



Interaction in Virtual Environments - How to Control the Environment by Using VR-Glasses in the Most Immersive Way

Barbara Streppel^{1(✉)}, Dorothea Pantförder², and Birgit Vogel-Heuser²

¹ MaibornWolff GmbH, Munich, Germany
barbara.streppel@maibornwolff.de

² Institute of Automation and Information Systems,
Technical University of Munich, Munich, Germany

Abstract. Not only in the gaming industry is Virtual Reality (VR) the new way to give users a new experience – in engineering or production plant operation we also see first attempts at finding innovative ways of visualizing data or training plant staff. This is necessary because processes are getting more and more complex thanks to higher interconnection and flexibility. This paper presents actual possibilities of interacting with a virtual environment and provides three concepts for immersive interaction. We also show the results of an evaluation of these concepts at the end of the paper.

Keywords: Virtual reality · Interaction · Virtual environment

1 Introduction

The world is getting more and more complex and needs new ways to display and further understand this complexity. First approaches for using Virtual Reality in this context exists. The advantage of representing complex problems in VR is the increased sense of presence [1]. Presence describes the feeling of being physically present in a non-physical virtual environment (VE) and is separated into immersion and involvement.

Immersion is a crucial factor for presence and can be influenced by interaction with the VE. Previous works indicate that operators in a monitoring task spotted more problems if they were able to interact with a 3D visualization [2].

Involvement is mainly influenced by the environment of a system – e.g. desktop PC, VRWall/Cave or VR-Glasses [3]. Hence, using VR to visualize data, processes or software structure supports the plant staff and software developer in analyzing and understanding of such complex problems. Especially the ability to look around freely and without using a mouse increase the perception of the environment [4].

An important part of every human-operated application is the human-machine-interface. How to design and visualize a 2D-interface has been extensively explored and there are existing guidelines for this, e.g. in ISO 9241. However, every developer/UX-designer must decide on their own how to design and visualize data and interfaces in real 3D/VR.

The first part of this paper summarizes the possibilities of how to interact with a virtual environment (Sect. 2). Following this we verify if we can transfer requirements of a 2D-application into VR for an intuitive and natural interaction (Sect. 3). Based on these requirements we divide different interaction methods into explicit interaction concepts (Sect. 4). In Sect. 5, we explain the basics of our software prototype, followed by Sect. 6 where the structure, provision and analysis of the evaluation is reflected. A summary, conclusions and an outlook on further research potentials is given in Sect. 7.

2 Related Work

There is much research on Human-Machine-Interfaces or Human-Computer-Interactions for 2D applications, which we present in this section. Existing works on interaction concepts in virtual reality are also presented here. At the end of this chapter, we evaluate existing controls and try to divide them into subcategories.

2.1 Human-Machine-Interface

Every task a machine should execute in response to human input needs an interface for transmitting this information. This can be a simple, physical button or a software user interface. Human Machine Interfaces (HMI) include all points of contact between machine and human. This contains everything that a machine displays to the user and – vice versa – all user inputs for the machine.

The research field of ergonomics follows up with a good handling for HMIs. For this work we take a closer look at the field of software ergonomics. The most important rules and recommendations for a good usage of an 2D user interface are described in ISO 9241. The norm defines a guideline for human computer interactions, e.g. amongst others guidance on dialogue principles or guidance on usability [5].

To rate the usability of a software you can use the questionnaire laid out in ISO 9241/110, which uses the seven ergonomic principles, which are: suitability for the task, suitability for learning, suitability for individualisation, conformity with user expectations, self-descriptiveness, controllability and error tolerance. Nielson (1994) recognized similar factors for the quality of a human machine interface: visibility of system status, match between system and real world, user control and freedom, consistency and standards, error prevention, recognition rather than recall, flexibility and efficiency of use, aesthetic and minimalist design, help users recognize, diagnose and recover from errors, help and documentation [6].

In this paper we try to figure out which of these 2D-principles can be transferred to a 3D world and can be used to create an intuitive and natural user interaction concept for VR (Sect. 3.2).

Even if a software or UX developer takes all these principles into account, he needs to perform user tests to get the final interface design. This is because every application needs a special interaction concept, which depending on its working task. To create a working interaction concept we performed some user tests to receive feedback on our concept ideas before implementing the final prototype for the final evaluation.

2.2 Virtual Reality – Interaction

Interactions with a virtual environment are not the same as with a real environment. Scenarios which partially allows more than in reality and the missing haptic feedback in a virtual world require new concepts of interaction.

In some cases, voice control is used to control a VE. More common is the usage of controller or gesture control. Generally, there are at least two steps for the user to interact with objects in a VE [7]:

1. *Selection*
 - a. Directly grabbing the object (e.g. by pressing a button) or
 - b. Selecting the specified object (e.g. by ray cast) and then grabbing it (e.g. by pressing a button)
2. *Manipulation*
 - a. Changing the position of the object
 - b. Changing the orientation of the object

Mine defined two more forms of interaction: *movement* and *scaling*, and from these derived a fifth form, *virtual menu and widget* [8].

