Noname manuscript No.
(will be inserted by the editor)

vDesign: A CAVE-based Virtual Design Environment Using Hand Interactions

Xiaoming Nan · Ziyang Zhang · Ning Zhang · Fei Guo · Yifeng He · Ling Guan ·

Received: date / Accepted: date

Abstract The Cave Automatic Virtual Environment (CAVE) system is one of the most fully immersive systems for Virtual Reality (VR) environments. By providing users with realistic perception and immersive experience, CAVE systems have been widely used in many fields, including military, education, health care, entertainment, design, and others. In this paper, we focus on the design applications in the CAVE. The design applications involve many interactions between the user and the CAVE. However, the conventional interaction tool, the wand, cannot provide fast and convenient interactions. In this paper, we propose *vDesign*, a CAVE-based virtual design environment using hand interactions. The hand interactions in *vDesign* are classified into menu navigation and object manipulations. For menu navigation, we define two interactions: activating the main menu and selecting a menu item. For object manipulations, we define three interactions: moving, rotating, and scaling an object. By using the proposed hand interactions, we develop the functions of image segmentation and image composition in *vDesign*. With the image segmentation function, the designer can select and cut the interested objects from different images. With the image composition function, the designer can manipulate the segmented objects and combine them as a composite image. We implemented the *vDesign* prototype in CAVE and conducted experiments to evaluate the interaction performance in terms of manipulation time and distortion. The

X. Nan · Z. Zhang · N. Zhang · F. Guo · Y. He · L. Guan
Department of Electrical and Computer Engineering, Ryerson University
350 Victoria Street, Toronto, Ontario, Canada M5B 2K3
E-mail: xnan@ee.ryerson.ca

Z. Zhang
E-mail: zhangzyster@gmail.com

N. Zhang
E-mail: ning.zhang@gmail.com

F. Guo
E-mail: f2guo@ee.ryerson.ca

Y. He
E-mail: yhe@ee.ryerson.ca

L. Guan
E-mail: lguan@ee.ryerson.ca

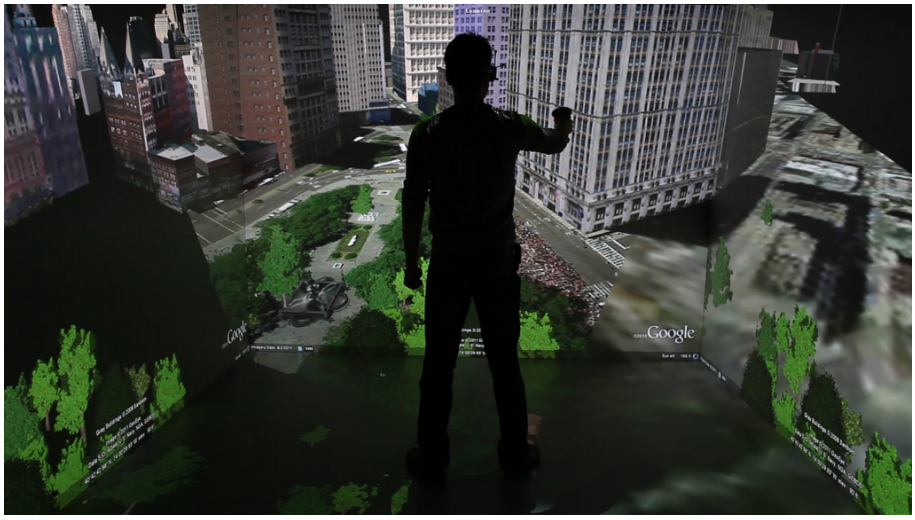


Fig. 1 User in the fully immersive VR system CAVE.

experimental results demonstrated that the proposed hand interactions can provide faster and more accurate interactions compared to the traditional wand interactions.

Keywords Virtual reality · Cave Automatic Virtual Environment (CAVE) · Hand interactions · Virtual design · Markers

1 Introduction

Virtual Reality (VR) is a term used for computer-generated Three-Dimensional (3D) environments that allow users to enter and interact with alternate realities [25]. In VR applications, users can be partially or fully immersed in an artificial 3D world that is completely generated by computers. Depending on the degree of user immersion, VR systems can be classified into non-immersive VR system, semi-immersive VR system, and fully immersive VR system [9], [20], [5]. As the name suggests, the non-immersive VR system, also called desktop VR, uses a conventional graphic workstation to display a 3D environment on a Two-Dimensional (2D) monitor. The non-immersive VR system is an economical solution. However, it cannot provide the realistic perception to users. The semi-immersive VR system supports the feeling of "looking at" the virtual environment. Head Mounted Display (HMD) is an example of the semi-immersive VR system. The fully immersive VR system supports the feeling of "being in" the virtual environment. The Cave Automatic Virtual Environment (CAVE) [10], as shown in Fig. 1, is a fully immersive VR system. Compared to non-immserive and semi-immersive VR systems, fully immersive VR systems provide users a wider field of view and a larger freedom of interactions.

In a CAVE system, multiple stereoscopic projectors are used to project a 3D environment into a room-sized cube [12], which consists of three walls and a floor. A tracking system is used to track the real-time position and orientation of the eyes of the user inside the CAVE. The images output from the projectors are controlled by the real-time tracking data of the user. The user can perceive the true 3D environment through a pair of stereoscopic shutter glasses, which alternately block the left or right eye such that each eye only sees the

Fig. 2 Illustration of the proposed *vDesign* in the CAVE.

corresponding images [11]. The user inside the CAVE can see the virtual objects floating in the air, and also can manipulate them with interaction tools.

CAVE systems have been widely used in many fields, including military, education, health care, entertainment, design, and others. In this paper, we focus on the design applications. Design refers to the creation of a plan for the construction of an object or a system. The design applications cover a wide scope, including interior design, architecture design, urban design, landscape design, product design, and others. Conventionally, designers use the desktop-based design software to do the job. However, the desktop-based design software have the following limitations: 1) The interface of the desktop-based software is not natural user interface (NUI), thus requiring a training process for a beginner to memorize the function of each button. 2) The desktop-based software cannot provide an immersive experience, without which users cannot get a realistic impression of the design. 3) The manipulations of 3D objects in a desktop computer are not convenient, since users cannot perceive the precise position along the axis which is perpendicular to the 2D monitor. By using virtual reality technologies, design applications can be performed in a 3D virtual environment. Designers can freely investigate the construction from different angles and dynamically change the appearance. Compared to the 2D-based design, 3D virtual design environment provides users more realistic experience and richer information.

