Thomas Jung*, Johann Habakuk Israel, Ruben Ahlhelm and Patrick Bauer

Constraint-based bare-hand immersive 3D modelling

https://doi.org/10.1515/icom-2023-0013
Received January 24, 2022; accepted May 18, 2023;
published online May 31, 2023

Abstract: Three-dimensional user interfaces that are controlled by the user's bare hands are mostly based on purely gesture-based interaction techniques. However, these interfaces are often slow and error prone. Especially in the field of immersive 3D modelling, gestures are unsuitable because they complicate and delay the modelling process. To address these problems, we present a new gesture-free 3D modelling technique called "3D touch-and-drag", which allows users to select vertices by approaching them and to terminate operations by moving the 3D cursor (e.g. the forefinger) away from the constraint geometry (e.g. a straight line or a plane). Our proposed technique makes it possible to transfer the existing 3D modelling concepts ("3D widgets") to virtual environments, as shown by an experimental 3D modelling tool. The gesture-free bare-hand interaction also improves the possibility of tactile feedback during 3D manipulation. We compared different modelling techniques for controlling the 3D widgets. We found that controller-based techniques are significantly faster than finger-tracking-based techniques. The 3D touch-and-drag technique is about as fast as gesture-based interactions. Mouse interaction in a two-dimensional GUI is only slightly faster than the 3D modelling techniques. Since our proposed technique has proven to be at least equivalent to gesture-based interaction techniques in terms of accuracy and efficiency, its further development using more accurate tracking techniques seems promising to exploit the advantages of hands-free and gesture-free interaction for immersive 3D modelling.

Keywords: 3D manipulation; 3D modelling; constraints; finger tracking

***Corresponding author: Thomas Jung**, HTW Berlin, University of Applied Sciences, Berlin, Germany, E-mail: Thomas.Jung@HTW-Berlin.de
Johann Habakuk Israel, **Ruben Ahlhelm and Patrick Bauer**, HTW Berlin, University of Applied Sciences, Berlin, Germany,
E-mail: JohannHabakuk.Israel@HTW-Berlin.de (J.H. Israel),
rahlhelm@live.de (R. Ahlhelm), patrick.jean.bauer@googlemail.com
(P. Bauer)

1 Introduction

Most computer systems have graphical user interfaces (GUIs) that include windows, icons, menus and pointers; therefore, common off-the-shelf geometric modelling systems rely on mouse and keyboard interactions. When 2D GUI-based systems are used for 3D modelling tasks, the problem of manipulating 3D objects with a two-degree-of-freedom (DOF) input device, such as a mouse or a graphics tablet, is usually solved by mapping the 2D movements of the input devices onto working planes in the 3D modelling environment. Such desktop-based computer-aided design (CAD) systems are the most widely used tools for creating and manipulating 3D objects. Difficulties in interacting with such systems are often caused by the complexity and inappropriate design of the user interfaces, rather than the 3D manipulation of objects [1]. Within CAD systems, 3D objects are often manipulated not directly, but via 3D widgets – artificial graphical elements that can be easily grasped and moved with the interaction devices [2–5]. This type of interaction has been shown to have many advantages: positioning tasks can be performed with high accuracy, and users do not tire quickly because they can rest their hands on a table.

Typical immersive 3D modelling systems include sketching [6] and sculpting applications [7]. They usually employ 3D controllers, such as styluses, brushes, or standard VR controllers with six DOF as input devices. The movement of the input devices are typically mapped one-to-one to operations in the digital geometry. Controlling such 6 DOF input devices is an initial challenge for most users, but their performance (in terms of modelling accuracy) improves after a short period of training [8]. Immersive modelling interfaces typically allow users to move freely and view 3D objects from different perspectives in relation to their own bodies. In [9], by comparing the two-dimensional (2D) and 3D sketches of 24 interior designers, the authors showed that the designers moved faster, sketched for significantly longer and had more detail in the 3D condition, suggesting that the immersive 3D modelling systems supported creative thinking and design [10, 11].