Several papers provide various classifications for moving. Ware gives a general classification of moving in a 3D world [9]. He divides moving into *walking* and *flying*, whereby moving has two degrees of freedom and flying has three. Mine introduced several other principles, one is called *dynamic scaling*. Here, the user can minimize the world, so he has to move a minimal way to reach his goal [8].

Sasse et al. use a division into two concepts called *world-in-hand* and *eyeball-in-hand*. With *world-in-hand* the user is fixed in position and can move the world around him. With *eyeball-in-hand* the world is fixed, and the user moves around the world [10]. This division can also be used for *scaling* and *manipulation*. In an application the movement method should be consistent for the user, i.e. either he moves or the world does.

Furthermore, Mine introduced three categories of how to implement the existing forms of interaction:

1. *Direct User Interaction*: Hand tracking and gesture recognition to map real user action to a resulting action in the virtual world.
2. *Physical Controls*: Real physical objects to interact with the virtual world, like a steering wheel in a driving simulation.
3. *Virtual Controls*: Interact with the environment by using Virtual Control elements.

Direct User Interaction could also be named gesture control. There is no agreed definition of the word gesture. Kurtenbach und Hulteen described a gesture as the motion of a part of the human body that contains information. For them pushing a button isn't a gesture, because it doesn't contain information [11]. For Gartner the most important part of gesture control is that the user doesn't touch the physical world [12]. Furthermore, Harling and Edwards consider only gestures of hands without the interaction with a physical interface. Therefore, they differentiate between hand poses and positions. Both can be static or dynamic. This leads to the following classification in Table 1 [13]:

Table 1. Classification of gesture of Harling and Edwards

Hand pose	Hand position	Example
Static	Static	Peace-Sign
Static	Dynamic	Waving goodbye
Dynamic	Static	Grabbing
Dynamic	Dynamic	Throwing

2.3 Existing Controls

There are several VR applications, most of them games. To get an overview of existing controls we investigated other VR applications. Selecting and manipulating objects are most of the times implemented in the same way. Only moving is different in many applications.

Apart from these classifications, most applications use a teleport system for moving the user, thereby avoiding motion sickness. For this the user can use one button to create a laser pointer and release it, or use a second button to teleport to the appearance point of the laser pointer. Sometimes the user can define his direction in which he wants to look after teleporting [14]. Another possibility is pausing the application at the end of the room tracking and let the user rearrange his avatar. This can be done by turning the head of the user, while the avatar remains stationary. In the game *Runes* the user can switch between first and third person view [15]. To avoid motion sickness, the user stands at a fixed point when he is using third person and moves his avatar with a remote control.

For grabbing things, the principles of Sect. 2.2 are used. Most of the time the user can press a button and hold it down or click the button to pick up a subject and click again to release it. Google Earth uses the first form for letting the user move over the world. This is like grabbing and pulling a fixed object.

Interacting with UI elements, like in a menu, the investigated applications use two different concepts. The user can use a laser pointer to select a UI element or he touches the element directly with the controller, which commonly gives haptic feedback. To use the selected element, he has to push a controller button.

Some controls are made for gesture control, so they can be used with a camera for hand or gesture recognition. There the user for example can grab subjects by using a fist or a pinch gesture.

In some cases moving with gestures is the same as with a controller, where the user grabs the world and pulls himself around. Another possibility is to have points of interest floating around the user, which can be selected. Furthermore, a neutral hand position can be defined and the distance between the hand and the camera defines the velocity of moving. Hence the user can move forward and backward [16, 17].

3 Requirements Analysis

The goal of this paper is to evaluate which are good interaction concepts for a VE. A good concept should be intuitive, natural and easy to use. To achieve this we set up

requirements for interaction in VR. To be able to evaluate the different concepts there should be a VR scenario in which the evaluation participants can test them. This scenario was chosen to be a so-called “software city”¹. After explaining the initial situation we will describe the derived requirements.

3.1 Application Example

We decided to use a software city as an application example for the different concepts. An example of a software city can be seen in Fig. 1. A software city is a representation of software code as a city. The streets of the city constitute the folders of the software code, and buildings constitute the classes. Height, base area and building color can represent different metrics, such as lines of code, complexity or test coverage. The user can therefore directly see where a large class is or where there are not enough tests [18].

