Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 218 (2023) 488–503

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

An Intelligent Virtual Environment for Designers with Reduced Motor Abilities

Rahul Bhaumik[a], Tarun Kumar[b]*, Unais Sait[c]

[a]*PES University, Outer Ring Road, Bengaluru, 560085, India*
[b]*Indian Institute of Science, CV Raman Road, Bengaluru, 560012, India*
[c]*Free University of Bozen-Bolzano, Piazza Università, 1, 39100 Bolzano BZ, Italy*

Abstract

Conventional CAD modelling software demands substantial utilisation of input modalities like the keyboard and mouse for creating 3-dimensional (3D) models. The dexterity measures involved in controlling input modalities could pose challenges to users with motor disabilities—including the inability to move their limbs, particularly their upper and lower arms, and fingers, due to traumatic damage or congenital problems. In order to meet these challenges, this paper proposes a virtual reality (VR)-based medium to help users with motor disabilities build simple 3D models for architectural design. The concept of operating buttons using head-gaze in the VR environment has been utilised to perform scaling—a 3D object manipulation method—to create simplified building models. Moreover, navigation in the VR space using tilting of the head has been employed with the user seated on a revolving chair, thus eliminating the need for any limbic movement. Unity game engine was used to develop two variations of the VR model with a different button layout for creating simple cuboidal volumes mimicking buildings in the virtual environment. Both variations have been tested with 32 individuals against a specific performance indicator (i.e., task completion time) and self-reported metrics, such as the perception of effort applied and degree of visual clutter, followed by retrospective participant feedback sessions. One of the VR application's variants (i.e., variant 1) produced promising results regarding overall usability and effort demand. This paper also proposes a methodological framework for an AI-based, intelligent, and adaptive VR application interface that caters to the user's abilities and pain points in real-time. In the future, this framework could be instrumental in creating a comprehensive gaze-based VR tool for 3D modelling having multiple functions to help users with motor disabilities.

Keywords: Virtual Reality; design tool; intelligent interface; gaze-based input; artificial intelligence

* Corresponding author. Tel.+91-9483742547.
E-mail address: tarunkumar@iisc.ac.in;tarunator1@gmail.com.

10.1016/j.procs.2023.01.031

1. Introduction

The words 'virtual' and 'reality', which have opposite meanings, are combined to form the term 'virtual reality', indicating the type of experience it alludes to. Virtual Reality (VR) is a human-computer interface that displays an artificial (or virtual) environment to a person and induces them to suspend disbelief and consider it real. The two key concepts of 'immersion', which refers to the user's ability to selectively focus on certain information while blocking out other distractions, and 'interactivity', which deals with people's capacity to engage with various events in the virtual world, constitute the fundamental principles of Virtual Reality [1]. VR experience involves devices like Head-mounted displays (HMDs), hand controllers, haptic gloves, etc.

1.1. Application of VR

Virtual Reality in the present age finds its practical applications across many fields like military training, industrial training, and medicine [2]. VR is also widely perceived as an emerging tool for the rehabilitation of individuals with a varied range of disabilities like Parkinson's disease rehabilitation, stroke rehabilitation, orthopaedic rehabilitation, and even balance training and daily-living training [2]. In these situations, VR-based tools have enabled people to transfer motor skills learnt in virtual environments to the real world [2]. VR-based interventions have also been proven safe and feasible for rehabilitating children with various sensorimotor deficits [3].

Apart from wide usage in the gaming and entertainment fields, VR has also been utilised in other creative domains like architecture and design for the visualisation of virtual environments [4], aiding the design process of complex systems like hospitals [5], and in architectural education and teaching [6] [7]. Besides spatial geometry visualisations, VR technology has also been used in the contexts of Building Information Modelling (BIM) and smart buildings. For example, a study by Su et al. talks about a VR tech application which integrates GIS and BIM with VR technology but also highlights the limitations of actively implementing VR in the architecture design process [8]. Also, a study by Sait et al. demonstrates the usage of VR to outline a framework for designing daylight-responsive smart buildings [9].

1.2. Gaze-based interfaces: a review of literature

Eye gaze tracking estimates users' point-of-gaze in a digital interface, and their eyes' movement relative to the head [10]. Users can control the graphical user interfaces (GUI) through their gaze in the present times, as eye gaze tracking has been made possible through devices like portable infrared-based eye trackers, webcams, head-mounted displays, and so on [11]. A study by Agarwal et al. (2019) shows the development and evaluation of interfaces for individuals with severe speech and motor impairment (SSMI) using webcam-based gaze trackers [12]. Gaze-controlled interfaces have been studied for pilots and drivers whose hands are occupied with driving as their primary activity, in addition to people with motor impairments who cannot use peripherals like the mouse, keyboard, or joystick because of restricted limbic mobility [10].

Virtual Reality can involve various input modalities associated with Head Mounted Displays (HMDs) like gaze-based input, hand-controllers, voice or speech-based input, hand gesture detection, and walking detection using sensors (like accelerometers), among others [13]. Gaze interactions specific to VR, which can be both eye-movement based or head-orientation based, have their advantages and shortcomings. The on-device buttons for simulating clicks on the smartphone screen and head-gaze—based on the direction normal to the user's head rather than pupil movements—are the typical modes of user input in low-cost VR solutions like Google Cardboard [14]. Head-gaze finds usefulness in *aiming* tasks, whereas *selection* or *triggering* tasks is predominantly achieved by dwell-time methods or clicking on a controller [15]. Some benefits of eye gaze-based interactions in VR and AR involve enhanced speed, reduced task-load and head movement, which are prominent in larger Field-of-View interfaces [15]. Another study using the *FOVE* HMD, by Qian and Teather (2017), demonstrated that the head-only input method performed better than eye-only and eye-with-head modes in terms of accuracy and error rates [16]. Sidenmark and Gellersen (2019) stated that a combined eye and head interaction technique—enabled through a VR

HMD integrated with an eye-tracker—resulted in greater user control and flexibility in gaze-based pointing and selection tasks [17]. Also, ray-casting-based controller interactions may provide better user experience and interaction speed than *gaze-dwell* and *gaze with trigger* methods in object selection tasks [18]. A study, which evaluated different techniques with varying proportions of the use of gaze and manual input for interaction with menus in VR, revealed that gaze aiming (assisted with button selection) and dwell-based gaze selection required less physical effort despite inducing eye fatigue [19]. Mobile-based VR, which usually employs head-gaze interaction techniques, is affordable, unlike sophisticated HMDs with integrated eye-tracking hardware. Mobile VR finds its way in various contexts like the design of a memory game considering four different parameters—background colour, visual field range, feedback system, and multi-dimensional information transfer—to assess users' interest and satisfaction levels [20]. Mobile VR has also been deployed for creating a virtual shopping experience with speech input and head-gaze [21].

In the context of gaze-based 3-D object manipulations in VR, one study by Yu et al. (2021) demonstrates four eye-gaze supported hand input techniques for object manipulation–predominantly translation–and indicated that gaze input would be more useful in larger spaces with distant objects rather than objects at an arm-reach distance [22]. This study [22] also indicated that head-gaze, a cost-effective alternative, can be adopted for object manipulation, and novel gaze-based methods for object scaling and rotation can also be explored. Another study dealing with object manipulation—scaling, rotation, and translation—in Augmented Reality (using the HoloLens device) states that users preferred head-gaze with a clicker device for interaction with virtual objects [23]. Hence, head-gaze input has been employed in this paper for scaling 3D objects in at least a 3-degree-of-freedom (DoF) VR setup instead of using a high-end VR HMD with an integrated eye tracking system.

