, with virtual camera shift, towards the respective direction. Task-1 tests Navigation-In and Navigation-Out. Task-2 and Task-3 are to evaluate Navigation-In, Right panning and Left panning. Task-4 assesses Up panning, Down panning and Navigation. Missed detection or false detection of the system after posing the required gestures were counted as errors. With this setup, overall accuracy rate for all the 336 trials, as shown in Table 1, is 69.9%.

As shown in Fig. 13, mean of navigation is comparatively high than panning; Fig. 14. The obvious reason is the crossing the detection limits of camera.

Table 1 Statistics of the evaluation

Interaction	Correct	Total	%Age
Navigation-in	48	56	85.7
Navigation-out	41	56	73.2
Panning left	40	56	71.4
Panning right	35	56	62.5
Panning up	36	56	64.2
Panning down	35	56	62.7
Total	235	336	69.9

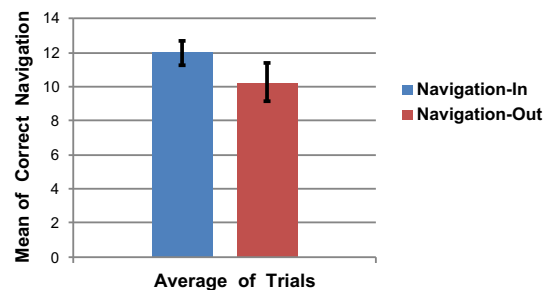


Fig. 13 Mean with standard deviation of *Navigation*

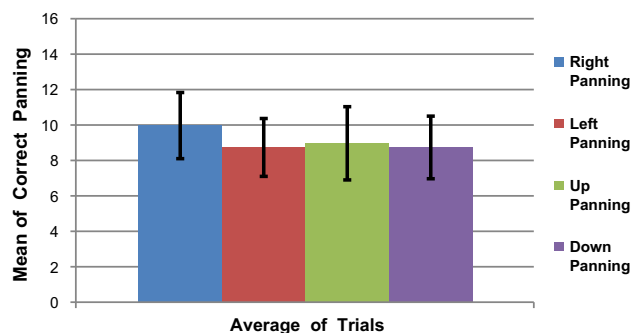


Fig. 14 Mean and standard deviation of *Panning*

A questionnaire measuring the four factors; *Ease of Use*, *Suitability in VE*, *Naturalism* and *Fatigue*, was presented to the users at the end of valuation session. Percentage of user's response to the four factors is shown in Fig. 15.

The learning effect was measured from the error occurrence rate. The graph in Fig. 16 indicates performance increases while error rate decreases in subsequent trials.

5 Applicability of the approach in interactive designing

Interactive design is all in all designing interactive digital products or environment offering an effective and expressive interface for interaction. A design artifact based on interactive approach, is not only friendlier and natural than traditional GUI [27] but can also assess user to analyze com-

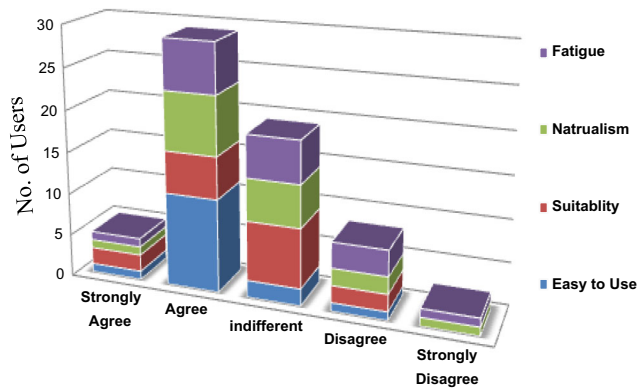


Fig. 15 Participant's response about the proposed approach

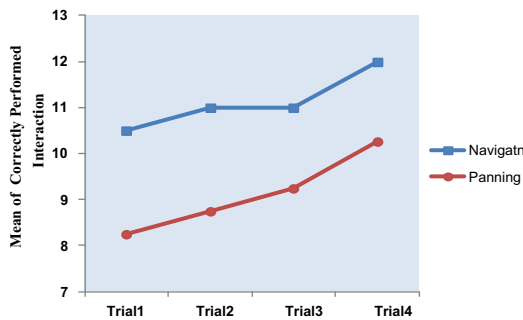


Fig. 16 Graph showing learning effect of the system

plex data and function [28]. The mingling of VR technology with CAD ensures such natural and multimodal interfaces; hence, VR-CAD is replacing the conventional CAD system [29]. Traditionally, haptic devices were used for interactive Designing [30] and Virtual Assembly (VA) [31]. However the cumbersome setup of wires restrict such systems to