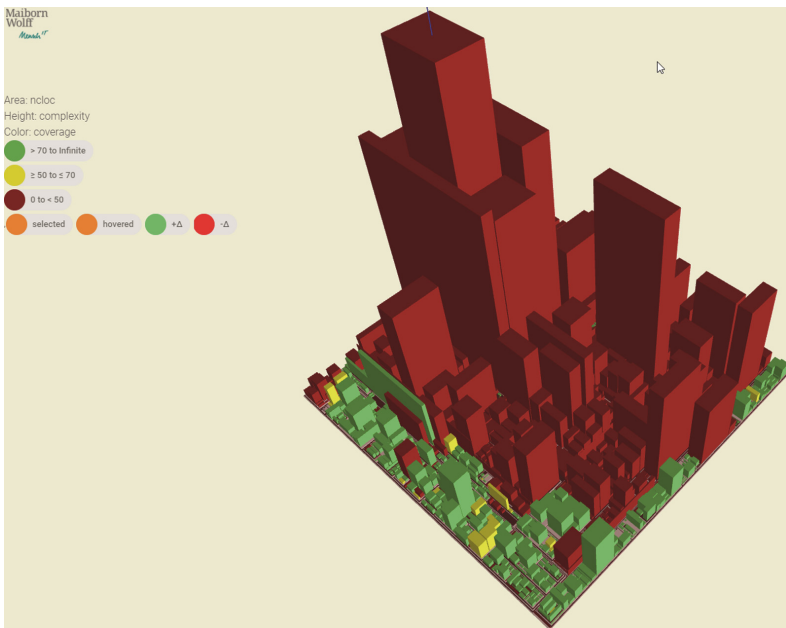


Fig. 1. SoftwareCity desktop version

A desktop application enables the transformation of code into a software city. The user can move, rotate and scale the city. Furthermore, he is able to select a building and get more information about the related class.

The software city is used to analyze software code. Therefore, a software developer can see the whole of his software and the relations between different classes as well as explaining the complexity or quality of the software e.g. to a manager who has no deeper understanding of software development.

¹ Now CodeCharta.

3.2 Requirements

As the VR version should have the same interaction possibilities as the desktop version, the following should be implemented:

- Movement between city and user
- Rotation between city and user
- Scaling between city and user
- Selecting a building to get more information
- Interacting with the UI elements which contain the information

The interactions should be immersive, so the VR-system needs hand tracking or a handheld controller. Because moving is a requirement for the application, motion sickness should be minimized. The VR-system should be portable so it can be taken to customers.

To get the advantages of VR, such as an increased perception of the environment, the application must have a high immersion. This is achieved through a good display and the application's interaction fidelity [19]. Display fidelity is affected by the VR-system used and its accompanying computer system. To have a high level of interaction fidelity, the interactions should be natural and intuitive, so that they are as near as possible to real or other known interactions.

Furthermore, all interactions should follow the rules of dialogue principles, as defined in ISO 9241-110. These rules are for desktop applications, but can be adapted for VR interaction controls:

- Suitability for the task: the control supports the user by executing his task
- Suitability for learning: the control offers instructions
- Suitability for individualization: the control can be customized
- Conformity with user expectations: the control is consistent and follows known conventions
- Self-descriptiveness: the user knows at every point what he is doing and can do
- Controllability: the user can go back and can undo actions
- Error tolerance: the control is robust against invalid user input and tries to avoid it.

Each of these rules can be adapted for VR interaction controls. It's unclear how necessary different rules have to be considered for the interactions to be implemented. E.g. for the rule of controllability the user can undo unwished interaction, e.g. a too far scaling, by scaling it again in the opposite direction and doesn't need a back button.

4 Concept

After defining the requirements for the application, we decided to define three different interaction concepts. Thereby, the classification of the concepts is inspired by the classification of Mine (Sect. 2.2). Hence, there are a gesture control (direct user interaction), a control by physical controller (physical controls) and a control by virtual UI elements

(virtual controls). It should be noted that we considered voice control but didn't implement it as scaling or moving a city by voice felt unnatural. For following researchs it would be interesting to implement parts of a VR application as voice control.

Because of the portability requirement the concepts should be implemented for a head mounted display (HMD). At the beginning of the work only the HTC-Vive had tracked controllers, so we used it as HMD [20]. To enable gesture control we used Leap Motion [21].

Below the procedure of creating the different concepts is presented and the usage of specific interactions is explained. For creating a natural and intuitive interaction concept we perform a preliminary experiment with some participants.

4.1 Preliminary Experiment Defining Suitable Interaction Elements

To have the controls as a product of human imagination and therefore natural and intuitive for most people, we created a previous version of the prototype. With it we conducted interviews with six participants. The participants had different knowledge about VR, the software city and controller usage. Features implemented in the pre-prototype were scaling, rotating and moving the city with controller and UI elements.

Every participant was interviewed alone and by using a guideline with pre-defined questions. By using these questions, the participants were lead through the interview. They tried out the pre-prototype and with this we discussed the points of discussion shown in Sect. 4.2.

4.2 Discussion of Different Interaction Elements

As described in Sect. 3, the user should be able to move around in the city. The user could do this by moving himself, or he could move the city around him as in the principles of Sect. 2.2 world-in-hand or eyeballs-in-hand. By using the HTC-Vive with its room scale tracking the user can also walk by himself in the provided space.

The same principles are important for rotating. Because of the tracking of the user, he could just turn his head to see the city from another perspective. So, perhaps he doesn't need the possibility to rotate it. Generally it's important for manipulation to know the center of rotation and scaling, i.e. if the city rotates/scales around the user or around its own center.

There were three topics of discussion before starting the implementation of the real prototype. Each topic has various implementation possibilities:

1. Moving:
 - a. The user moves inside the city
 - (1) In real world by room scale tracking
 - (2) By using moving elements
 - b. The user moves the city around him
2. Rotation:
 - a. The user doesn't need rotational elements, because he can turn his head
 - b. The user can rotate the city

- (1) Around himself
 - (2) Around the center of the city
3. Scaling:
- a. Scaling is around the user himself
 - b. Scaling is around the center of the city.