However, users are often not very comfortable with unrestricted freehand interaction in 3D spaces because the (passive) haptic support that is available for drawing on a table, modelling clay or many other everyday tasks is usually not available in immersive environments. In physical interaction scenarios, physical constraints are often applied and the number of DOF reduced to reduce interaction complexity. Stuerzlinger and Wingrave, among others, suggest that this should also be done in immersive environments [12]. Other limitations of accurate interactions in immersive 3D environments are that the accuracy of input devices is typically low, users are less able to estimate distances in the z-direction, and working freehand for long periods of time can be tiring, leading to "gorilla arm syndrome" [13].

The techniques used in current geometric modelling systems rely heavily on 2D input; therefore it is difficult to transfer modelling techniques into immersive modelling environments which employ 3d interaction techniques. Although considerable work has been done in the development of modelling interfaces that combine 2D and 3D interaction techniques [14–16], there are no modelling systems that offer the same or similar functionality for both 2D GUI-based and immersive interfaces. In this paper, we call such 3D modelling systems "hybrid modelling systems".

A major emerging trend is the use of finger tracking to avoid the use of 3D controllers. With a few exceptions (e.g. [17]), such user interfaces require the execution of gestures (e.g. pinch or grab) to select or manipulate objects. However, the use of gestures intensifies the accuracy concerns because it is difficult for users to maintain the target position when performing a gesture [18]. Furthermore, gestures used to issue commands can be considered semaphoric (i.e. communicative) gestures (as opposed to manipulative gestures) [19]; these gestures need to be learned in advance by users; they require a lot of attention and interrupt the creative action-reflection design processes [20]. Therefore, we argue that it would be advantageous to avoid gestures during modelling interactions.

The use of fingers for interaction also allows the development of user interfaces based on (passive) haptic feedback, which can allow the user to feel constraint geometries such as planes. Passive haptic feedback can improve performance [21, 22] and enhance realism [23]. However, in gesture-based interfaces, it is difficult to implement and make the user feel these constraints on the fingertips, because when the hand is closed (in pinch and grasp gestures), the fingers cannot remain in the desired position on a passive haptic display, but have to move towards each other. We argue that this is another reason to avoid gestures.

This paper introduces the new modelling technique "3D touch-and-drag" (3Dtad) for immersive environments, which allows users to interact with their fingers but avoids gestures to increase accuracy. This interaction technique uses constraints to facilitate interactions and fits into the concept of hybrid geometric modelling systems.

In the following sections, we give an overview of related work, describe the 3Dtad interaction technique [24, 25] in detail, and present studies which compare it to controller-based and gesture-based interaction techniques.

2 Related work

2.1 Immersive 3D modelling and constraints

In one of the first interactive geometric modelling systems, the user wore a head-mounted display and interacted using a 3D input device (a "wand") that moved control points on free-form surfaces [26]. Subsequently, Butterworth et al. developed a fully immersive 3D modelling tool ("3 dm") [27] in which the user could interact with objects using a 6-DOF mouse. The system had polygonal modelling components and supported extrusions; it also provided 3D menus for selecting modelling operations. Butterworth et al. saw weaknesses in the lack of constraints in their system. To overcome these weaknesses, they introduced snap-to-grid and snap-to-plane modes.

The idea of using constraints to simplify interactions and increase accuracy in immersive 3D modelling has since found its way into a variety of modelling systems: Bowman et al. developed an immersive modelling system specifically for the design of animal habitats [28]. Part of the system was input using a pen and tablet. The physical surface of the device constrained the input so that the user only had to control two DOF. Zhong and Müller-Wittig [29] limited the DOF in their immersive solid modelling system through a constraint solving mechanism. Using a 3D input device, users could then control only one or two DOFs, depending on the context. Piekarski and Thomas developed an augmented reality geometric modelling system [30] in which 2D cursor positions were projected onto distant work planes that constrained the modelling operations. Shen et al. developed an augmented reality modelling system [31] in which the position of a tracked marker was mapped onto the surface of 3D scene objects during modelling ("grid-and-snap mode"). Keefe et al. implemented single and bi-manual interaction techniques using force feedback devices to constrain freehand drawing movements. They found empirical evidence for the improvement in accuracy compared to unrestricted drawing [32]. Wacker et al. showed that physical guides can improve the accuracy of 3D sketching in augmented reality without reducing performance [33].