a controlled environment. LCM based systems have been developed to avoid the hindrances of haptics, but recognition accuracy of LCM is good only within limited zones [32]. As only intuitive interface with universal applicability is preferred [28], the NBIT project was enhanced to NBIT-SD (NBIT for Simple Designing) to see applicability of the proposed method in the realm of VR-CAD. With NBIT-SD three interaction tasks; *Selection*, *DeSelection* and *Translation* were introduced with a Virtual Hand (VH) to represent finger's position in the VE. *Selection* and *DeSelection* were performed by keep hovering VH over an object for a threshold time; hover-time (HT) of 3 seconds. To make translation of a selected object possible, panning in NBIT-SD was performed only when VH move beyond the boundaries of interface window. Five skilled engineers (two Auto-CAD experts and three software engineers) were invited to evaluate the system. Two trials (one pre-trials and one true-trial) were performed by each engineer to design the scene shown in Fig. 17 by *Selecting*, *Translating* and *DeSelecting* virtual objects presented in the NBIT-SD application, shown in Fig. 18.

Based on their subjective analysis, shown in Fig. 19, the approach can be easily enhanced to make it applicable in VR-CAD technology particularly for semantic zooming [33] and analyses of widely spanned complex 3D objects [27]. Most of the experts suggested that selection and DeSelection should be performed by distinct gestures to make the approach applicable.

6 Conclusion and future work

Navigation is often required in many 3D interactive virtual spaces. The emergence of virtual and augmented reality