1.3. Adaptive and Intelligent User interfaces

The paper also touches upon the concept of Intelligent user interfaces (IUI) to explore the possibilities of a complete user-centric VR application for 3-D modelling tasks, accommodating users with reduced mobility or motor impairments. Maybury (1998) defined IUIs as human-machine interfaces that act, represent, and analyse models of the user, domain, task, discourse, and media to enhance the efficiency, effectiveness, and naturalness of human-machine interaction [24]. IUIs, which generally deal with adapting the interface content based on the user's information and environment, incorporate several disciplines into their development, including artificial intelligence (AI), user modelling, psychology, human-computer interaction (HCI), and others [25]. The terms 'adaptive user interface' (AUI) and IUI are used interchangeably and refer to the automatic adjustment of interface elements by monitoring the user's activity and accounting for any changes in the user's skills, preferences or knowledge [25]. Rissland (1984) argues that for an interface to be 'intelligent', it must have access to the knowledge of the user, user's tasks, the domain of user's task, available tools, interaction modalities, knowledge of how to interact with users, and evaluation knowledge of the interactions [26]. Machine Learning (ML) techniques have been used widely to predict user models for developing IUIs based on the past data of user's interactions, irrespective of shortcomings like incomplete datasets, noisy data, or data type being non-numeric [25]. In the context of VR, studies have been conducted to predict the user's intent of interaction for object selection tasks in a virtual environment by analysing gaze dynamics using logistic regression [27]. Also, emergency response scenarios have been designed in VR and tested for first responders (e.g., firefighters) along with assessing Intelligent User Interfaces that focus on presenting task-relevant informatics through the VR headsets [28]. As described in sections 3.2 and 3.3, a preliminary supervised ML model has been used in this study to determine 'when' to update a VR application's interface configuration based on user activity and feedback.

1.4. Aim of the paper

The paper aims to design an intelligent virtual environment which helps individuals with reduced motor abilities to perform 3D modelling tasks with the least effort. The following objectives were set to achieve the aim:

- To create two UI-based variations of a preliminary design tool in VR for performing basic 3-D object manipulation using head-gaze as the input modality and assess their usability based on the performance metric of the users and user-reported parameters.

- Also, to arrive at a methodological framework for an intelligent and adaptive user interface, employing artificial intelligence (AI) to accommodate the user requirements while concisely incorporating desired functionalities within a VR application.

This paper initially explores the interaction issues of the users with the two VR model variants, embodying two different UI configurations, to propose a methodological framework for creating a user-centric intelligent interface.

2. Methodology

The overall methodology of the paper, as illustrated in Fig. 1, involves the following steps: 1) creating two variants of the gaze-based VR model for basic 3D modelling comprising different button configurations; 2) testing two variants of the VR model with 32 participants; 3) analysing the user-testing data comprising both performance metrics and self-reported metrics of users; 4) predicting the degree of effort exerted by the users using an AI model (more precisely a neural network), and incorporating the AI model in an implementation strategy for a responsive VR interface; 5) proposing a comprehensive methodological framework for adaptive and intelligent interfaces based on user data and attributes. The details of the two variants of the VR model comprising gaze detection, button activation and navigation in the virtual environment, along with the experimental setup and considerations for user testing, are detailed in the following sub-sections.

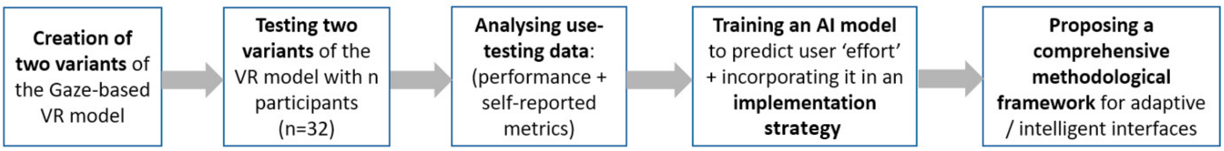

Fig. 1. Representation of the overall methodology.

2.1. Creation of the VR Model and two variants

In this article, the terms 'VR model' and 'VR application' have been used interchangeably, referring to the proposed gaze-based VR application experienced through a head-mounted display (HMD). The environment for the VR model depicts a site for a hypothetical city to be built by the user. The environment comprises two primary elements: a) 'road', where the user can navigate or position themselves to manipulate the geometries representing buildings, and b) 'building blocks', which are cuboids arranged adjacent to each other on either side of the road—analogous to buildings in a real city—whose heights can be altered by gazing on specific buttons in the VR environment.

Two variations of the VR model—variant 1 and variant 2—have been developed in the Unity game engine, embodying differences in the button layout for altering the building heights in the virtual environment. An Android-based smartphone application was developed using the *android build* feature in Unity to experience the VR model using Google Cardboard. However, for testing with the users, a Windows Mixed Reality headset—having 6 degrees of freedom—was used to experience the VR model.

In variant 1, buttons for manipulating the height of the building extrusions in upward or downward directions—labelled as 'Build Up' and 'Build Down'—are explicitly visible to the user at all times in designated zones of the virtual environment. The buttons are positioned in a vertical arrangement—the 'Build Up' button being atop the 'Build Down' button—at the front faces of every building beneath the eye-level of the camera (Fig. 2a). The buttons will be activated if the gaze pointer is within the button area and would get deactivated if the gaze pointer leaves the button area (Fig. 3a).

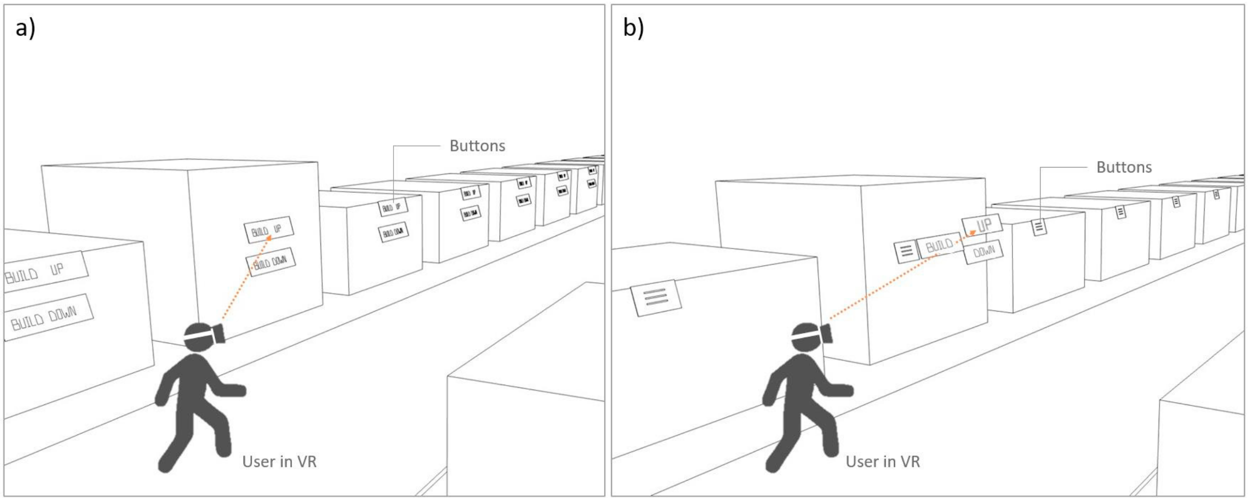

Fig. 2. (a) A Simplified representation of VR model variant 1; (b) a simplified representation of VR model variant 2, both from a 3rd person perspective.