Design in a 3D virtual environment involves frequent manipulations of virtual objects. For example, an interior designer, who wants to place a virtual sofa in a desired position, needs to perform a number of basic manipulations, such as moving, rotation or scaling, to the virtual sofa. Sometimes, designers want to combine the interested elements from different images into their design. For instance, an interior designer may want to compose multiple interested patterns into the personalized wall paper. In order to achieve these objectives, designers require accurate and frequent manipulations of virtual objects in the CAVE. Traditionally, the user in the CAVE uses the wand, which is a 6 Degrees-Of-Freedom (DOF) interaction tool similar in function to a mouse for a computer, to interact with the CAVE environment. However, the wand is not a convenient and fast interaction tool for object manipulations.

To enhance the quality of user experience and provide convenient interaction tool for designers, we propose *vDesign*, a CAVE-based virtual design environment using hand interactions, which is illustrated in Fig. 2. The user in *vDesign* system wears a marker on each hand. Both markers are tracked by the tracking system. Based on the real-time positions of the markers, we can determine if any interaction is triggered. We define and implement the basic hand interactions for menu navigation and object manipulations in *vDesign*. With these basic hand interactions, we develop two functions, image segmentation and image composition, for designers. In the image segmentation, the designer can select the interested object and cut the selected object from the image. In the image composition, the designer can compose multiple segmented patterns together into a new image and apply it in the future design. We implemented *vDesign* prototype and conducted experiments to compare the performances between the proposed hand interactions and the conventional wand interactions. The experimental results demonstrated that the proposed hand interactions outperform the conventional wand interactions in terms of manipulation time and distortion.

The remainder of this paper is organized as follows: Section 2 discusses the related work. Section 3 presents the overview of the proposed *vDesign* system. In Section 4, we describe the menu navigation and object manipulations with hand interactions in *vDesign*. In Section 5, we present two functions in *vDesign*: the image segmentation and the image composition. In Section 6, we conduct user tests to evaluate the proposed *vDesign* system and compare the performances between the proposed hand interactions and the conventional wand interactions. Finally, the conclusions are drawn in Section 7.

2 Related Work

2.1 Virtual Applications in CAVE

The CAVE system was first introduced by Cruz-Neira *et al.* in [12]. In the CAVE, the user is surrounded by three rear-projection screens, and a fourth projector that is mounted overhead points to a mirror, which reflects the images onto the floor [11]. To generate a stereoscopic visual experience, two off-axis perspective images are projected on screens: one is visible to the left eye, and the other to the right eye. The user can alternatively view each image through each eye by wearing a pair of shutter glasses, which rapidly turned on and off in synchrony with the corresponding images on the screen [14]. A tracking system is used in CAVE to detect the position and the user's orientation. The tracking transmitter is installed on the ceiling of the CAVE, while markers are attached on the stereo glasses. Therefore, the projected images will be automatically adjusted following the user's head motion in real time [19]. Users can walk or jump in the CAVE and everything looks as natural as the real world.

CAVE has been used in a variety of virtual applications, including game engine [18], manufacturing system [31], entertainment [30], information technology [28], and aerospace [4]. Lugrin *et al.* [18] proposed *CaveUDK*, a VR game engine middleware for CAVE, aiming to enhance the immersion experience. Yang *et al.* described Virtual Factory (VF) concept in manufacturing system considering the impact of human factors [31]. A CAVE-based virtual theater system was proposed to allow actors to animate virtual characters in real time, resulting in a more flexible and interactive theatrical performance experience [30]. Wijayasekara *et al.* [28] presented a CAVE-based immersive visual data mining system, which allows users to explore and interact with the multi-dimensional data in a natural and intuitive way. Aerospace scientists are often faced with the difficult task of interpreting the sizes and rel-

ative positions of objects in an environment when viewing an image of the environment on computer monitors or prints. To address this challenge, a cylindrical immersive display system, known as *Stage*, was built to provide a more accurate awareness of the environment [4].

2.2 Human Computer Interactions in CAVE

Various interaction and object manipulation techniques for CAVE have been examined in the literature. Traditionally, the so called Flystick, a wand type remote control, is used with various buttons. Individual button and a combo of buttons correspond to certain actions such as menu selection, directional up/down, etc. An experimental study on interaction in CAVE showed that virtual hand is helpful in reducing the interaction errors in the CAVE [27]. Abramyan *et al.* used two types of wands (Nintendo Wii controller and Nunchuk joystick) in angle viewing and manipulation control [4]. Wand was also used in a virtual table tennis game as the hand tracker to mimic the racket in Li *et al.*'s work [17]. Koike and Makino proposed a 3D solid modeling system using wand to draw sketches on the screen, and a 3D model was then converted from the basic sketch [16].

Besides the traditional point-based wand controller in CAVE, other researchers proposed various approaches in transferring command from 2D touch screens to the 3D CAVE. Kim *et al.* clicked button and drew arrows on the iPhone/iPod for touch screen to command the CAVE in menu selection and navigation [15]. In data mining, Prachyabrued *et al.* developed an interface based on iPod touch technology, in order to achieve complicated and cluttered 3D data visualization and manipulation in the CAVE environment [21]. In this interface[21], occluded data in the 3D space is simplified to the 2D overview and presented in iPod, while the selection and navigation commands executed at the iPod are transferred back to the CAVE system. Song *et al.* proposed a set of interaction command based on iPod touch, including volume data slicing, drawing, and annotation [26].

Cooperative object manipulations in virtual environments were investigated in [24]. Symmetric action integration was superior for the task when both participants had to perform similar actions, while asymmetric integration was superior when participants had to move the object in different ways [24]. Wu *et al.* introduced approaches to navigating and manipulating objects in a collaborative virtual environment that engage tangible objects and an interactive table interface [29].

3 vDesign System Overview

In this paper, we propose the *vDesign*, a CAVE-based virtual design environment using hand interactions. Fig. 3 gives the structural layers of the proposed *vDesign* system. In *vDesign*, we propose to use hand interactions such that the user in the CAVE can perform various tasks in a natural and intuitive way. The proposed hand interactions include menu navigation and object manipulation, such as moving, rotating, and scaling a virtual object. Based on the hand interactions, we can develop a series of functions, such as image segmentation and image composition, for the designers. Design is a very big area. There are numerous design applications in practice. With the basic object manipulations, we can easily add more function-level applications into the *vDesign* framework according to the designer's practical requirement. The proposed *vDesign* framework defines the specifications of 3D object op-