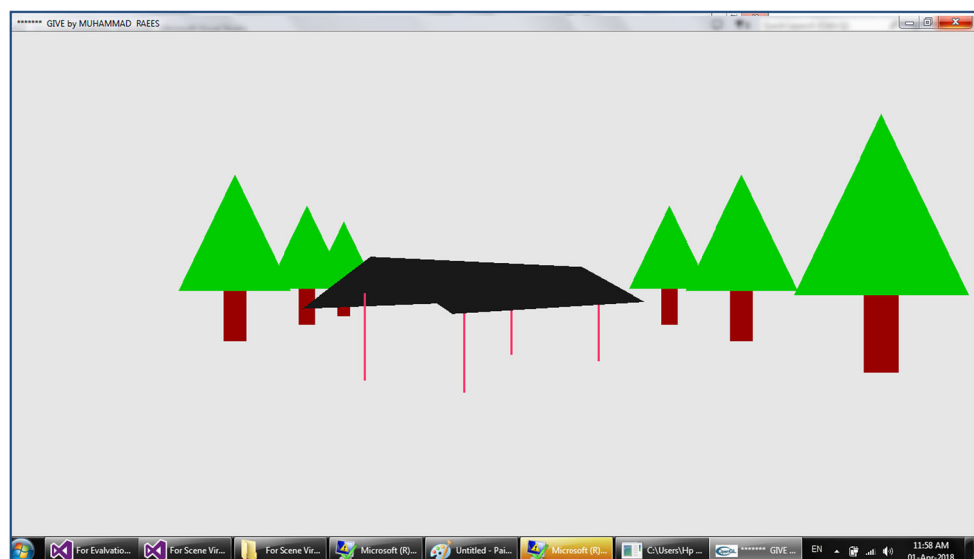


Fig. 17 The required virtual scene

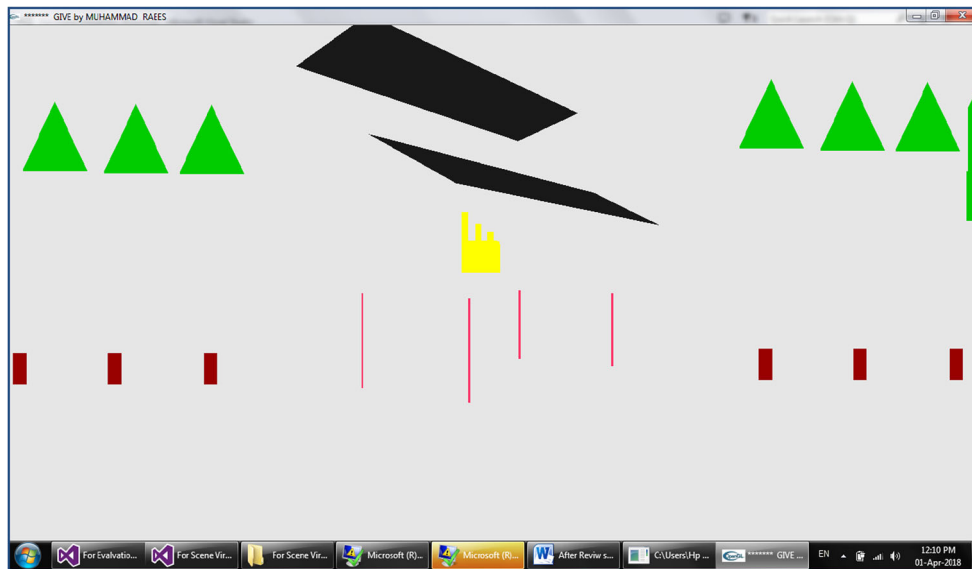


Fig. 18 The NBIT-SD interface with virtual objects

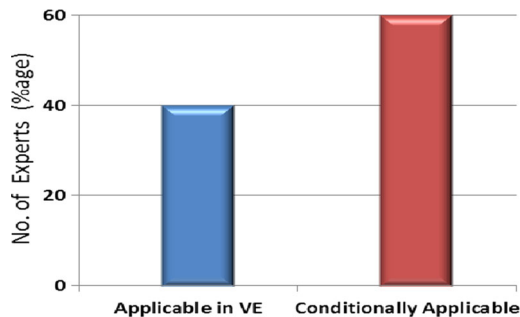


Fig. 19 Response of the experts about applicability of the approach in VR-CAD

applications in different fields necessitates natural and simple way of navigation. With this contribution we proposed a novel navigation technique which needs no extra device other than ordinary camera and a piece of paper. Intuitive gestures of index finger are used for panning and navigation to ensure naturalism. Experimental results show that the proposed approach has reliable recognition and accuracy rates. As the system needs mere conversion of scanned images to HSV color space rather than feature extractions, therefore time is not wasted in computation. Furthermore, as single finger is used for interaction; therefore possibility of occlusion is reduced. The proposed system is equally applicable in a wide spectrum of HCI including CAD, 3D gaming, engineering, medical and simulation. The work also covers the smooth integration of image processing and virtual environment which can leads to the design of more sophisticated virtual and augmented reality applications.

This research work is a fraction of our aim of making possible interaction outside virtual reality labs. Although,

we have succeeded for navigation, rotation via single finger gestures is yet to be covered. In future, we are determined to enhance the system for rotation as well. Furthermore, we are planning to make the system capable to be used in complex virtual collaborative environment.

References

- Gabbard, J.L.: A taxonomy of usability characteristics in virtual environments. Doctoral dissertation, Virginia Tech (1997)
- Tan, D.S., Robertson, G.G., Czerwinski, M.: Exploring 3D navigation: combining speed-coupled flying with orbiting. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 418–425 (2001)
- Di, G.G., Rocco, M., Andrea, T.: From virtual reality to web-based multimedia maintenance manuals. *Int. J. Interact. Des. Manuf. (IJIDeM)* 7(3), 183–190 (2013)
- Schlmer, T., Benjamin, P., Niels, H., Susanne, B.: Gesture recognition with a Wii controller. In: Proceedings of the 2nd International Conference on Tangible and Embedded Interaction, pp. 11–14 (2008)
- Konig, W.A., Jens, G., Stefan, D., Harald, R.: Adaptive pointing design and evaluation of a precision enhancing technique for absolute pointing devices. In: IFIP Conference on Human-Computer Interaction, pp. 658–671 (2009)
- Terziman, L., Maud, M., Mathieu, E., Franck, M., Bruno, A., Anatole, L.: Shake-your-head: revisiting walking-in-place for desktop virtual reality. In: Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology, pp. 27–34 (2010)
- Fiorentino, M., Rafael, R., Christian, S., Antonio, E.U., Giuseppe, M.: Design review of CAD assemblies using bimanual natural interface. *Int. J. Interact. Des. Manuf. (IJIDeM)* 7(4), 249–260 (2013)
- Malik, S., Abhishek, R., Ravin, B.: Interacting with large displays from a distance with vision-tracked multinger gestural input. In: Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology, pp. 43–52 (2005)