To determine which possibilities to implement, we interviewed the six participants, as described in Sect. 4.1. At this point there were a few interactions implemented, so they could try out some possibilities. Users consistently noted that the movement of the city relative to the user was crucial. There are two different main positions for the user: (A) The user is outside the city and the city is small right before him and (B) he stands inside a large scaled city. Table 2 shows the decisions of the participants which interactions should be possible for which position.

Table 2. Possible interactions depending on the users position

	Position A	Position B
Moving	1a & 1b	1a & 1b
Rotating	2b(2)	2b(1)
Scaling	3a	3a

Generally, the suggestions of the participants point to a manipulation of the city not of the user. There should be a way to move/scale/rotate the city, because the user can move and rotate himself without controls just by real moving or turning his head. However, moving the city and himself in some way should be possible, because there will be not enough space to walk around in a large city inside a small testing room.

After trying the first parts of the prototype they also mentioned that the cables of the HTC-Vive were disturbing when turning their head, and it would be better to always have the ability to rotate the city. If the user has the possibility to move the city it would be enough to rotate the city around the user, so he doesn't lose track of his position inside it.

However, for rotating there should be two possible rotation centers. If they are in position B, the city should rotate around the user. In position A, it should rotate around the city itself, so the city can be viewed from all sides.

During the interviews we asked the participants also which gestures they would prefer to interact with the software city for the required interactions.

4.3 Direct User Interaction

In this concept all interactions are made with gestures. In this case gestures are a combination of the pose of the hand and its movement. Both will be recognized by a leap motion camera [21]. The leap motion can recognize the position of a hand and its fingers, as well as the direction the palm and the fingertips are pointing in. It follows that the gestures have to be unique and easily recognizable for a leap motion camera.

Gestures should be as natural as possible so the user can remember them. Don Norman, Director of the Design Lab of The University of California and usability

specialist, considers just a handful of gestures to be natural for him. The rest have to be learned and remembered later. This is only possible for far fewer gestures than would be required [22].

Moving. Most of the interviewed participants wanted to move the city and not themselves. Only one suggested a gesture control for moving himself inside a fixed city. We conclude that in this concept it's enough to only implement a way of moving the city around the user. The most commonly suggested way to do this was to grab the city and drag it around. For grabbing, the hand has to be clenched to a fist.

Rotating and Scaling. For rotating and scaling we pursued two strategies: either using one hand or two. To rotate or scale the city around its own center, the user can grab it with both hands, move them relative to each other for rotating, and pull them apart for scaling. This is inspired by the way a map is manipulated on a touch device, e.g. a smartphone, where the user moves two fingers for rotating und scaling a map.

To rotate the city around the user himself, he has to open one hand and point with his palm in the direction he wants to rotate the city (Fig. 2 left). For scaling he has to point his palm into in the direction of the camera. If he points away from the camera and moves his hands, the city is minimized. The other way around maximizes the city (Fig. 2 right).

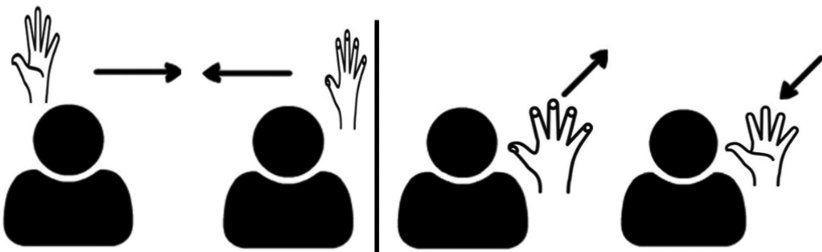


Fig. 2. One-handed gestures to rotate (left) and scale (right) the city around the user

Selecting a Building. To select a building the participants suggested simply touching the chosen building with an outstretched index finger.

Operating the UI. The control of the UI is built up as on a touch device, e.g. a smartphone. For this, the user can use his whole hand in every hand holding position.

4.4 Physical Control

In this concept the user interacts with the virtual environment only by using the controllers of the HTC-Vive. To make it as “physical” as possible, the interactions should be executed, as far as it seems reasonable, only by the buttons of the controller, not by describing gestures with it.

To better understand the different buttons used, Fig. 3 describes the key assignment of the used HTC-Vive Controller. The trigger (Fig. 3 number 7) is used for turning on

a laser pointer. For this the user has to slightly touch it. By pushing it up to the end, this means clicking it, the user can select or confirm something. The grab buttons (Fig. 3 number 8) are for grabbing things and the trackpad (Fig. 3 number 2) is used for scaling and rotating the city.