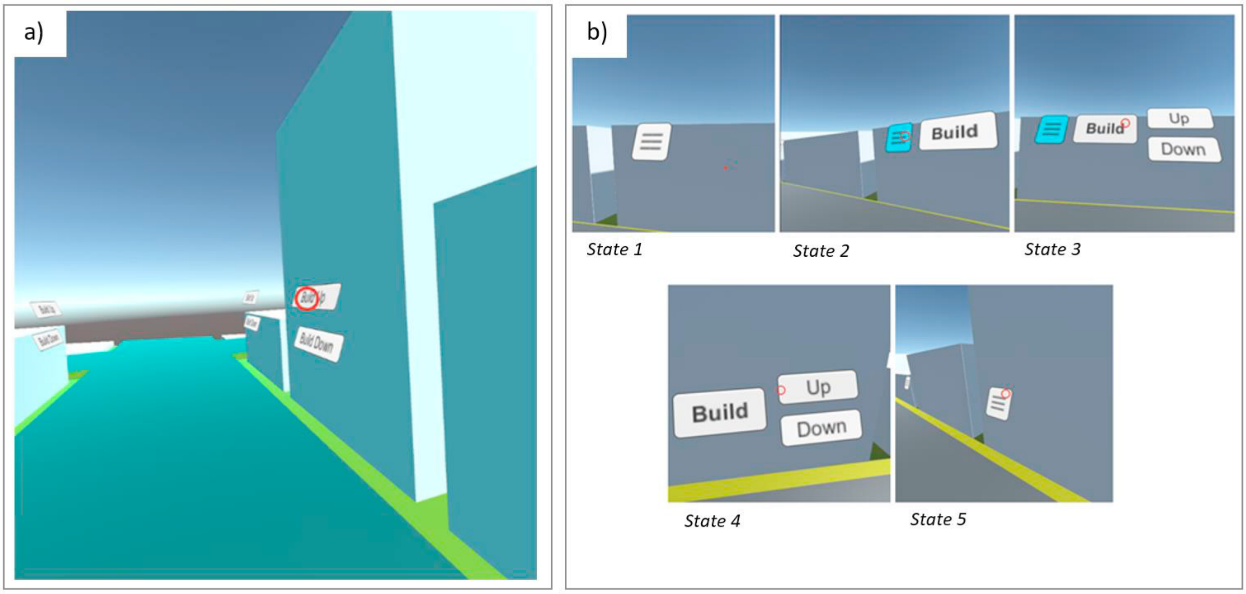

Fig. 3. (a) Sample scene from VR model variant 1 from user's perspective; (b) sample scene from VR model variant 2 with states of button activation from user's perspective.

In variant 2, the buttons for manipulating the height of the building extrusions in the virtual environment are concealed in the hamburger buttons positioned at the front faces of every building (Fig. 2b). The primary objective of employing hamburger buttons is to assess the usability factor of the gaze-based VR application if multiple functions are concealed within a button, to keep the UI visually neat and free from clutter.

The hamburger buttons would be activated when the gaze pointer lies within the button area as the user traverses through the virtual environment. Upon activation of each of the hamburger buttons, a secondary button labelled as 'build' appears. Subsequent gaze activation of the 'build' button leads to the emergence of the tertiary buttons, labelled as 'up' and 'down'. The expanded secondary and tertiary buttons would collapse on the deactivation of the hamburger button. The activated hamburger button—a situation when the secondary or tertiary buttons are expanded, and the button colour of the hamburger button transforms—is deactivated back to its initial state by re-gazing at it. The sequence of events comprising the activation and deactivation of the hamburger, secondary and tertiary buttons are represented in Fig. 4. Screenshots of the button activation sequence from the user's perspective (in variant 2) are represented in Fig. 3b. In this variant, tertiary buttons are the ones which are responsible for the alteration of the building height in the upward or downward direction, when triggered.

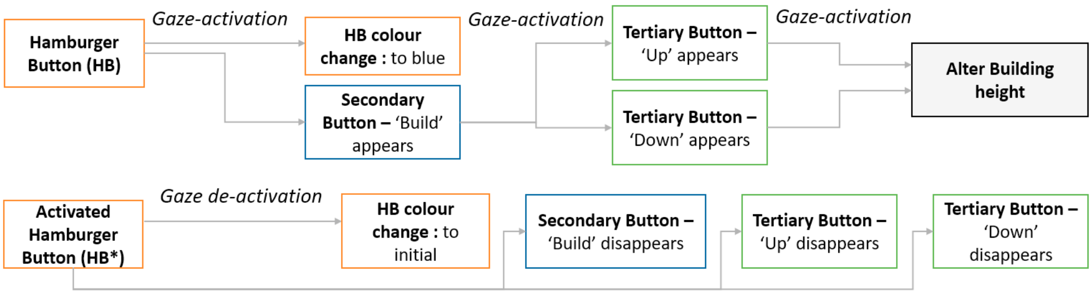

Fig. 4. Schematic of hamburger button activation/deactivation in VR model variant 2.

2.2. Gaze detection in the VR models

As part of rendering a scene in real-time in a virtual reality setup, a series of matrix transformations (or T) is applied to determine the final location of any point P (x, y, and z) on the virtual screen. The formulae for these transformations are given below:

$T = T_{vp}\, T_{can}\, T_{left}\, T_{eye}\, T_{rb}$ (for the left eye) [29]

and,

$T = T_{vp}\, T_{can}\, T_{right}\, T_{eye}\, T_{rb}$ (for the right eye) [29]

Where, T is the overall transformation of a point for viewing in the VR, and each of T_{vp}, T_{can}, T_{right}, T_{left}, T_{eye}, and T_{rb} are transformation matrices with specific transformation characteristics. For example, T_{rb} stands for rigid body transform, T_{eye} transforms the virtual world from the coordinate system of the eye, T_{left} and T_{right} transform the scene leftwards or rightwards for stereoscopic viewing, T_{can} is the canonical view transform through perspective projection, and T_{vp} is the viewport transformation based on the display device resolution [29]. The matrix operations are non-commutative and are performed from right to left.

A reticle pointer lies along the gaze direction of the user wearing the VR headset and is situated at the centre point of the viewing plane of a virtual camera anchored at the midpoint of both eyes. The reticle pointer P (as illustrated in Fig. 5) and other points in the virtual environment would undergo the aforementioned transformations to be rendered in the virtual screens for both left and right eyes in the VR head-mounted display.

The gaze detection in the proposed VR model has been implemented using the GvrReticlePointer prefab [30] from the Google VR SDK for Unity. A prefab in Unity acts as a template to store a GameObject—with all its properties and components—which can be used to create new object instances in a scene [31]. The GvrReticlePointer—responsible for generating a reticle pointer based on the user's gaze—is kept as a child of the 'Main Camera' in the Unity VR model, which makes the pointer continuously follow the user's gaze. Alternatively, the gaze pointer was also implemented using a vector (or a ray) with origin on the 'Main Camera' and direction 'forward' to the 'Main Camera', and applying an opaque shader to a sphere object positioned at the vector. Fig. 6. elucidates the schematic for the gaze pointer and interaction with target buttons.

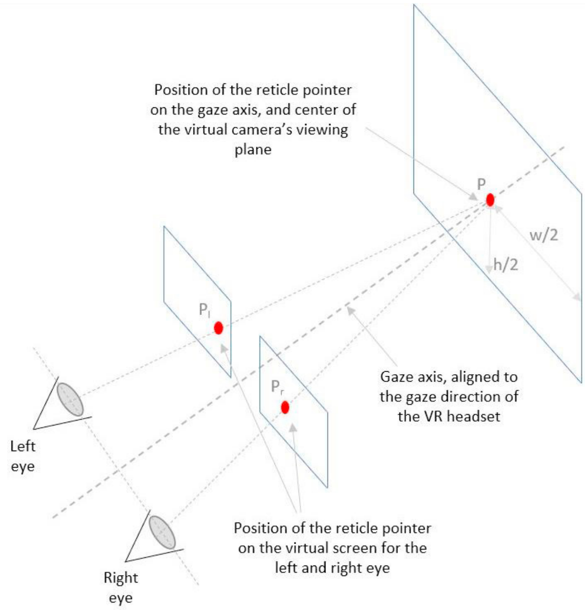

Fig. 5. Rendering of the Reticle pointer for right and left eyes in a VR HMD.