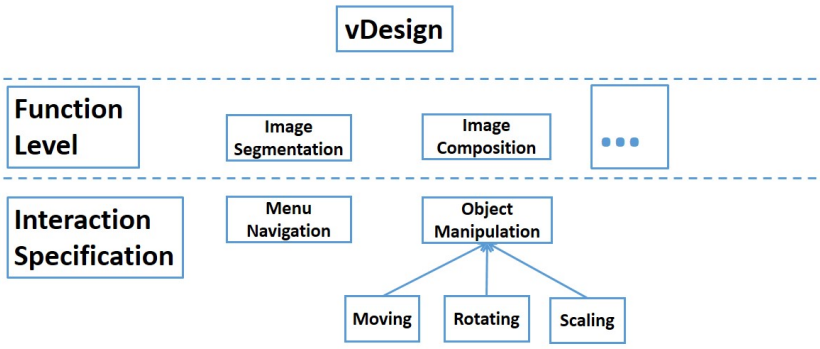
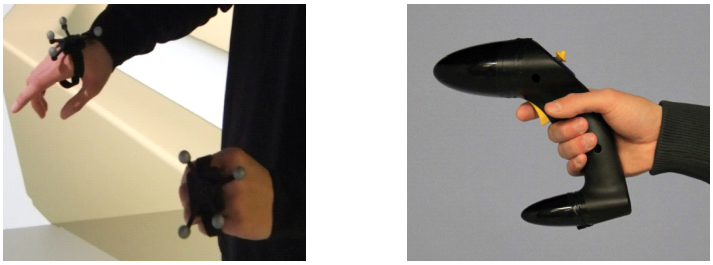


Fig. 3 Structural layers of the proposed *vDesign* system.

erations (such as moving, rotating, and scaling 3D objects), which are critical for immersive 3D design tasks.

In *vDesign*, the user can interact with the CAVE by wearing a marker on the left hand and another marker on the right hand, as shown in Fig. 4(a). The left marker represents the left hand and the right marker represents the right hand. We will hereafter refer to the right (left) marker as the right (left) hand. Both markers will be tracked by the tracking system in real time. Based on real-time positions of the markers, we can calculate the distance between the two markers, the distance between a marker and a menu item, and the distance between a marker and a point of a virtual object in real time. The various distances can be then used to trigger different actions, including menu navigation and object manipulations. Conventionally, the wand, as illustrated in Fig. 4(b), is employed as the interaction tool in the CAVE. In *vDesign* system, we use hand interactions instead of the wand, since the hand interactions can provide easier and faster operations in the CAVE.



(a) (b)

Fig. 4 Two interaction techniques: (a) hand interactions via markers, and (b) wand interactions

We plan to use hand interactions in a series of applications, including virtual gaming, virtual learning, virtual working, virtual presenting, virtual designing, and many others. *vDesign* is our first project under the hand interaction framework. The menu navigation is illustrated in Fig. 5. Once the user inside the CAVE activates the main menu via the right hand, the level-1 menu will be floating in front of the user. The user can navigate the menu from the low level to the high level by touching the menu item via the right hand, similar to the touch screen functions in tablets. After completing the menu selections, the user will perform manipulations on the selected object. For example, the interior designer, who wants to put a desktop computer in an appropriate position in an office room, will first select the menu item of the desktop computer, as shown in Fig. 5, before he or she can manipulate it.

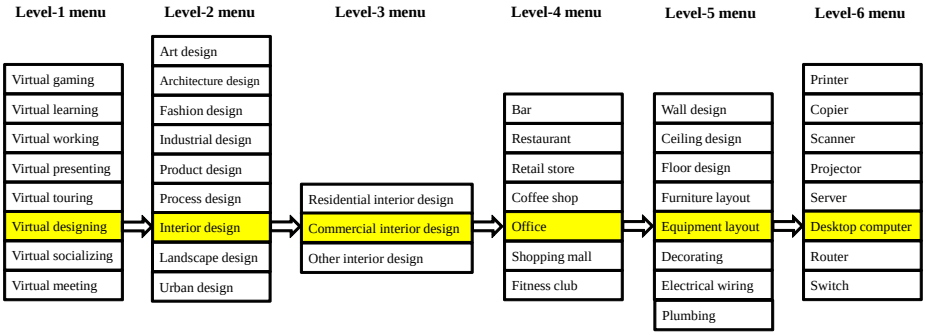


Fig. 5 Illustration of the menu navigation in *vDesign* system

We also develop two useful functions, which are image segmentation and image composition, in *vDesign*. When the function of image segmentation is activated, a series of photos will be displayed in the 3D space and the user can pick up one by touching the photo. The selected photo will be automatically enlarged in front of the user. The user can draw the strokes on the interested object with the right hand, and on the unrelated background with the left hand. With the strokes as seeds, a graph-cut based image segmentation will be performed to cut the interested object from the image. The user can place additional strokes in an iterative way to improve the segmentation result. After image segmentation, the user can use the image composition function to place the segmented objects in a proper position on a new image by moving, rotating, and scaling them with hands.

4 The Proposed Hand Interactions

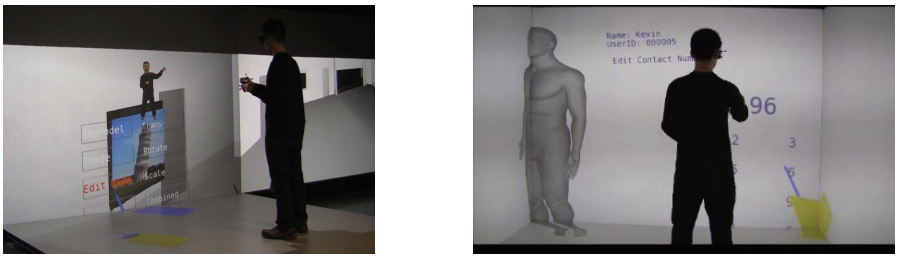
The proposed hand interactions in *vDesign* system can be classified into two classes: *menu navigation* and *object manipulations*. In this section, we will describe the design of the hand interactions.

In *vDesign*, an interaction is triggered based on the positions of the markers and the virtual objects. A marker in the CAVE can be tracked by the tracking system. The tracking system provides the 6-DOF tracking data in the format of $(x, y, z, \eta, \theta, \phi)$, for any marker in real time. The coordinates (x, y, z) represent the position of the marker in the 3D space, and (η, θ, ϕ) are three Euler angles, which represent the rotation of the marker relative to its local coordinate system. We use an *interaction thread* to monitor the interactions. We divide

the time into multiple time slots of equal length. The length of the time slot is denoted by τ. The interaction thread maintains a sliding window which stores the tracking data of the markers during the time period from $(t_c - T_s)$ to t_c where t_c is the current time and T_s is the length of the sliding window. At the beginning of each time slot, the interaction thread does the following tasks in sequence: 1) it reads the tracking data of the markers from the tracking system; 2) it updates the sliding window, 3) it checks the conditions for all of the pre-defined interactions, and 4) if the condition for any interaction is satisfied, it sends the trigger signal immediately to the main program which will handle the interaction event.