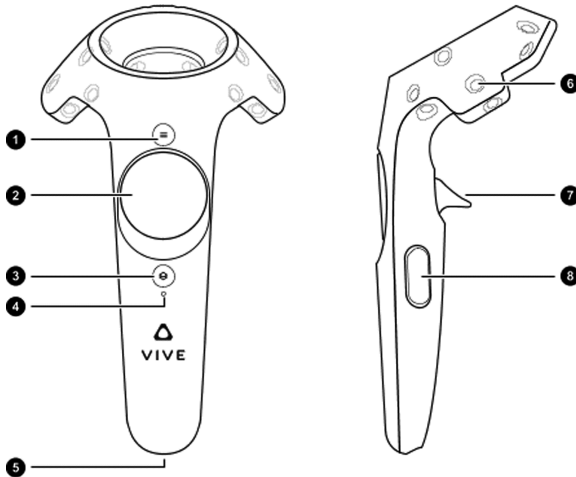


Fig. 3. Key assignment of the HTC-Vive Controller

Moving. Even in this concept, the participants decided that the possibility to move the city, not themselves, is enough. Here, they can use the same method as in gesture control, so they grab the city with the controller and pull it around. Grabbing is executed by pressing the grab buttons of the controller.

Furthermore, the user has the possibility to use a laser pointer to teleport himself through the city. The laser pointer is turned on by touching on the trigger. By pushing the trigger up to the end, this means clicking the trigger, the user conforms that he wants to teleport to the endpoint of the laser pointer.

Rotating and Scaling. For rotating and scaling the trackpad is used. Pushing the upper or lower half of it maximizes or minimizes the city respectively. When the left or right half is pushed, the city rotates. In this context the size of the city is important. The physical control concept constantly recognizes how large the city is. If the city is big, it rotates around the user, and if the city is small, it rotates around its own center. Interview participants said it's enough to scale the city around the user himself if they have the possibility to move the city.

Selecting a Building. To have a self-contained concept, the user here also uses the laser pointer. By touching the trigger, the laser pointer appears. If the user wants to select a building, he can touch it with the beam (Fig. 4) and open the additional information window by clicking the trigger.

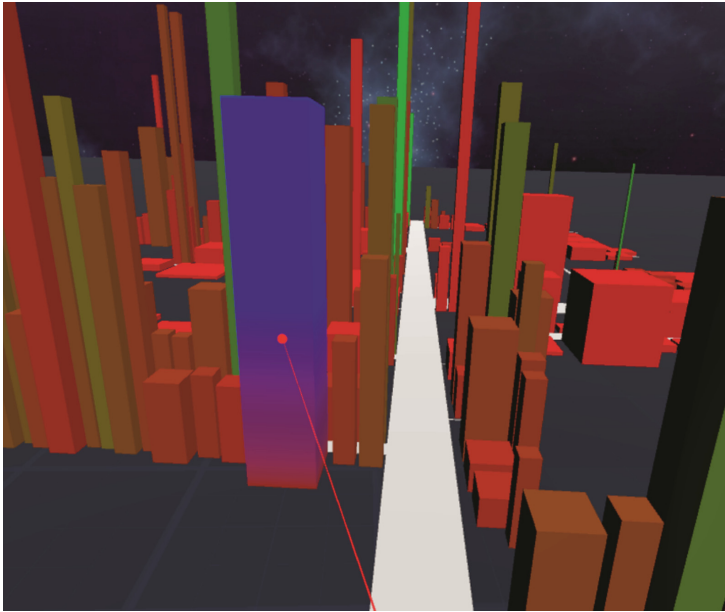


Fig. 4. Selecting a building with the laser pointer

Operating the UI. Also for operating the UI the laser pointer is used. It can be used like a mouse on a desktop PC.

4.5 Virtual Control

In virtual control the user should use virtual elements to manipulate the city. To operate the virtual elements the user should use directly his own hands. It also should be possible to move the elements, so that they do not disturb the user's field of view. The position of the elements can be manipulated by using the trackpad of one of the HTC-Vive controllers. To get moving elements into the foreground and back the user swipes the trackpad from side to side. If he wants to use the scale/rotate elements he swipes from bottom to top or other way around, depending on whether the elements should be in his upper or lower field of view.

Moving. For moving there is a mini map of the city, where the user can see himself and the city from above (Fig. 5 left). To move the city, he can grab inside the map, by clenching his hand to a fist, and pull it in the direction he wants the city to move. This pulling is executed in the coordinate axes of the map, not of the city.

Rotating and Scaling. On the right side of Fig. 5 the elements for rotating and scaling are presented. It was important that these elements are positioned and the user can use them, as in normal touch interfaces. Standard UI elements, e.g. buttons, sliders and knobs, are used for this. This special VR-UI-elements are three-dimensional and can be

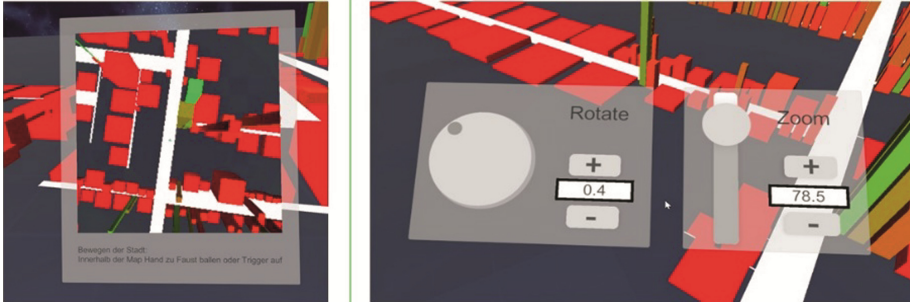


Fig. 5. Moving, rotating and zooming elements (from left to right)

used by pushing them down and then manipulating them. Instead of the missing haptic feedback, there is visual feedback, so the elements change color as soon as they're pushed down. For fast rotating or scaling the user can handle the knob or slider. For fine adjustment he can use the plus and minus buttons.