9. Papadopoulos, C., Daniel, S., Arie, K.: Nunav3d: a touch-less, body-driven interface for 3d navigation. In: Virtual reality short papers and posters (VRW), pp. 67–68 (2012)
10. Lee, D., Seung, Gwan L.: Vision based finger action recognition by angle detection and contour analysis. *ETRI J* **33**(3), 415–422 (2011)
11. Moerman, C., Damien, M., Laurent, G.: Drag'n Go: simple and fast navigation in virtual environment. In: 3D user interfaces (3DUI). In: 2012 IEEE symposium on, pp. 15–18 (2012)
12. Tan, D.S., Robertson, G.G., Mary, C.: Exploring 3D navigation: combining speed-coupled flying with orbiting. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 418–425 (2001)
13. Ajaj, R., Frdric, V., Christian, J.: Follow my Finger navigation. In: IFIP conference on human-computer interaction, pp. 228–231 (2009)
14. Mine, M.R., Frederick P.J., Carlo H.S.: Moving objects in space: exploiting proprioception in virtual environment interaction. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, pp. 19–26 (1997)
15. Pierce, J.S., Forsberg, A.S., Conway, M.J., Seung, H., Zeleznik, R.C., Mark R.M.: Image plane interaction techniques in 3D immersive environments. In: Proceedings of the Symposium on Interactive 3D Graphics, pp. 39–ff (1997)
16. Fuhrmann, A., Dieter, S., Michael, G.: Strolling through cyberspace with your hands in your pockets: head directed navigation in virtual environments. In: Virtual Environments, pp. 216–225 (1998)
17. Furness, T., Configuring virtual space for the super cock-pit, Human Interface Technology (HIT) Laboratory Technical Report (1989)
18. LaViola J., Joseph J., Daniel A. F., Daniel F., and Robert Z.: Hands-free multi-scale navigation in virtual environments. In: Proceedings of the Symposium on Interactive 3D Graphics, pp. 9–15 (2001)
19. Glueck, M., Sean A., Azam, K.: DeskCube: using physical zones to select and control combinations of 3D navigation operations. In Proceedings of the Spring Simulation Multiconference Society for Computer Simulation International, p. 200 (2010)
20. Caputo, Fabio M., Irina M.C., Davide C., Marco D.S., Andrea, G.: Gestural interaction and navigation techniques for virtual museum experiences. In: AVI CH, pp. 32–35 (2016)
21. Kim, J.-S., Denis, G., Kreimir, M., Francis, Q.: Finger Walking in Place (FWIP): a traveling technique in virtual environments. In: International Symposium on Smart Graphics, pp. 58–69 (2008)
22. Radkowski, R., Christian, S.: Interactive hand gesture-based assembly for augmented reality applications. In: ACHI: The Fifth International Conference on Advances in Computer Human Interactions, IARIA, pp. 303–308 (2012)
23. Bowman, D., Ernst, K., Joseph, J., LaViola, J., Ivan, P.: 3D User Interfaces: Theory and Practice, Course Smart eTextbook. Addison-Wesley, Boston (2004)
24. Reisman, Jason L., Davidson, Philip L., and Je erson Y. H.: A screen-space formulation for 2D and 3D direct manipulation. In: Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology, pp. 69–78 (2009)
25. Phung, S.L., Abdesselam, B., Douglas, C.: A novel skin color model in YCbCr color space and its application to human face detection. In: Proceedings of International Conference on In Image Processing, vol. 1, pp. 1–I (2002)
26. Muhammad, R., Sehat, U., Sami, U.R., Ihsan, R.: Image based recognition of Pakistan sign language. *J. Eng. Res.* **4**(1), 21–41 (2016)
27. Konig, W.A., Roman, R., Harald, R.: Interactive design of multimodal user interfaces. *J. Multimodal User Interf.* **3**(3), 197–213 (2010)
28. Marcus, A., in Jacko, J., and Spears, A.: The Human Computer Interaction Handbook; Fundamentals, evolving Technologies and Emerging Applications (Chapter 18), Taylor and Francis Group, Madison Ave, NY. (2008)
29. Toma, M.L., Florin, G., Csaba, A.: A comparative evaluation of human interaction for design and assembly of 3D CAD models in desktop and immersive environments. *Int. J. Interact. Des. Manuf. (IJIDeM)* **6**(3), 179–193 (2012)
30. Seth, A., Judy, M.V., James, H.O.: Virtual reality for assembly methods prototyping: a review. *Virtual Real.* **15**(1), 5–20 (2011)
31. Chen, C.J., Ong, S.K., Nee, A.Y., Zhou, Y.Q.: Haptic based interactive path planning for a virtual robot arm. *Int. J. Interact. Des. Manuf. (IJIDeM)* **4**(2), 113–123 (2010)
32. Valentini, Pier P., Eugenio, P.: Accuracy in fingertip tracking using leap motion controller for interactive virtual applications. *Int. J. Interact. Des. Manuf. (IJIDeM)* **11**(3), 641–650 (2017)
33. De Luca, Livio, Tudor, D., Emilie, P., Dominique, L., Michel, B.: A complete methodology for the virtual assembling of dismantled historic buildings. *Int. J. Interact. Des. Manuf. (IJIDeM)* **8**(4), 265–276 (2014)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.