4.1 Menu Navigation

Menus are common interfaces in 2D graphical user interfaces (GUI). But *should menus be used in the VR system*? As common opinions, the traditional windows, icons, menus, pointers are not natural command interfaces, which should not be used in VR applications. However, the study in [6] takes a different view that the naturalism of interface should be based on the application and user expectations. In a virtual tour application, users expect a realistic experience to the greatest possible degree, and menus should be avoided to use. However, when user's major goal is the efficient completion of tasks, menus can be used to minimize the time and errors. In *vDesign*, we attempt to achieve a balance between the interaction efficiency and immersive experience. We used the virtual-pen-and-virtual-pad menus, which are reported to be significantly faster than other menus by [6]. Fig. 6(a) illustrates the virtual menus in *vDesign*. Moreover, we introduce the virtual number pad, which can be used to efficiently input numbers in *vDeisng* system. Fig. 6(b) presents the virtual number pad.



(a) (b)

Fig. 6 Illustration of the virtual-pen-and-virtual-pad menus in *vDesign*: (a) virtual menu, and (b) virtual number pad.

Menu navigation in *vDesign* mainly consists of two operations: activating the main menu (e.g., level-1 menu) and selecting a menu item.

The activation of main menu is triggered by the pull-down action performed by the right hand. At the current time t where t is the beginning time of a time slot, the trigger of main menu occurs if $z_t - z_p \geq \rho^a_{th}$ where z_t is the coordinate of the vertical axis perpendicular to the floor at the current time t, z_p is the coordinate at the previous time $(t - \tau)$, and ρ^a_{th} is a distance threshold which is used to determine if the activation occurs. Once the main menu is triggered, it will appear in front of the user.

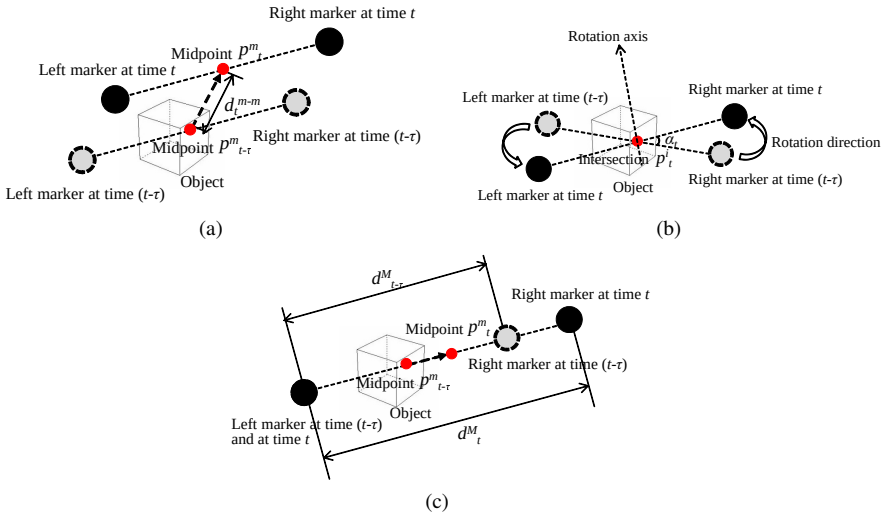


Fig. 7 Object manipulations with the proposed hand interactions: (a) moving an object, (b) rotating an object, and (c) scaling an object

The selection of a menu item is trigger by touching the menu item with the right hand. At the current time t, where t is the beginning time of a time slot, a menu item is triggered if $d_k \leq \rho^s_{th}, \forall k \in [t - T_h, t]$, where d_k represents the distance between the right marker and the center of the menu item at time k, k is the time instant which is at the beginning of a time slot and in the range between $(t - T_h)$ and t, T_h represents the holding duration of the touch, and ρ^s_{th} is a distance threshold which is used to determine if a touch occurs. With the tracking system, we can acquire the real-time position of the right hand. Since the menu is activated in front of the user, the position of each menu item is also known by the system. By calculating the difference between the two positions, we can acquire d_k in real time. Once a menu item is selected, the next-level menu will appear, or the objects to be manipulated will appear.

4.2 Object Manipulations

Object manipulations mainly consist of three interactions: moving, rotating, and scaling an object. The action of moving, rotating or scaling an object is performed in every time slot by two hands together. There are two ways to select a 3D object in *vDesign*. The first way is to select a 3D object by touching the corresponding item in the menu. This selection method allows users to efficiently select the desired object even if it is occluded by some other objects from the user's perspective. The second way is to select an object by touching the object by the right hand and then flipping the left hand. With this selection method, users do not need to navigate menu items. Touching the object with the right hand will change the object to "to-be-selected" status, and flipping the left hand confirms this selection. After selecting an object, the user is free to move, rotate, or scale the object.

The moving action is determined by two factors: *moving direction* and *moving distance*. Let p^m_t denote the position of the midpoint of the segment between the two markers at time t, and let $p^m_{t-\tau}$ denote the position at time $(t-\tau)$. The moving direction of the object is from

point $p^m_{t-\tau}$ to point p^m_t. The moving distance d^{m-m}_t is the distance between point $p^m_{t-\tau}$ and point p^m_t. The moving manipulation is shown in Fig. 7(a). At the current time t where t is the beginning time of a time slot, the object is moved by d^{m-m}_t along the direction from point $p^m_{t-\tau}$ to point p^m_t.

The rotating action is determined by four factors: *rotation plane*, *rotation axis*, *rotation direction*, and *rotation angle*. Let L_t denote the line passing through the two markers at time t, and $L_{t-\tau}$ denote the line passing through the two markers at time $(t-\tau)$. The intersection point between lines L_t and $L_{t-\tau}$ is denoted by p^i_t. The *rotation plane* is the plane containing lines L_t and $L_{t-\tau}$. The *rotation axis* is the line perpendicular to the rotation plane and through point p^i_t. The *rotation direction* is the same to the rotation direction of the two markers. The *rotation angle* is the angle through which line $L_{t-\tau}$ is rotated to coincide with line L_t around the rotation axis along the rotation direction. At the current time t where t is the beginning time of a time slot, the object is rotated by the *rotation angle* α_t around the *rotation axis* along the *rotation direction*, as shown in Fig. 7(b).

The scaling action that we use is *uniform scaling*, which is a linear transformation that enlarges or shrinks the object by a *scale factor* that is the same in all directions. At the current time t where t is the beginning time of a time slot, the *scale factor* is given by $s_t = d^M_t / d^M_{t-\tau}$, where d^M_t is the distance between the two markers at time t, and $d^M_{t-\tau}$ is the distance between the two markers at time $(t-\tau)$. The object is enlarged when $s_t > 1$, shrunk when $0 < s_t < 1$, or kept the same when $s_t = 1$, as shown in Fig. 7(c).