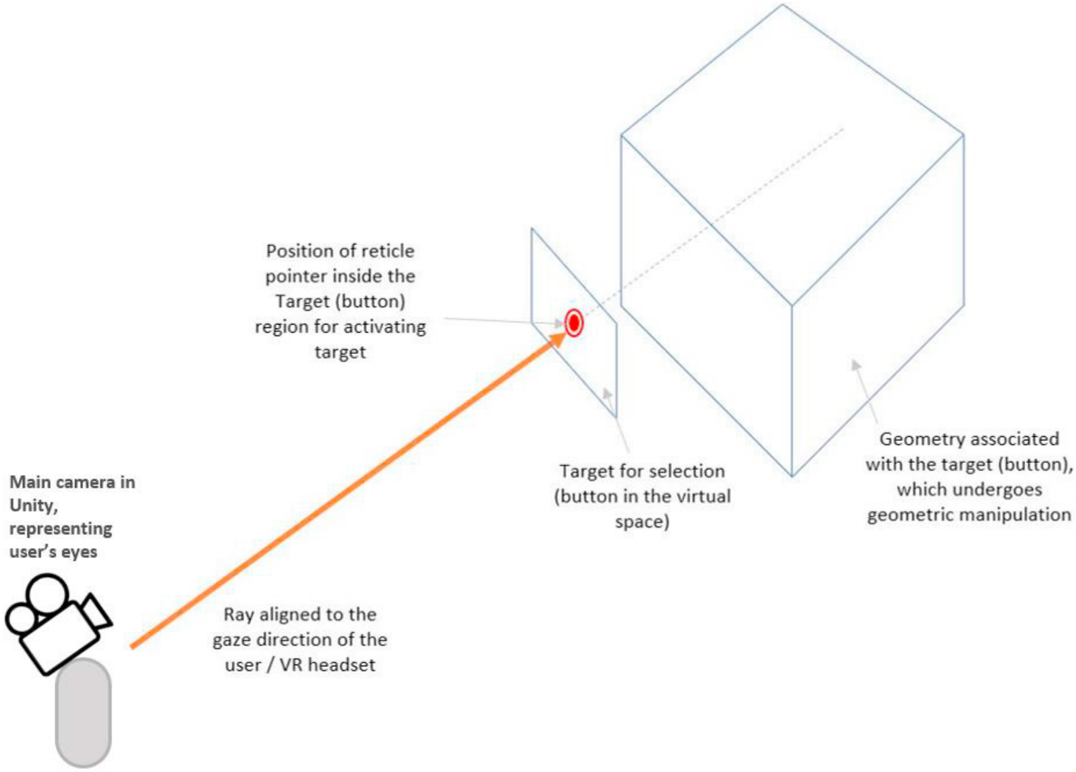

Fig. 6. Gaze/ Reticle pointer lies along the vector forward to the 'Main Camera'.

2.3. Button activation and object manipulation in the proposed VR model

Event Trigger is a class in the Events library of Unity, where the functions for scaling up or scaling down the tagged GameObjects (representing the buildings in the VR world) are executed using the event types ‘Pointer Enter’ and ‘Pointer Exit’. For instance, in model variant 1, whenever the ‘Pointer Enter’ event for the ‘Build Up’ button is triggered—which is symbolic of the gaze pointer entering and being located within the button—the tagged GameObject (or the ‘building’) is continuously scaled up by a vector in the vertical direction. Similarly, the event ‘Pointer Exit’—indicative of the exit of the gaze pointer from the button area—for the button ‘Build Up’ would halt the upward vertical scaling of the building GameObject. Also, the event ‘Pointer Enter’ for the ‘Build Down’ button would lead to constant scaling down of the associated building GameObject, and the ‘Pointer Exit’ event would halt the scaling down operation for the same button.

In the case of model variant 2, the events ‘Pointer Enter’ and ‘Pointer Exit’ for each of the hamburger, secondary and tertiary buttons lead to sequential outcomes ranging from the appearance or disappearance of secondary and tertiary buttons to scaling up or down the buildings in the virtual environment. For example, ‘Pointer Enter’ for the (non-activated) hamburger button would draw up a secondary button (labelled ‘Build’), and ‘Pointer Exit’ would retain the activation of the hamburger button and the secondary button. Subsequently, ‘Pointer Enter’ for the ‘Build’ button would bring up the tertiary buttons labelled ‘Up’ and ‘Down’ in the virtual environment. ‘Pointer Enter’ for the ‘Up’ and ‘Down’ buttons would continually scale up and down the associated building GameObject along the vertical axis, respectively, whereas ‘Pointer Exit’ for the same buttons would stop the axial scaling operations on the GameObjects. ‘Pointer Enter’ for the activated hamburger button would restore it to its normal state (de-highlighted) and retract any secondary or tertiary buttons visible in the VR scene.

2.4. Navigation in the proposed VR model

The navigation in the VR environment in the forward direction for both model variants is achieved through the downward tilting of the user’s head—attached to the head-mounted device—beyond a specified threshold angle (Fig. 7). In the Unity-based VR application, a ‘Character Controller’—a component added to an empty ‘Game Object’—has been created, with the ‘main camera’ as its child. A script for walking, corresponding to forward translational motion in the virtual environment, has been attached to the Game Object, which moves the character controller in the forward direction when the camera rotation in the horizontal direction lies between 15 degrees and 90 degrees. The pseudocode for the same is given as follows:

```
SET walkAngle to 15.0
SET walkSpeed to 3.0
SET cc of type CharacterController
FUNCTION Start()
        cc = GetComponentInParent<CharacterController>()
END FUNCTION
FUNCTION Update()
        IF rotation of camera angle about x direction IS GREATER THAN walkAngle AND rotation of camera angle about x
        direction IS LESSER THAN 90
                SET walkForward to true
        ELSE
                SET walkForward to false
        END IF
        IF walkForward IS true
                CREATE a Vector forward in the forward direction of the camera
                MOVE cc in the direction of forward times walkSpeed
        END IF
END FUNCTION
```

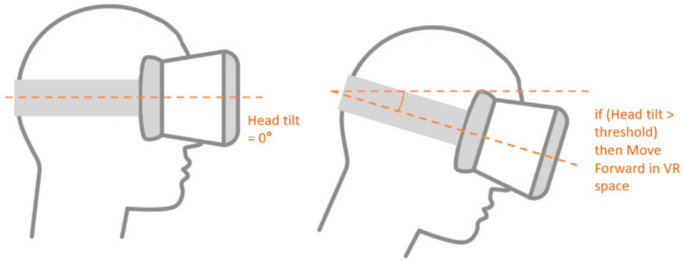

Fig. 7. Head-tilt resulting for forward navigation in the virtual environment.

2.5. Experimental setup

Each of the two variants of the VR model was tested with 32 users, with a 1:1 ratio of females to males and ages ranging from 18 to 46. The participants' experience with CAD or 3D modelling software varied from 0 to 22 years, the average being 5.4 years.

Acer's Windows Mixed Reality Headset—AH101-D8EY—having a Field of View of 100°, a maximum resolution of 2880 x 1440, and a DoF of 6, has been utilised as the VR headset for the experiments. The headset was connected to a desktop bearing an i7-8700K processor with 16 GB RAM and NVIDIA GeForce GTX 1080 8GB GPU.

Firstly, the participants were individually allowed to explore the environment and functionalities in both variants of the VR model. Once acclimatised to the VR setup, they were allowed to perform a common task involving extruding the buildings in staggered order for both application variants. All participants were shown a reference image of buildings that were symmetrically extruded in a staggered order on either side of the road, representing the desired outcome of the modelling in the virtual environment. The participants were seated on a revolving chair, where they could rotate and navigate back and forth in the VR environment and were also asked to restrict any movement of their upper arm, forearm, and hands while performing the task in VR. The participants were only allowed to rotate their heads while seated to navigate the VR environment and trigger the virtual buttons through the head-directed gaze. This setup was primarily done to emulate a possible scenario wherein a physically disabled person—especially with compromised upper limb movement—uses the proposed VR application for 3D modelling.