Selecting a building and operating the UI in this concept works like in the gesture control. Selecting a building can be done by touching it with an outstretched index finger. UI can be operated like a normal touch interface.

5 Implementation of the Software Prototype

Different interaction concepts must be evaluated with a software prototype. The prototype was implemented with Unity3d [23] for an HTC-Vive and the Leap Motion. In Unity we used the SteamVR assets for connecting to the Vive [24]. For connecting the Leap Motion camera with Unity, we used their suggested plug-in. In addition, Leap Motion has prefabricated UI objects we can use for the virtual control concept. We used VRTK², a VR toolkit for rapidly building VR solutions in Unity3d, to use the controller more easily, e.g. when creating a laser pointer for teleporting or controlling UI elements [25].

The general structure of the implementation can be seen in Fig. 6. In Unity an implementation is divided into sub modules called "scene". For each concept a different application is built by using a different start scene, that reference to the main scene, the scene with the model of the city and the associated control scene. The *WelcomeScenes* are the entry points for the application. They welcome the user and have a confirmation button. Is this button clicked the *ScaleRotateExampleScene* is loaded where the model of the city is stored. Together with the *WelcomeScene* the associated *PlayerScene* is loaded. This scene contains all data for the different interaction concepts, so that pushing of the button is directly done within the interaction concept. The *MainScene*, which is also loaded when starting the *WelcomeScene*, contains all elements that should be available for every concept and in every scene.

² SteamVR Unity Toolkit.

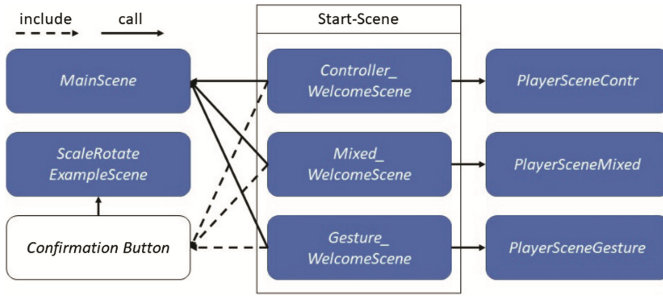


Fig. 6. Procedure organization of the scene flow

6 Evaluation

After an introduction in the structure of the evaluation, there will be an overview over the results (Sect. 6). In the end we conduct an evaluation with 30 participants of different ages and knowledge about VR. The distribution can be seen in Table 3.

Table 3. Distribution of the participants

Alter	Female	Male
18–25	1	5
26–35	4	16
36–45	0	4

6.1 Structure

To give all participants the same information, they get a written and spoken introduction in the beginning. The user is told how the controller works and what the evaluating issue was. The task was to explore a given software city and identify which classes, and therefore buildings, are interesting and why. For this the participants should use all necessary interaction possibilities.

At the beginning of every concept test they get an introduction to the different interaction possibilities (Fig. 7 bottom), which appear in front of the user. When the user performs the given interaction, the explanation disappears and the next information text shows up. Furthermore, when the user made all possible interactions, some informational icons appear on the controller – if any are used – to support the user (Fig. 7 top).



Fig. 7. Help system: top - help signs on the controller, bottom - explanations of an interaction possibility with video (Translation of the text: “Rotate & Scale (1): Make a fist with both hands and grab the city -> Interaction like with Google Maps used with touchscreen”).

At the end of every concept the participants complete a questionnaire based on ISO 9241-110. After testing all three concepts they fill out a questionnaire designed by ourselves for exactly this test case. Questions included “What was your preferred concept and why?” or “What concept would you use in a customer project”. Here, also previous experiences with VR, 3D applications or controller games were questioned.

6.2 Results

For better understanding please refer to Table 4 to see how many participants are meant with which description.

Table 4. Information to the description of the number of participants

Description	Number of participants
A few	2–5
Some	5–10
Various	10–20
Many	20–25
The most/almost all	25–30

Observation During Test Execution. There were different results, which we noticed directly while observing the user testing the concepts. For example, various participants had problems interacting with the UI elements. Initially we thought that the user would interact easily with them, because they knew the elements from everyday touch devices,

like their smartphone. But the lack of haptic feedback makes interaction more difficult for the user.

Although we discussed the scaling of the city with participants in the first evaluation, the participants in the final evaluation had problems with it. For some the city scaled in the wrong direction. While every interaction in the concepts is based on manipulating the city (except the teleportation in the physical control concept), these participants expect by pushing the trackpad up, that they move up, not the city, and hence, the city gets smaller not bigger.

The gesture control was difficult to learn for most of the participants. They said it was hard to avoid making unintended gestures. They often moved the city when they closed their hands to select a building or while opening them to release the city. The direction of the palm for scaling the city was also confusing for many of the participants. They expected, that each time they move their hand relative to the ground the same happens. However, Leap Motion expects the palm to be in the direction of the camera view. Some users complained about the cognitive effort in this type of gesture control, because of the big focus on hand poses and positions.