In *vDesign*, two different manipulation modes are provided: the *combined mode* and the *individual mode*. In the combined mode, the interactions of moving, rotating, and scaling are combined together. The user can move, rotate, and scale a 3D object at the same time. In the individual mode, each interaction is performed individually. The combined mode provides users the most freedom to manipulate 3D objects, while the individual mode can be used to realize a specific manipulation and avoid other unintentional manipulations. The user can choose the combined mode or the specific interaction in the individual mode by touching the corresponding menu item.

4.3 Evaluation Metrics for Interactions

The performance of interactions in the virtual environment can be evaluated by two metrics: *manipulation time* and *distortion*. *Manipulation time* is defined as the time spent by the user in manipulating the object from its original state, represented by its position, orientation, and size, to its final state which satisfies the user's requirement in *distortion*. *Distortion* represents the deviation between the target state and the actual state of the object after manipulations. The target state is the state desired by the user after object manipulations, and it is defined by the user. For object n, let $\mathbf{V}_n$ denote the set of vertices on the object represented by a 3D polygonal mesh. The distortion $D^{(t)}_n$ of object n at time t, represented by the Mean Squared Error (MSE), is given by

$$D^{(t)}_n = \frac{1}{\mid \mathbf{V}_n \mid} \sum_{i=1}^{|\mathbf{V}_n|} (d^{(t)}_{ni})^2, \tag{1}$$

where $|\mathbf{V}_n|$ represents the number of the vertices in the set $\mathbf{V}_n$, and $d^{(t)}_{ni}$ represents the distance between the i-th vertex of object n at time t and the corresponding vertex of the object at the target state.

5 vDesign Functions

Based on the proposed hand interactions, a series of functions can be developed in *vDesign* system to satisfy different requirements for designers. In this section, we will describe two typical functions, image segmentation and image composition. When designing a poster or decorating a room, designers prefer to use some composite images, like photomontage [2], to create an unusual visualization experience for audiences. The composite image is generally created by cutting and joining two or more images into an illusion of an unreal subject. Conventionally, this process is realized by image-editing software. However, the desktop based design software have various limitations in interactions and user experience, as mentioned in Section 1. Therefore, we develop the image segmentation and image composition functions in the proposed *vDesign* system such that designers can complete these tasks using hand interactions in the 3D space.

5.1 Image Segmentation and Image Composition

Image segmentation is the process of partitioning an image into different regions which may have similar intensity, color, or texture [22]. In this work, we employ the graph-cut based image segmentation [8]. Compared with other methods, the graph-cut based image segmentation can achieve a globally optimal solution. Also, the segmented result can be efficiently refined in an iterative way by providing additional seeds on the object and background. Let $\mathcal{P}$ denote the set of pixels and $A = (A_1, \cdots, A_p, \cdots, A_{|\mathcal{P}|})$ be a binary vector whose component A_p denote the assignment for pixel p ($p \in \mathcal{P}$). Each pixel can be assigned as interested object or unrelated background. User's indications are taken as hard constraints. For undetermined pixels, the cost function [8] can be formulated as follows.

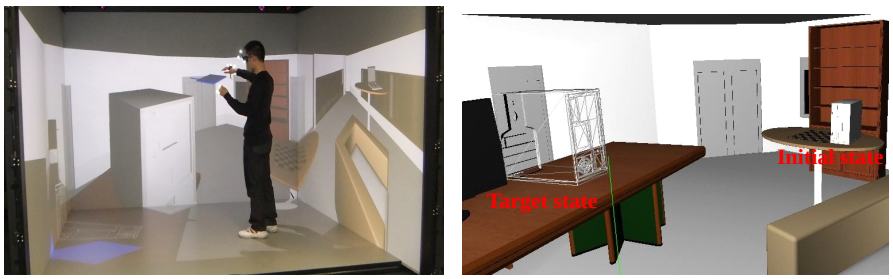
$$E(A) = \lambda \cdot R(A) + B(A), \tag{2}$$

where $R(A)$ is the regional term and $B(A)$ is the boundary term. According to the study in [7], the minimal solution for Equation (2) can be achieved by finding the minimal cut in graph $\mathcal{G}$, whose nodes correspond to pixels in the image. In graph $\mathcal{G}$, the interested object is taken as source node and the background set as sink node. The minimal cut in graph $\mathcal{G}$ is the optimal segmentation with the minimal cost. Graph-cut segments the object from the background by solving a cost minimization problem, in which Equation (2) is the objective function. The interactions specify the object and background seeds, which serve as the constraints for the cost minimization problem.

Image composition is the process of combining segmented objects into a final image, to create an illusion that all those objects are parts of the same scene. In *vDesign*, the user first extracts interested objects from separate images, and then manipulates the objects to the desired states (e,g., the position, the rotation, and the size) on the final image by moving, rotating, and scaling the objects with hand interactions.

5.2 Evaluation Metrics for Image Segmentation and Image Composition

In order to evaluate the performance of image segmentation, we employ the *region-based segmentation accuracy* [13]. The *region-based segmentation accuracy* measures the region



(a) (b)

Fig. 8 *vDesign* user tests: (a) a user performed test in *vDesign*, and (b) initial state and target state of the desktop computer in the object manipulation task.

coincidence between the segmentation result and the ground truth, which is formulated as follows [13].

$$d(S,T) = \frac{|S \bigcap T|}{|S \bigcup T|} = \frac{|S \bigcap T|}{|S| + |T| - |S \bigcap T|}, \tag{3}$$

where $|\cdot|$ is the calculation of the region area, S is the segmented result, and T is the ground truth of interested object. The numerator $|S \bigcap T|$ denotes the coincidence between the segmented result and the ground truth, while the denominator $|S \bigcup T|$ is a normalization factor. The region-based segmentation accuracy $d(S,T)$ is in the range of $[0,1]$. A higher value of $d(S,T)$ indicates a better segmentation.

A composite image consists of multiple objects from different images. Thus, the quality of the composite image depends on the final state of each object. We use *composition distortion* to measure the deviations of objects in the image composition. Let $\mathbf{N}$ denote the set of objects to be manipulated onto the composite image. The composition distortion $D_{com}^{(t)}$ is the sum of the distortions of objects manipulated onto the final image. It is formulated as

$$D_{com}^{(t)} = \sum_{n \in \mathbf{N}} D_n^{(t)}, \tag{4}$$

where $D_n^{(t)}$, given by Equation (1), is the distortion of object n at time t.

6 Experimental Evaluation

6.1 Tasks Description

In order to evaluate the performance of the proposed hand interactions, we conducted a series user tests: menu navigation, object manipulation, image segmentation, and image composition. The user test in *vDesign* is shown in Fig. 8(a). Before starting each task, users were given 10 minutes for training to be familiar with the wand and the hand interactions.