2.6. User testing of the two VR model variants

The beginning and ending timestamps of each participant's activities for both model variants have been noted and analysed. Subsequently, each participant was asked to respond to a brief questionnaire in a particular model variant upon completing the task. The questionnaire consisted of questions regarding the user perceptions related to the amount of visual clutter in the VR environment encountered by the user and the amount of effort exerted by the user to accomplish a task using gaze-based VR as a medium. The perception of visual clutter reflects the user's opinion on one of the design aspects of the UI, whereas the perception of effort exerted is reflective of the user's performance while executing a task. The participants were allowed to respond to each criterion on a five-point scale, where '1' represents the poor perception of either criterion and '5' represents the best perception. At the end of these activities, each participant was asked to reflect on their experiences with both model variants and voice any concerns or suggestions for creating an enhanced user experience.

3. Results and Discussion

3.1 Results of user testing of two variants of the VR model

It has been observed that the average and median task completion time in VR model variant 1 (i.e., Task 1 or T1) is 78.84 seconds and 64 seconds, respectively. Similarly, the average and median task completion time in variant 2 (i.e., Task 2 or T2) is 93.87 seconds and 78.5 seconds, respectively. The difference between the average task completion times for variants 1 and 2 is 15.03 seconds, and the difference between the median task completion time in both variants is 14.5 seconds. These results indicate that the individual participants, in general, took greater time to accomplish the same activity in variant 2 than in variant 1 (Fig. 8a). Fig. 8b and 8c show the average user-reported scores of the visual clutter of the VR interface and the effort applied for both tasks, respectively. For tasks 1 and 2, the average user-reported scores for visual clutter are 4 and 3.78, respectively, where a score of 1 indicates the highest perceived visual clutter, and 5 indicates the lowest clutter. These findings imply that the participants perceived more visual clutter in variant 2 than in variant 1. Similarly, the average user-reported scores for effort applied are 4.46 and 3.65 for tasks 1 and 2, respectively, where 1 refers to the maximum effort exerted by the user and 5 stands for the minimal effort. This observation indicates that the users felt more effort was needed for task execution in variant 2 than in variant 1. The average user scores for Task 1 were better than Task 2 in terms of overall user experience based on the performance metric (i.e., time) and the two user-reported metrics.

The participants, on average, took higher time to execute the same task in variant 2 because they spent additional time triggering every tertiary button, which only appears after activating hamburger and secondary buttons. On the contrary, for variant 1, the users only had to trigger one pair of buttons, always active in the scene, to manipulate the building heights resulting in quicker task completion. Despite using hamburger buttons and button-nesting as a design strategy to reduce visual clutter in variant 2, the average score for visual clutter for variant 2 was marginally lower than for variant 1. It has generally been observed that when the hamburger buttons for each of the building blocks and all of the associated buttons were activated one after the other (in variant 2), it eventually led to the progression of visual cluttering in the VR environment because there was no provision for the buttons to auto-collapse after being acted upon by the participants. The number of UI elements (i.e., buttons in the case of variant 2) increased in the VR scene as the participants continued to manipulate the building heights, leading to high visual clutter towards the end of the participant's task. Also, the perceived effort for task execution in variant 2 was higher than in variant 1 since the participants had to undergo additional head movements for aiming and selecting the hamburger, secondary and tertiary buttons for manipulating every building geometry. However, the effort on the participant's end for navigation in both variants through head tilting is arguably similar, as there is no difference in the scale or terrain of environments of the two variants.

According to retrospective feedback gathered from the participants after both tasks, around 11 participants (or 34% of the total participants) reported discomfort when selecting the buttons (in both Variants 1 and 2) at the extreme ends from the current user position in the virtual environment. The button and button-label sizes have been kept constant in either of the model variants irrespective of their positioning in the environment. Around 5 participants (16%) expressed difficulty in selecting and retaining the actionable buttons in variant 2 because of the expansion and retraction mechanism of the buttons. Additionally, 4 participants (13%) found it challenging to complete the task while alternately shifting their gaze between the actuator buttons and the buildings' top edges for both variants. Apparently, as the distance between the top edge of buildings and their actuating button gradually increased, the user's gaze traversed increasing distances in the virtual environment. Also, 3 participants (9%) explicitly reported accidental activation of buttons (for both variants) while navigating the virtual environment. Two participants suggested automatic retraction of the previously expanded buttons in variant 2, when not in use, to minimise any added effort on the user's side to voluntarily retract them.

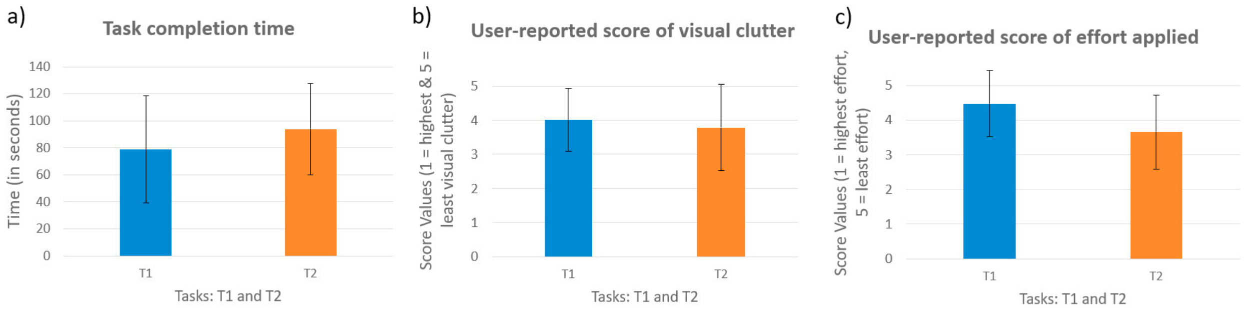

Fig. 8. (a) Bar graphs of average task completion times of 32 participants for tasks 1 and 2; (b) Bar graphs for user-reported scores of visual clutter for T1 and T2; (c) Bar graphs for user-reported scores of effort applied for T1 and T2. Error lines in each case represent Standard deviations.

For the two tasks, T1 and T2, Fig. 9 displays a Radviz visualisation (using Orange data mining software) [32] of the users' data for the three parameters of task completion time, effort applied, and visual clutter. The three parameters are represented in this visualisation as a 2-dimensional projection on a unit circle, with all variable values scaled between 0 and 1. The probability of the location of the data points belonging to T1 and T2 is represented by the coloured regions in the visualisation. T1 points are primarily associated with the perception of least effort applied, whereas T2 points are characterised by higher task completion times or perception of least visual clutter.