In virtual control concept the mini map obtained little approval. Some mentioned that they liked the map, but only for an overview, not for manipulation. They could indicate specific points of interest and move there e.g. by directly touching the map. In particular, the movement in different coordinate systems was confusing for most of the participants.

Comparison Questionnaire. At the end of all concept tests the participants completed a questionnaire to compare the tested concepts. The result of this questionnaire conveys the same picture as the surveillance of the tests. The virtual control has the fewest votes in questions regarding which interaction possibility the users found the best. This question was a multiple-choice question. The answer distribution can be seen in Table 5.

Table 5. Answer distribution of the question: “What concept for * pleases you the most?” (Original: “Welches Konzept für * hat dir am besten gefallen?”.)

*	Controller laser pointer	Controller button	Virtual control	Gestures one hand	Gestures two hands
Move	15	14	3	4	–
Scale	–	18	5	5	12
Rotate	–	14	2	11	15
Select	25	–	–	–	9

The test users voted for the virtual control as least. The rating of gesture control is often similar to the rating of physical control, with a tendency to the physical control. The most pronounced result here is for selecting a building. Some users explained this after the evaluation verbally: because the task was to analyze the city, they wanted a possibility to select a building from further away.

There were also the questions regarding users’ preferred interaction concept in general, and which realization the participants like the most. For this they could rate

them on a scale from one to five, where one is the worst and five is the best possible value.

The virtual control has the worst rates average of the three possibilities (2.7 for concept, 3.0 for the realization). The average for the concept rates for the physical control (4.30) is minimally higher than the average for the gesture control (4.0). However, in realization the physical control is much better (4.2) than the gesture control (3.5).

The tendency that the users like the controller and gesture concept the most, is seen in the answers to the question “Which concept would you use in a real project?”³. 14 participants answered with gesture control, the other 16 participants with physical control. The reason why they chose this answer was very similar amongst all. The participants could answer this question with plain text. In sum the answers can be seen in Table 6.

Table 6. Count of specific qualities in the reasons why the choose a control

	Physical control	Direct user interaction
Easy to operate	4	2
Easy to learn	1	2
Intuitive	2	4
Go well with VR	0	4
Less mistakes	3	0
Exact/precise	3	0
Fast	2	0

Those who opt for the physical control justified their decision by stating that this form of control worked best, had least sources of mistakes and therefore, is the easiest and fastest control. One participant described his decision with the words “exact”, “precise” and “stable”. Another justified their answer with the sentence “Intuitive operation which is fast and easy to learn”.

A similar reason, but for the gesture control, was “It is the easiest way to operate [...] and easy to learn”. Two participants considered the gesture control as “natural”, many others as “intuitive”. Four mentioned that they decided for gesture control because it’s the most obvious and go best with VR.

In particular the physical control convinced with its speed, accuracy and low error susceptibility. The gesture control has its advantage in being felt as intuitive and consistent with VR. A relation to personal preferences of the participants, like if they used 3D- or VR-applications before, was not found.

The only possible relation could be with the age of the participant. Four of five participants in the age of 18–25 chose the gesture control in this question. Three of four in the age of 36–45 decided upon the controller concept. In the age of 26–35 the decisions were balanced. If this is a real correlation or just coincidence should be tested in a next step.

³ Original: “Welches Konzept würdest du am ehesten in einem Kundenprojekt nutzen?”.

ISO 9241/10 Questionnaire. We also used a customized ISO 9241/10 questionnaire to get more information about the usability of the different concepts. The physical control got the best rating in most of the questions, followed by the gesture control. Only in the category of error tolerance gesture control was the worst. In the category of self-descriptiveness, it was in some questions adjusted to the same height with the virtual control. The largest difference between the valuation of physical control to the other two concepts was seen in the question of time requirement.

7 Conclusion

The aim of this paper was to find out which interaction concepts are the best for controlling a virtual environment without haptic feedback. For this we looked for concepts that are felt as natural and intuitive as possible. After a research of existing controls and concepts we decided to implement and evaluate three different concepts leaned on the concepts of Mark. R. Mine.

Generally, gesture and physical control are liked equally. Physical control, such as using a controller, has the advantage being able to give haptic feedback, and is therefore more precise and easy to use. Because the user can feel different buttons, he doesn't use them accidentally and can learn their usage fast. The existence of help icons and text, which are located near the buttons, also supports the self-descriptiveness of this interaction concept. The possibility of an explicit tracking of the controllers also make it more precise and accurate.

Gesture control has the advantage that it's perceived as natural and appropriated for VR. Although the physical control is the best rated in nearly every question in the ISO questionnaire, half of the participants would use gesture control in a real project. Like every new technology gestures have to be learned and people have to get accustomed to it. If in the future there are well known gestures and the technology for tracking and recognizing a hand is good, the most immersive way to control a virtual environment could be gesture control. Until then everyone has to decide personally whether to go the safe way and use a controller, or if they want natural interaction and use gesture control.