6.1.1 Menu Navigation

In this task, users need to select a desktop computer by navigating a 6-level menu in *vDesign*. The menu is presented in Fig. 5. With the hand interactions, users can activate the main menu with a pull-down action and select one item or trigger the next level menu by touching the

corresponding menu item. With the wand interactions, the menu navigation is performed by pressing buttons. Specifically, there are four buttons on the wand, named Button 1 to Button 4 from left to right. Button 1 is used to activate the main menu, Button 2 used to select menu item or trigger the next level menu, Button 3 used to move up, and Button 4 used to move down. All tests are performed first with the hand interactions, and then with the wand interactions. Times for triggering each level menu were recorded.

6.1.2 Object Manipulation

In the object manipulation task, users were asked to move, rotate, and scale a desktop computer from the initial state to the target state, as shown in Fig. 8(b). The initial state represents the position, the orientation, and the size of the object before any manipulation, while the target state represents the expected state after object manipulations. We compare the performance between the proposed hand interactions and the conventional wand interactions. With the wand interactions, the action of moving or rotating an object is performed by moving the wand, and the scaling of an object is performed by pressing buttons. The time and distortions for manipulating the object were recorded.

6.1.3 Image Segmentation

In the image segmentation task, users need to segment one object from the first image and another object from the second image. The performance of image segmentation was evaluated by the *region-based segmentation accuracy* in Equation (3). The value of region-based segmentation accuracy was calculated in real time and shown to the user when the user was performing image segmentation in the CAVE. With hand interactions, the user moves the right hand to select the interested object while the left hand to select the unrelated background. With wand interactions, the user can draw strokes on the interested object and background by pressing buttons.

6.1.4 Image Composition

In the image composition task, the user needs to move, rotate, and scale the segmented objects and place them onto the specified locations of the background image. *Composition distortion* is used to evaluate the performance of image composition. The tested images and ground truth images are selected from the GrabCut image data set of Microsoft Research Cambridge [23]. With the wand interactions, the user can move or rotate the segmented object by moving the wand, and scale the object by pressing buttons.

6.2 Experiment Setup

We implemented the prototype of *vDesign* in the CAVE at Ryerson University, Canada. The prototype is developed on Windows operating system in C++ language based on the libraries of VR Juggler [3] and OpenSceneGraph [1]. We also conducted the above mentioned user tasks to evaluate the performance of the proposed *vDesign* system.

We invited fifteen students (2 females and 13 males) in Ryerson University to participate in the user tests. The average age was 28-year old. One was left-handed, while the others were right-handed. Eight participants were in the expert group, who had been using the wand

and the hand interactions for more than 6 months. The other participants were in the novice group, who had used wand occasionally and had not used any hand interactions before the test. The novice group was trained on the usages of the wand and hand interactions for 10 minutes before the test. Both participants performed the same test 2 times with hands and 2 times with wand.

6.3 Experimental Results and Discussions

Table 1 Completion times for menu navigation task in seconds

Interaction	Subject expertise	Mean	Standard deviation
Hand	Expert	8.00	1.08
	Novice	10.61	0.57
	Total	9.22	1.59
Wand	Expert	10.33	0.93
	Novice	14.59	2.18
	Total	12.31	2.70

Table 2 ANOVA results for menu navigation task completion time in seconds

Source of variation	Sum of squares	Degrees of freedom	Mean square	F	P-value
Between subjects	71.759	1	71.759	14.570	6.850×10^{-4}
Within subjects	137.903	28	4.925		
Total	209.662	29			

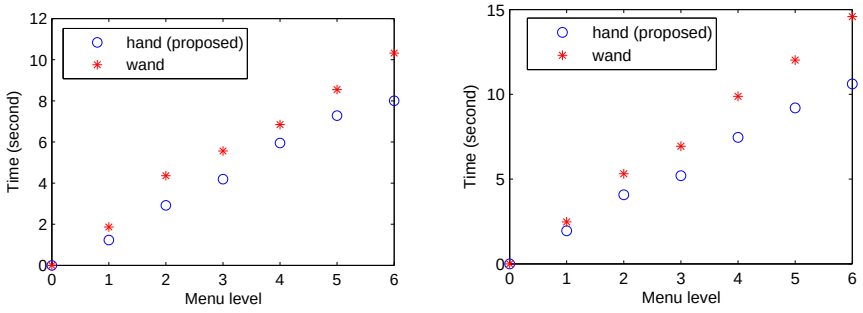


Fig. 9 Comparison of the average trigger time for menu navigation: (a) by the expert group, and (b) by the novice group.

We first compare the time spent on the menu navigation task between the wand interactions and the hand interactions. Table 1 shows the mean and standard deviation of the time

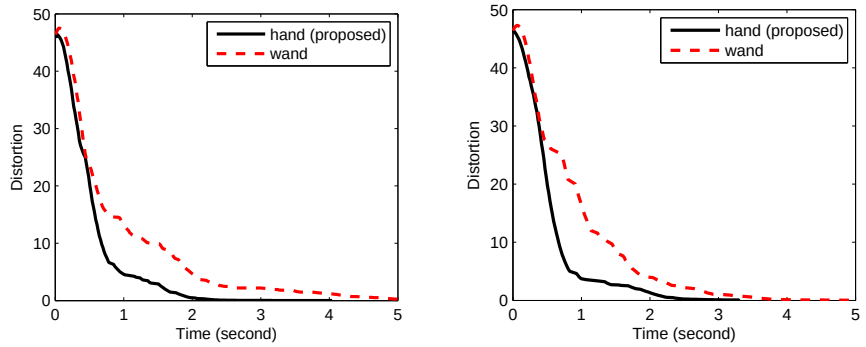


Fig. 10 Comparison of the time-distortion relationship for the manipulations of the desktop computer: (a) by the expert group, and (b) by the novice group.

to complete the menu navigation task. The expert group completes the 6-level menu navigation using the proposed hand interactions within 8 seconds, reducing the time by 22.5% compared to the wand interactions. The average time for selecting a menu item is 1.33 seconds with the hand by the expert, which is lower than that (1.72 seconds) with the wand. In the novice group, the proposed hand interactions reduce the time by 27.2% compared to the wand interactions. We use ANOVA to analyze the significance of difference. In our case, the interaction type is the between-subject factor, and it includes two levels: the proposed hand interactions and the traditional wand interactions. We analyze the task completion time from 15 participants using ANOVA. The ANOVA result for menu navigation task completion time is shown in Table 2. From Table 2, we can see that the difference of completion time between hand and wand interactions is statistically significant ($F(1, 28) = 14.57$, $p = 6.850 \times 10^{-4} < 0.05$). Fig. 9(a) and Fig. 9(b) show the average trigger time of each of the 6 menu levels for the operation by the expert group and the novice group, respectively.