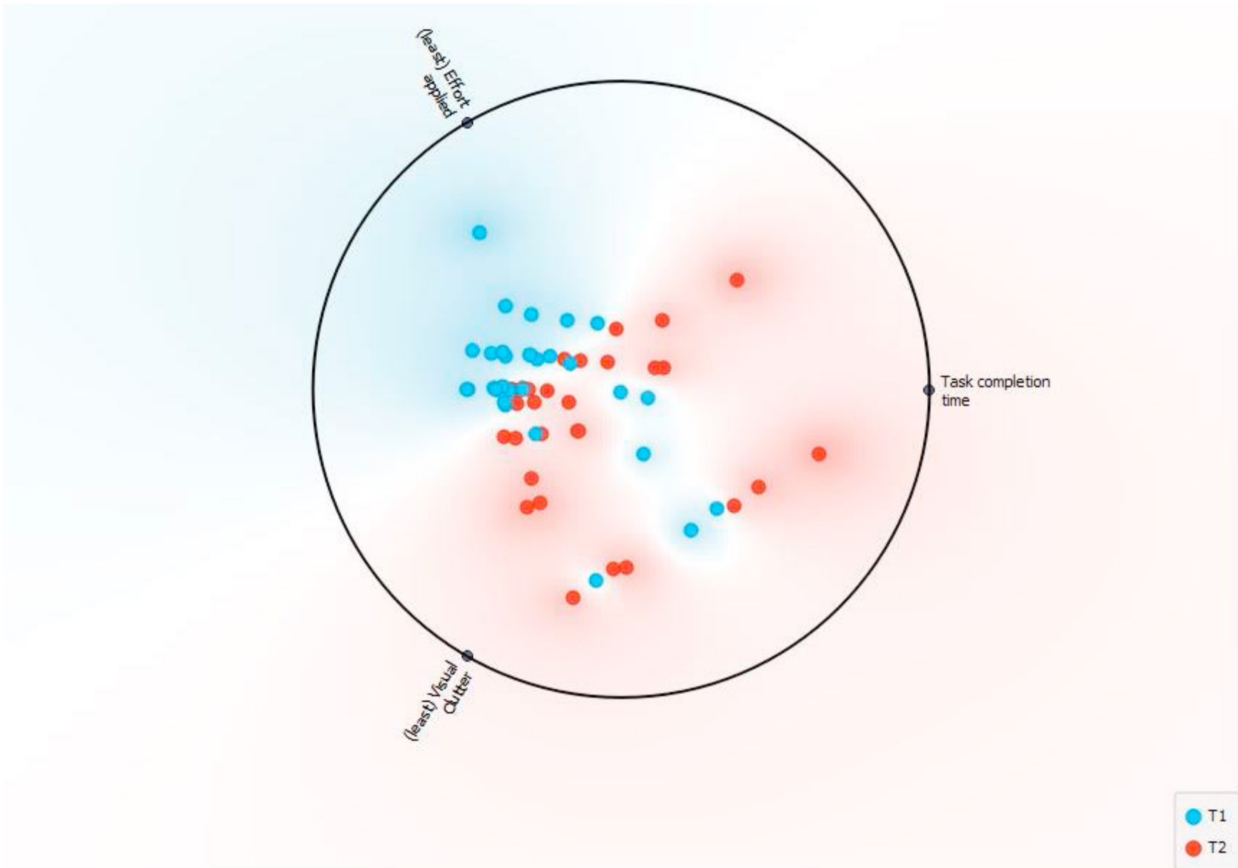

Fig. 9. Radviz visualisation of task completion time, effort applied, and visual clutter for tasks T1 and T2.

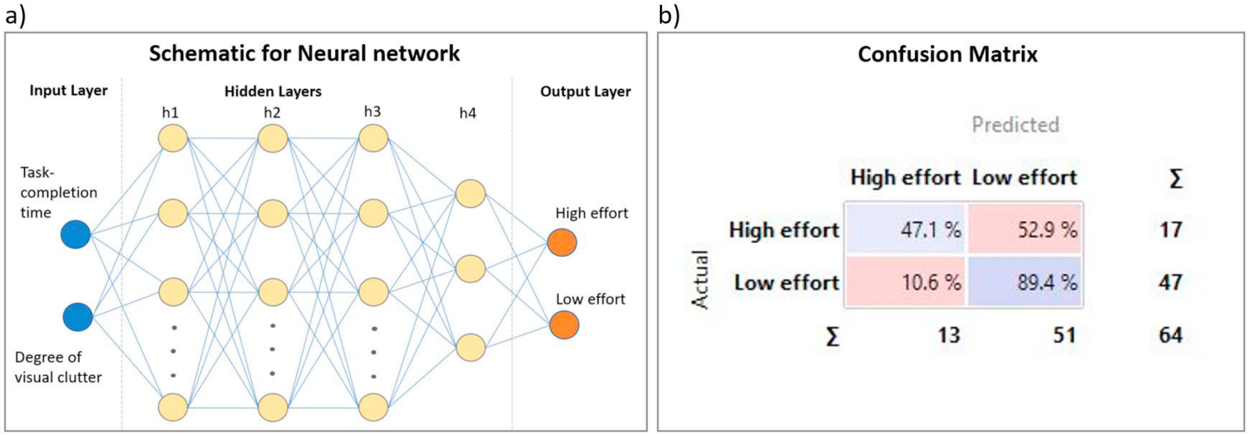

Fig. 10. (a) Schematic for the neural network for predicting effort degree; (b) Confusion matrix for the neural network.

3.2 An AI model for predicting the user effort

The perceived effort applied by a person to complete a task in the VR interface reflects the users' physical exertion and any resistance provided by the interface during task execution. The permissible rotational movement of the head (about the spinal axis) is a prime contributing factor toward the physical effort perception, which needs to be kept to a minimum, especially if the application is meant to be used by motor-impaired users. In the present context, spatial location and arrangement of objects and interactable entities (buttons)—partly reflected in visual clutter perception—influence users' head movements for navigation and selection tasks. In addition, longer task completion times suggest that users spent more time aiming, selecting, and navigating to complete a task due to the virtual environment's design, which might influence the effort perception of the users. Hence it was attempted to predict the degree of effort applied from the task completion times and the perceived degree of visual clutter, based on the data of 32 participants for both tasks T1 and T2, using a Neural Network (NN) model. The scores for the 'effort applied' parameter were set as the target variable (or label) in the existing dataset. This parameter was categorised into two groups:—a) 'High effort' for participant scores less than 4; b) 'Low effort' for scores of 4 and 5. The *input layers* of the NN correspond to the visual clutter scores and task completion times, and the *output layer* indicates the category of the 'effort applied' (Fig. 10a). The neurons in the hidden layer of the NN are in the configuration of {30, 20, 10, 3}, with ReLu as the activation function, Adam as the solver, and the regularisation parameter (α) as 0.0001. The NN model yielded: a) a classification accuracy of 0.78, b) an F1 value of 0.771, c) a precision of 0.768, and d) a recall value of 0.781 over a stratified 5-fold cross-validation dataset sampling. Fig. 10b presents the confusion matrix of the NN model, with the predicted results represented as a percentage of the actual data.

3.3 An implementation strategy for the VR application

An AI model predicting the effort exerted by a user based on the performance metrics and real-time user feedback can be instrumental in deciding whether a particular UI version—with a certain configuration of buttons, menus, and information—can be retained or modified in real-time based on the users' requirements. Fig. 11 shows an implementation strategy for the present VR application, wherein the decision to update the current interface configuration would be made if the AI model predicts the degree of the user's effort as 'high'. Based on this strategy, a complex task undertaken by the user can be perceived as a set of sequential sub-tasks, which can be completed to obtain quick feedback and real-time user performance metrics for the AI model. Alternatively, user data can be obtained amid a task-in-progress to predict the effort exerted by the user in real time.