To combine the best of both controls, it's conceivable to use gesture control with controllers. Here, the movement of the controllers would map the movement of a hand. Pushing different buttons could correspond to different hand poses. Also, the further development of controllers with proximity sensors should be observed. There, the user can use the benefits of a controller, like haptic feedback and the usage of buttons, and additionally utilize basic gestures.

The results of this work are subjective impressions of the participants, which are influenced by the type of the controls implementation. A user study with more participants, fully developed interactions and a real user task, like finding a specific class with measuring the expired time, is needed to get objective results. Also, an evaluation for another application, e.g. the virtual operation of a machine, could bring new insights how much a specific task influences the evaluation of different concepts.

References

1. Witmer, B.G., Singer, M.J.: Measuring presence in virtual environments: a presence questionnaire. *Presence: Teleoperators Virtual Environ.* **7**(3), 225–240 (1998)
2. Pantförder, D., Vogel-Heuser, B., Gramß, D., Schweizer, K.: Supporting operators in process controll tasks – benefits of interactive 3-D visualization. *IEEE Trans. Hum.-Mach. Syst.* **46**(6), 895–907 (2016)
3. Ulewicz, S., Pantförder, D., Vogel-Heuser, B.: Interdisciplinary communication and comprehension in factory automation engineering – a concept for an immersive virtual environment. *IFAC-PapersOnLine* **49**(19), 227–232 (2016)
4. Pausch, R., Proffitt, D., Williams, G.: Quantifying immersion in virtual reality. In: *The Art and Interdisciplinary Programs of SIGGRAPH 1997. Visual Proceedings*, pp. 13–18. ACM SIGGRAPH, New York (1997)
5. DIN Deutsches Institut für Normung e.V.: *Ergonomie der Mensch-System-Interaktion – Teil 110: Grundsätze der Dialoggestaltung. DIN EN ISO 9241-110* (2008)
6. Nielson, J.: *Usability Engineering*. Kaufmann, San Diego (1994)
7. Bowman, D.A., Hodges, L.F.: An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In: van Dam, A. (ed.) *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, p. 35–ff. ACM, New York (1997)
8. Mine, M.R.: *Virtual Environment Interaction Techniques* (1995)
9. Nazemi, K.: Information visualization. *Adaptive Semantics Visualization. SCI*, vol. 646, pp. 17–68. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30816-6_2
10. Sasse, M.A., Johnson, C., Johnson, C.W.: International federation for information processing: human-computer interaction. In: *IFIP TC 13 International Conference on Human-Computer Interaction, INTERACT 1999*, vol. 1, Edinburgh (1999)
11. Kurtenbach, G., Hulteen, E.: Gestures in human-computer communication. In: Laurel, B. (ed.) *The Art of Human-Computer Interface Design*, pp. 309–317. Addison-Wesley Publishing Co., Boston (1990)
12. Gartner: IT Glossar. *Gesture Control*. <http://www.gartner.com/it-glossary/gesture-control/>. Accessed 29 Nov 2016
13. Harling, P.A., Edwards, A.D.N.: Hand tension as a gesture segmentation cue. In: Harling, P.A. (ed.) *Proceedings of Gesture Workshop Progress in Gestural Interaction 1996*, 19th March 1996, pp. 75–88. Springer, London (1997). <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.52.9492>. University of York
14. Robo Recall. <https://www.roadtovr.com/robo-recall-mod-adds-full-locomotion-address-room-scale-niggles/>. Accessed 8 May 2018
15. Stormborn Studio – *Runes: The Forgotten Path*, a VR wizard adventure. <http://www.kickstarter.com/projects/1501963972/runes-the-forgotten-path-a-vr-wizard-adventure>. Accessed 13 May 2017
16. Chastine, J., Kosoris, N., Skelton, J.: A study of gesture-based first person control. In: Mehdi, Q. (ed.) *18th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational & Serious Games (CGAMES)*, pp. 79–86. IEEE, Piscataway (2013)
17. Colgan, A.: 5 experiments bleeding edge vr locomotion. <http://blog.leapmotion.com/5-experiments-bleeding-edge-vr-locomotion/>. Accessed 13 May 2017
18. CodeCharta. <http://www.maibornwolff.de/codecharta>. Accessed 8 Feb 2018
19. McMahan, R.P., Bowman, D.A., Zielinski, D.J., Brady, R.B.: Evaluation display fidelity and interaction fidelity in a virtual reality game. *IEEE Trans. Vis. Comput. Graph.* **18**(4), 626–633 (2012)

20. HTC Vive. <https://www.vive.com>. Accessed 8 Feb 2018
21. Leap Motion. <http://www.leapmotion.com>. Accessed 8 Feb 2018
22. Norman, D.: Gestural Control: the Good, the Bad, and the Ugly. <https://www.linkedin.com/pulse/20140320012035-12181762-gestural-control-the-good-the-bad-and-the-ugly>. Accessed 30 Jan 2017
23. Unity: <https://unity3d.com/>. Accessed 8 Feb 2018
24. Steam VR: <store.steampowered.com/steamvr>. Accessed 8 Feb 2018
25. VRTK. <https://vrtoolkit.readme.io/>. Accessed 8 Feb 2018