We next compare the time-distortion relationship for the object manipulation task between the wand interactions and the hand interactions. Specifically, we use distortion in Equation (1) to represent the deviation from the target state. In Fig. 10, we plot the distortion at different time when the desktop computer is manipulated from its initial state towards its target state. Fig. 10(a) gives the average time-distortion relationship performed by the expert group, while Fig. 10(b) by the novice group. We can see that the hand interactions can get a lower distortion at any time compared to the wand interactions by either the expert group or the novice group. We define the *satisfactory state* as the state of the object with a distortion less than 0.1. As shown in Fig. 10(a), the desktop computer manipulated by the expert group with hands can reach the satisfactory state within 2.3 seconds, which is lower by 52.1% compared to that with wand. As shown in Fig. 10(b), the novice users can manipulate the desktop computer with hands to the satisfactory state within 2.5 seconds, reducing the time by 36.3% compared to the wand. In addition, the time for manipulating object with hand interactions by the novice group is close to that by the expert group, indicating that the proposed hand interaction technique is easy to learn.

We also compare the mean and the standard deviation of the time for the object manipulation task between the wand interactions and hand interactions. As shown in Table 3, the proposed hand interactions reduce the completion time by 24.3% for the expert group, and 19.6% for the novice group, compared to the wand interactions. The standard deviation in

Table 3 Completion times for object manipulation task in seconds

Interaction	Subject expertise	Mean	Standard deviation
Hand	Expert	4.04	0.13
	Novice	4.34	0.54
	Total	4.18	0.48
Wand	Expert	5.34	0.13
	Novice	5.77	0.32
	Total	5.54	0.32

Table 4 ANOVA results for object manipulation task completion time in seconds

Source of variation	Sum of squares	Degrees of freedom	Mean square	F	P-value
Between subjects	11.154	1	11.154	67.698	5.908×10^{-9}
Within subjects	4.613	28	0.165		
Total	15.767	29			

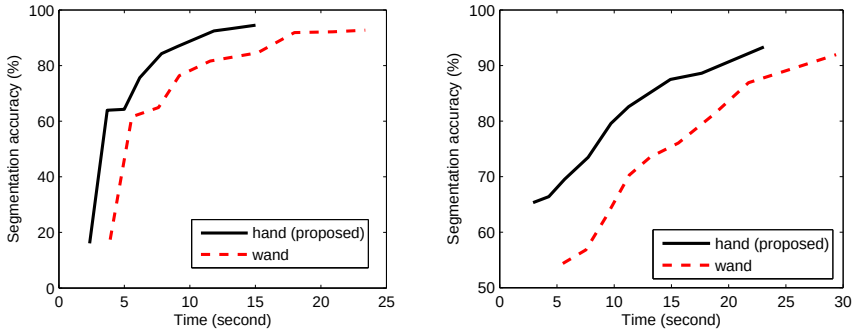


Fig. 11 Comparison of the performance on the image segmentation task: (a) by the expert group, and (b) by the novice group.

each case is low, indicating that the total time tends to be very close to the mean. The time taken by the novice group with the hand interactions is close to the expert, indicating that the novice users can perform the hand interactions quite well after a short time of training. We analyze the object manipulation task completion time with ANOVA, which is shown in Table 4. From Table 4, we can get that there is a significant effect of hand and wand interactions on completion time, since $F(1, 28) = 67.698$ and $p = 5.908 \times 10^{-9} < 0.05$.

We next compare the performance on the image segmentation task between the hand interactions and the wand interactions. We use the region-based segmentation accuracy in Equation (3) to measure the accuracy of the segmentation result. In our experiments, the same images are segmented by the same user with the same segmentation algorithm under the same CAVE environment. The accuracy of the segmentation results therefore depends on the user's interactions during the segmentation, i.e. how accurate and efficient the user can interactively select the seeds of foreground and background objects. Fig. 11(a) shows the average time and segmentation accuracy by the expert group, while Fig. 11(b) by the novice group. Observing Fig. 11(a) and Fig. 11(b), we can find that the proposed hand interactions perform much better than the wand interactions in terms of convergence time by

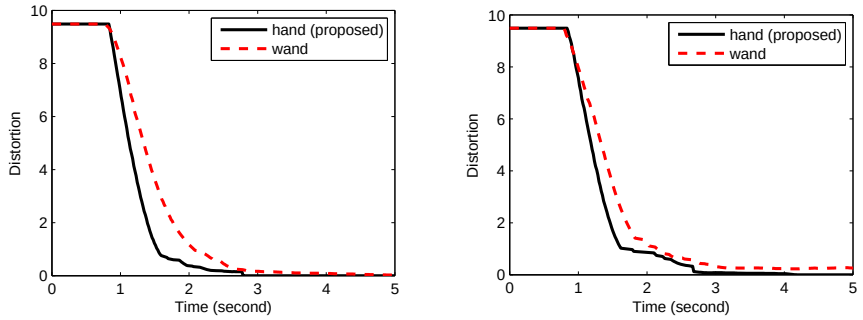


Fig. 12 Comparison of the time-distortion relationship on the image composition task: (a) by the expert group, and (b) by the novice group.

either the expert group or the novice group. Compared to the wand, the hand interactions enable quicker and more accurate operations in drawing the strokes on the object and the background. The object and the background can be selected simultaneously with hand interactions, while they can only be selected one by one with wand interactions. As shown in Fig. 11(a), the images segmented by the expert group with hand can reach 90% segmentation accuracy within 10.8 seconds, which is lower by 35.8% compared to that of with wand interactions.

We finally compare the time-distortion relationship on the image composition task between the hand and the wand interactions. The distortion in Equation (4) is used to represent the composition distortion. Fig. 12(a) and Fig. 12(b) show the average time-distortion relationship performed by the expert group and the novice group, respectively. We can see that the proposed hand interactions can provide faster interactions in object manipulations compared to the conventional wand interactions. As shown in Fig. 12(a), the expert users with hand interactions can reach the satisfactory state (distortion under 0.1) within 2.8 seconds, reducing the time by 25% compared to the wand interactions.

The composite image is illustrated in Fig. 13. The first object is the sheep, which is extracted from image 1 shown in Fig. 13(a). The second object is the lady, which is extracted from image 2 shown in Fig. 13(b). The two objects are manipulated and then placed onto another background image. The composite image is used as a picture frame hung on the wall of the virtual room, as shown in Fig. 13(c).