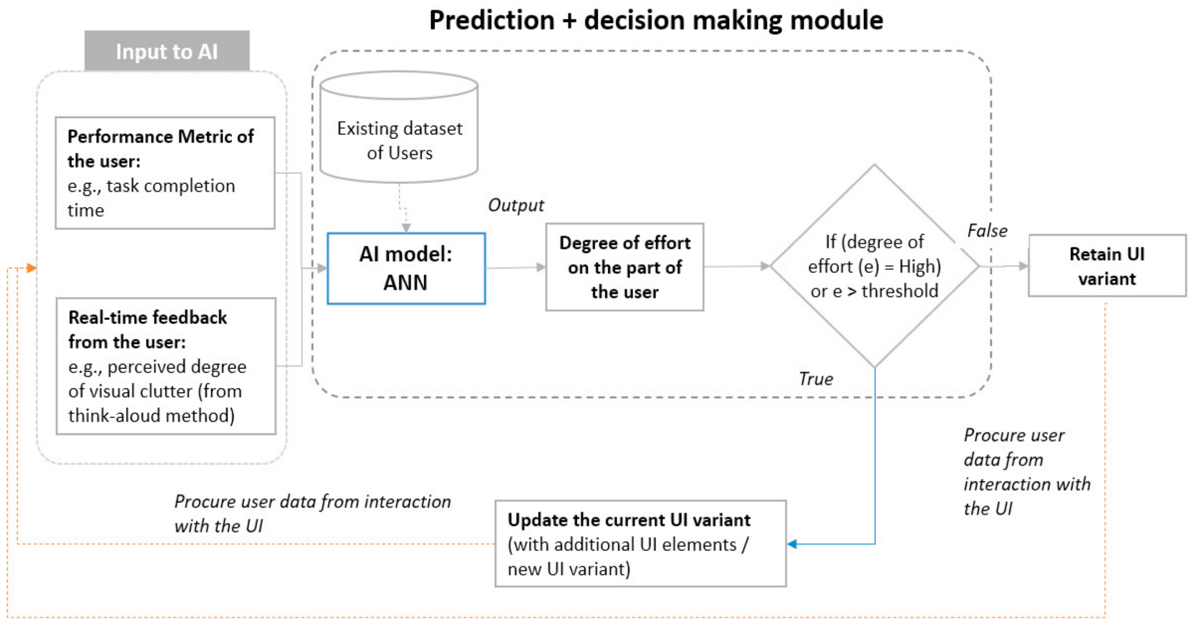

Fig. 11. Implementation strategy incorporating neural network (NN) for effort prediction for the proposed VR application.

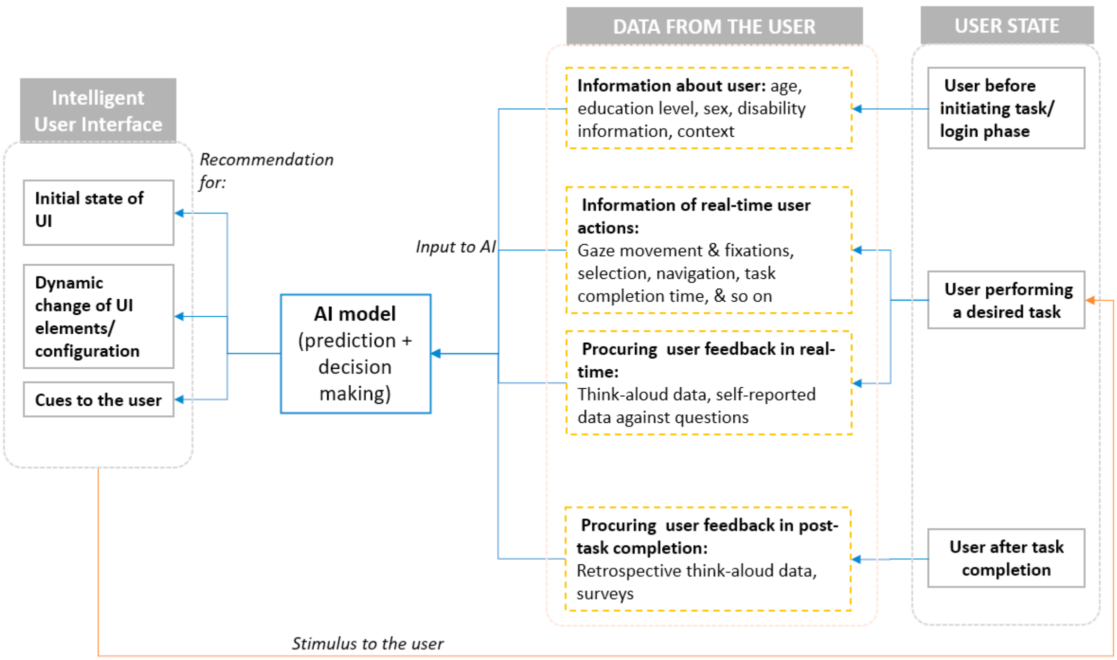

Fig. 12. Schematic for the methodological framework for intelligent and adaptive user interfaces.

In the current scenario of a gaze-based VR for basic 3D modelling, if the user experiences high task completion times or high visual clutter for a specific interface configuration (say variant 2), it would be updated with a UI variant (say variant 1) that requires less effort from a wide range of users. In this case, variation in the UI versions involves pertinent transformations in the button layouts for the gaze input-based VR application. Collecting real-time user feedback has the potential drawback of interrupting the natural task flow of the user and making the user uncomfortable, which can be minimised by keeping the instances of collecting feedback to minimal, adopting a multi-modal way of obtaining user input (e.g., voice input from a think-aloud method), or observing user behaviour instead of collecting obtrusive feedbacks.

In the supervised learning method, the user's data must be obtained from a single UI configuration and multiple UI configurations to create an extensive dataset. Also, data from a user for an interface configuration can be collected broadly at three stages: task initiation stage, task execution stage and post-task-completion stage to assess the suitability of the UI and overall experience of the user. In order to develop a variety of user-centric VR applications, including a future intelligent version of the current VR application, a comprehensive methodological framework incorporating data across user journey stages has been proposed in the following sub-section.

3.4 A methodological framework for user interface development

One of the primary objectives of an intelligent and adaptive user interface (UI) is to become receptive to the user's needs and reconfigure itself to address them. Figure 12 illustrates how data from the user at different stages—task initiation, task execution, and post-task completion—can be gathered and fed into an AI model. The AI model can then provide recommendations for reconfiguring the UI in terms of its initial stage, changing UI elements in real-time, and giving cues to the user during task execution. Information about the user comprising user attributes, like age, gender, education level, disability levels, and others, procured at the task initiation phase can be instrumental in selecting the initial state of the UI. During the task execution phase, two types of data can be procured, one related to the real-time user actions or user performance metrics—like eye gaze movements and fixations, task completion times, selection and clicks, etc.—and the other being user feedback or real-time self-reported data collected using the think-aloud method or short surveys. User information, post-task completion phase, would involve user feedback through surveys, retrospective think-aloud methods or interviews. Based on extensive user data, the AI can continually adapt the interface to the users' current requirements from the initial configuration to the task completion stage, with adequate user permissions. Moreover, the AI model in this user data-driven framework can learn from all instances of the past interaction and feedback data of an individual with a specific interface, ongoing interaction data of an individual, or past interaction data from multiple individuals on a single interface, to generate the most appropriate UI configuration for a particular user context.

4. Conclusion

This paper proposes a head gaze-based VR application for basic 3D modelling involving 1-dimensional scaling of 3D geometries employing two variations of UI concerning the button layout. The two variants of the VR application demonstrated head rotation-based navigation and head-directed activation of the visible buttons and hamburger-concealed buttons in the virtual environment. The VR variants were tested with 32 participants and analysed against user performance and self-reported metrics. Based on the user feedback on both UI variants, variant 1 performed better in terms of overall usability, and an AI model based on neural networks was trained to predict the 'effort' applied by the user in a VR interface. This prediction model is based on two input metrics: a) task completion time, a performance metric, and b) the visual clutter degree in the UI, a user-reported metric. Subsequently, a comprehensive AI-based framework has also been proposed for designing an intelligent interface based on the data from various user journey stages.

In the future, the testing of the VR application with actual motor-disabled individuals is planned to learn about their core pain points while interacting with the virtual environment. Additionally, the development of an adaptive interface for the VR application based on the proposed AI-enabled strategy is planned for testing. Though the proposed AI model to predict the user 'effort' is based on supervised learning, reinforcement learning could also be explored in the future, as the user behaviour and actions constantly change in real-time while acting upon an interface.

The significance of the adaptive AI-based Virtual Reality (VR) interface is that it can potentially benefit users with reduced motor abilities. The physical effort required to operate any interface should be kept to the minimum, especially for individuals with any degree of motor disability, to make the interface universal in nature. Though gaze-based VR interfaces can significantly curtail the need for limbic movement of the users for performing tasks, an AI-enabled VR application can further configure the virtual environment by predicting the effort exerted by the users and assessing the suitability of an interface for an individual with specific limitations.

References

[1] Zheng, J. M., K. W. Chan, and Ian Gibson. (1998) "Virtual reality." *IEEE Potentials* **17(2)**: 20–23.