7 Conclusion

In this paper, we proposed *vDesign*, a CAVE-based virtual design environment using hand interactions. Hand interactions are triggered based on the real-time positions of the markers worn on the hands of the user. The hand interactions in *vDesign* are classified into menu navigation and object manipulations. For menu navigation, we define two interactions: activating the main menu and selecting a menu item. For object manipulations, we define three interactions: moving, rotating, and scaling an object. With the hand interactions, we develop image segmentation and image composition functions in *vDesign*. In the image segmentation, the user can use the right hand to select the interested object and the left hand to select the unrelated background. Based on the user's selection, a graph-cut based image segmenta-

(a) (b)

(c)

Fig. 13 Illustration of image segmentation and composition: (a) image 1 with a sheep, (b) image 2 with a lady and a boy, and (c) the composite image with the sheep extracted from image 1 and the lady extracted from image 2.

tion is performed to extract the interested objects from images. In the image composition, the user can manipulate the segmented objects with hands and then combine them together into a final image. We implemented the *vDesign* prototype, via which we conducted experiments to compare the hand interactions and the traditional wand interactions. The experimental results demonstrated that the proposed hand interactions can provide faster and more accurate interactions compared to the traditional wand interactions.

Acknowledgements This work was supported in part by the Canada Research Chair Program, NSERC Discovery Grant, and NSFC Grant 61210005.

References

1. Openscenegraph. http://www.openscenegraph.org/projects/osg
2. Photomontage. http://en.wikipedia.org/wiki/Photomontage
3. Vr juggler. http://vrjuggler.org/
4. Abramyan, L., Powell, M., Norris, J.: Stage: Controlling space robots from a cave on earth. In: Aerospace Conference, 2012 IEEE, pp. 1–6. IEEE (2012)
5. Biocca, F., Delaney, B.: Immersive virtual reality technology. Communication in the age of virtual reality pp. 57–124 (1995)

6. Bowman, D.A., Wingrave, C.A.: Design and evaluation of menu systems for immersive virtual environments. In: Virtual Reality, 2001. Proceedings. IEEE, pp. 149–156. IEEE (2001)
7. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient nd image segmentation. International Journal of Computer Vision **70**, 109–131 (2006)
8. Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In: Proc. of IEEE International Conference on Computer Vision, vol. 1, pp. 105–112 (2001)
9. Burdea, G., Coiffet, P.: Virtual reality technology. Presence: Teleoperators and virtual environments **12**(6), 663–664 (2003)
10. Creagh, H.: Cave automatic virtual environment. In: Proc. of IEEE Electrical Insulation Conference and Electrical Manufacturing, pp. 499–504 (2003)
11. Cruz-Neira, C., Sandin, D.J., DeFanti, T.A.: Surround-screen projection-based virtual reality: the design and implementation of the cave. In: Proceedings of the 20th annual conference on Computer graphics and interactive techniques, pp. 135–142. ACM (1993)
12. Cruz-Neira, C., Sandin, D.J., DeFanti, T.A., Kenyon, R.V., Hart, J.C.: The cave: audio visual experience automatic virtual environment. Communications of the ACM **35**(6), 64–72 (1992)
13. Ge, F., Wang, S., Liu, T.: New benchmark for image segmentation evaluation. Journal of Electronic Imaging **16**(3) (2007)
14. Kenyon, R.V., Sandin, D., Smith, R.C., Pawlicki, R., Defanti, T.: Size-constancy in the cave. Presence: Teleoperators and Virtual Environments **16**(2), 172–187 (2007)
15. Kim, J.S., Gračanin, D., Matković, K., Quek, F.: The effects of finger-walking in place (fwip) for spatial knowledge acquisition in virtual environments. In: Springer Smart Graphics, pp. 56–67 (2010)
16. Koike, M., Makino, M.: Crayon a 3d solid modeling system on the cave. In: Proc. of IEEE International Conference on Image and Graphics, pp. 634–639 (2009)
17. Li, Y., Shark, L.K., Hobbs, S.J., Ingham, J.: Real-time immersive table tennis game for two players with motion tracking. In: Information Visualisation (IV), 2010 14th International Conference, pp. 500–505. IEEE (2010)
18. Lugrin, J.L., Charles, F., Cavazza, M., Le Renard, M., Freeman, J., Lessiter, J.: Caveudk: a vr game engine middleware. In: Proceedings of the 18th ACM symposium on Virtual reality software and technology, pp. 137–144. ACM (2012)
19. Ohno, N., Kageyama, A.: Introduction to virtual reality visualization by the cave system. Advanced Methods for Space Simulations, edited by H. Usui and Y. Omura, TERRAPUB, Tokyo pp. 167–207 (2007)
20. Pausch, R., Proffitt, D., Williams, G.: Quantifying immersion in virtual reality. In: Proceedings of the 24th ACM annual conference on Computer graphics and interactive techniques, pp. 13–18 (1997)
21. Prachyabrued, M., Ducrest, D., Borst, C.: Handymap: a selection interface for cluttered vr environments using a tracked hand-held touch device. Advances in Visual Computing pp. 45–54 (2011)
22. Raut, S., Raghuvanshi, M., Dharaskar, R., Raut, A.: Image segmentation – a state-of-art survey for prediction. In: Proc. of IEEE International Conference on Advanced Computer Control, pp. 420–424 (2009)
23. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics (TOG) **23**(3), 309–314 (2004)
24. Ruddle, R.A., Savage, J.C., Jones, D.M.: Symmetric and asymmetric action integration during cooperative object manipulation in virtual environments. ACM Transactions on Computer-Human Interaction (TOCHI) **9**(4), 285–308 (2002)
25. Silva, R., Giraldi, G., Oliveira, J.C.: Introduction to virtual reality. Tech. rep., Technical Report: 06/2003, LNCC, Brazil (2003)
26. Song, P., Goh, W.B., Fu, C.W., Meng, Q., Heng, P.A.: Wysiwyf: exploring and annotating volume data with a tangible handheld device. In: Proc. of SIGCHI Conference on Human Factors in Computing Systems, pp. 1333–1342 (2011)
27. Sutcliffe, A., Gault, B., Fernando, T., Tan, K.: Investigating interaction in cave virtual environments. ACM Transactions on Computer-Human Interaction (TOCHI) **13**(2), 235–267 (2006)
28. Wijayasekara, D., Linda, O., Manic, M.: Cave-som: Immersive visual data mining using 3d self-organizing maps. In: Neural Networks (IJCNN), The 2011 International Joint Conference on, pp. 2471–2478. IEEE (2011)
29. Wu, A., Reilly, D., Tang, A., Mazalek, A.: Tangible navigation and object manipulation in virtual environments. In: Proceedings of the fifth international conference on Tangible, embedded, and embodied interaction, pp. 37–44. ACM (2011)
30. Wu, Q., Boulanger, P., Kazakevich, M., Taylor, R.: A real-time performance system for virtual theater. In: Proceedings of the 2010 ACM workshop on Surreal media and virtual cloning, pp. 3–8. ACM (2010)
31. Yang, X., Deines, E., Lauer, C., Aurich, J.C.: A human-centered virtual factory. In: Management Science and Industrial Engineering (MSIE), 2011 International Conference on, pp. 1138–1142. IEEE (2011)