[2] Holden, Maureen K. (2005) "Virtual environments for motor rehabilitation." *Cyberpsychology & behavior* **8(3)**: 187–211.

[3] Laufer, Yocheved, and Patrice Tamar L. Weiss. (2011) "Virtual reality in the assessment and treatment of children with motor impairment: a systematic review." *Journal of Physical Therapy Education* **25(1)**: 59–71.

[4] Popovski, Filip, Igor Nedelkovski, and Svetlana Mijakovska. (2014) "Generating 3D model in virtual reality and analysing its performance." *International Journal of Computer Science & Information Technology (IJCSIT)* **6(6)**: 123–128.

[5] Chías, Pilar, Tomás Abad, Manuel de Miguel, Gonzalo García-Rosales, and Ernesto Echeverría. (2019) "3D Modelling and Virtual Reality Applied to Complex Architectures: An Application to Hospitals' design." *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* **42**: 255–260.

[6] Horne, Margaret, and Emine M. Thompson. (2007) "Virtual Reality and 3D modelling in built environment education." *CONVR2007 7th Conference on Construction Applications of Virtual Reality*.

[7] Kreutzberg, Anette. (2014) "New virtual reality for architectural investigations." *Proceedings of the 32nd eCAADe Conference* **2**: 253–260.

[8] Su, Ping, and Shuo Wang. (2012) "Virtual reality practice in architecture design." *2012 IEEE Symposium on Electrical & Electronics Engineering (EEESYM)*: 98–101.

[9] Sait, Unais, Gokul Lal KV, Tarun Kumar, Rahul Bhaumik, and Kriti Bhalla. (2019) "A framework outlining a daylight responsive model for smart buildings." *Journal of Physics: Conference Series* **1343(1)**: 012166.

[10] Krishna Sharma, Vinay, Kamalpreet Saluja, Vimal Mollyn, and Pradipta Biswas. (2020) "Eye gaze controlled robotic arm for persons with severe speech and motor impairment." *ACM Symposium on Eye Tracking Research and Applications*: 1–9.

[11] Murthy, L. R. D., M. D. Babu, V. Yelleti, K. P. S. Saluja, and P. Biswas. (2020) "Eye-Gaze Interface to Operate Aircraft Displays." *Proceedings – The European Test and Telemetry Conference – ettc2020*: 134–140.

[12] Agarwal, Ayush, D. V. JeevithaShree, Kamalpreet Singh Saluja, Atul Sahay, Pullikonda Mounika, Anshuman Sahu, Rahul Bhaumik, Vinodh Kumar Rajendran, and Pradipta Biswas. (2019) "Comparing Two Webcam-Based Eye Gaze Trackers for Users with Severe Speech and Motor Impairment." In A. Chakrabarti (eds) *Research into Design for a Connected World. Smart Innovation, Systems and Technologies*, Singapore, Springer

[13] Khan, Rimsha, Farooque Azam, Sheeraz Ahmed, Waseem Anwar, Rudeema Chughtai, and Anum Farid. (2020) "Comparative Analysis of Interaction Techniques in Virtual Reality." *2020 IEEE 23rd International Multitopic Conference (INMIC)*: 1–6

[14] Ma, Zhiyi, Hongjie Chen, Yanwei Bai, and Ye Qiu. (2020) "Research on the Input Methods of Cardboard." *International Conference on Computational Science and Its Applications – ICCSA 2020, Lecture Notes in Computer Science* **12250**: 18–29.

[15] Blattgerste, Jonas, Patrick Renner, and Thies Pfeiffer. (2018) "Advantages of eye-gaze over head-gaze-based selection in virtual and augmented reality under varying field of views." *Proceedings of the Workshop on Communication by Gaze Interaction*: 1–9.

[16] Qian, Yuan Yuan, and Robert J. Teather. (2017) "The eyes don't have it: an empirical comparison of head-based and eye-based selection in virtual reality." *Proceedings of the 5th Symposium on Spatial User Interaction*: 91–98.

[17] Sidenmark, Ludwig, and Hans Gellersen. (2019) "Eye&head: Synergetic eye and head movement for gaze pointing and selection." *Proceedings of the 32nd annual ACM symposium on user interface software and technology*: 1161–1174.

[18] Nukarinen, Tomi, Jari Kangas, Jussi Rantala, Olli Koskinen, and Roope Raisamo. (2018) "Evaluating ray casting and two gaze-based pointing techniques for object selection in virtual Reality." *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*: 1–2.

[19] Pfeuffer, Ken, Lukas Mecke, Sarah Delgado Rodriguez, Mariam Hassib, Hannah Maier, and Florian Alt. (2020) "Empirical evaluation of gaze-enhanced menus in virtual reality." *26th ACM symposium on virtual reality software and technology*: 1–11.

[20] Kim, Mingyu, Jiwon Lee, Changyu Jeon, and Jinmo Kim. (2017) "A study on interaction of gaze pointer-based user interface in mobile virtual reality environment." *Symmetry* **9(9)**: 189–210.

[21] Speicher, Marco, Sebastian Cucerca, and Antonio Krüger. (2017) "VRShop: a mobile interactive virtual reality shopping environment combining the benefits of on-and offline shopping." *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* **1(3)**: 1–31.

[22] Yu, Difeng, Xueshi Lu, Rongkai Shi, Hai-Ning Liang, Tilman Dingler, Eduardo Velloso, and Jorge Goncalves. (2021) "Gaze-supported 3d

object manipulation in virtual Reality." *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*: 1–13.

[23] Pathmanathan, Nelusa, Michael Becher, Nils Rodrigues, Guido Reina, Thomas Ertl, Daniel Weiskopf, and Michael Sedlmair. (2020) "Eye vs. head: Comparing gaze methods for interaction in augmented Reality." *ACM Symposium on Eye Tracking Research and Applications*: 1–5.

[24] Maybury, Mark. (1998) "Intelligent user interfaces: an introduction." *Proceedings of the 4th international conference on Intelligent user interfaces*: 3–4.

[25] Alvarez-Cortes, Victor, Benjamin E. Zayas-Perez, Victor Huga Zarate-Silva, and Jorge A. Ramirez Uresti. (2007) "Current trends in adaptive user interfaces: Challenges and applications." *Electronics, Robotics and Automotive Mechanics Conference (CERMA 2007)*: 312–317.

[26] Rissland, Edwina L. (1984) "Ingredients of intelligent user interfaces". *International Journal of Man-Machine Studies* **21(4)**: 377–388.

[27] David-John, Brendan, Candace Peacock, Ting Zhang, T. Scott Murdison, Hrvoje Benko, and Tanya R. Jonker. (2021) "Towards gaze-based prediction of the intent to interact in virtual Reality." *ACM Symposium on Eye Tracking Research and Applications*: 1–7.

[28] Spain, Randall, Jason Saville, Barry Lui, Donia Slack, Edward Hill, John Holloway, Sarah Norswothy, Bradford Mott, and James Lester. (2020) "Investigating a Virtual Reality-based Emergency Response Scenario and Intelligent User Interface for First Responders." *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* **64(1)**: 2114–2118.

[29] Lavalle, Steven M. (2019) "The Geometry of Virtual Worlds", in Steven M. Lavalle (eds) *Virtual Reality*, Cambridge, Cambridge University Press

[30] GvrReticlePointer prefab. Retrieved from https://developers.google.com/vr/reference/unity/prefab/GvrReticlePointer. Last accessed 12-06-2022.

[31] Unity Documentation Prefabs. Retrieved from https://docs.unity3d.com/560/Documentation/Manual/Prefabs.html. Last accessed 12-06-2022.

[32] Hoffman, Patrick, Georges Grinstein, Kenneth Marx, Ivo Grosse, and Eugene Stanley. (1997) "DNA visual and analytic data mining." *Proceedings. Visualization'97 (Cat. No. 97CB36155)*: 437–441.