2.2 Bare-hand human-computer interaction

Initially, direct hand manipulation of 3D objects required the use of data gloves. Nishino et al. presented an early gesture-based 3D modelling system using superquadrics, which allowed users to deform shapes (e.g. by tapering or bending). The users wore CyberGloves on both hands [34]. Matsumiya et al. developed an immersive system for modelling implicit surfaces using the metaphor of clay work [35]. Fingertip positions and hand gestures were tracked using a data glove. Kim et al. first developed a bare-hand tracking system so that users could perform various modelling operations with hand gestures (e.g. pinching) [36]. For example, a clef could be freely sketched in 3D space.

Other researchers have described modelling systems in which users interact with their bare hands. For example, Dave et al. used Microsoft Kinect to select and control manipulation tasks in a non-immersive CAD environment [37]. Jang et al. developed an immersive sculpting interface based on finger tracking using a Leap Motion sensor device [38, 39]. Vinayaj et al. developed a gestureless gesture-free bare-hand modelling system using the "virtual pottery metaphor" with a Leap Motion device [17]. Weichel et al. presented a mixed reality system for designing objects for fabrication that provided multiple gestures, such as creating, manipulating and cutting objects using a Kinect [40]. The system developed by Schkolne et al. enabled users to model large but coarse and less accurate models in a short time using their bare hands [41]. Zhang et al. developed an immersive mid-air sketching system for tree modelling [42].

3D mid-air hand interaction, often with the use of gestures, is a common technique for the control of immersive systems [43]. A study of bare-hand interactions outside the context of 3D modelling has shown that uninstrumented pointing in mid-air is less efficient than pointing with a 2D mouse [44]. Speeds and accuracies for docking tasks for unconstrained, bare-handed interactions in mid-air were comparable to those for a constrained device [45]. Users seemed to favour grasping gestures over pinching gestures [46]. Object reconstruction is more accurate when users create 3D lines using control points than when they draw these lines freely in 3D space; however, using control points is also slow [47]. Ricca et al. found that the presence of the user's hand representation is not necessary when performing a tool-based motor skill task in a VR trainer [48].

2.3 Hybrid modelling systems

Certain systems combine the advantages of 2D user interfaces (such as precision) and immersive user interfaces in hybrid cross-device scenarios (e.g. [49]). However, to the best of our knowledge, there are no modelling systems that work in a similar way for immersive environments and 2D GUIs.

Such hybrid modelling systems should allow users to transfer experience with modelling techniques from the 2D variant of a modelling tool to an immersive modelling environment. Commercial modelling tools are usually not open source, making it difficult for developers to create hybrid modelling systems. A rare example is the immersive version of SketchUp developed by Mine et al. [50], in which manipulation operations requiring low accuracy were performed using a 3D input device, and accurate positioning was achieved using a touchscreen.

This chapter has introduced a number of immersive 3D modelling systems, some of which use hands-free interaction. However, there is still no system that allows hands-free constraint-based 3D modelling without the use of gestures. The contribution of this paper is to present an interaction technique called "3D touch and drag" that meets these requirements. It is presented in the next chapter, together with an explanation of the advantages of gestureless interaction.

3 Modelling technique "3D touch-and-drag"

Three-dimensional modelling systems often require the selection of vertices or control points to start the manipulation operations. In 2D environments (e.g. desktop CAD systems), this selection can be performed by clicking a mouse button when the mouse cursor hovers over a vertex. In order to ease the selection, visual feedback is provided, usually in the form of a mark around the touched point over which the mouse cursor hovers. Without this feedback, users would have to focus much more when choosing vertices.

Current immersive modelling systems also expect users to explicitly perform selection actions, e.g. for selecting vertices, either by clicking on a controller button or, as in bare-hand interaction scenarios, by performing a gesture, such as pinch or air-tap gestures [51]. When gestures are employed, it is not enough to just start them, but also to complete them by performing a reverse movement (e.g. opening the hand or finishing the touch of two fingers), which involves additional movements that can shift the interaction location.

In 3D environments, vertices could also be selected solely by touching them with a 3D cursor. However, once selected, a technique would be needed to release the vertex again. This is where our approach comes into play: the moment a vertex is touched, a (invisible) plane is generated with the vertex in the centre on which it can be freely dragged. However, if the user moves his finger a defined distance above or below this surface, this is interpreted as a de-selection and the vertex remains in its last position on the plane.

The four phases of this interaction technique, which we call “3D touch-and-drag” (3Dtad), can be seen in Figure 1; it shows the scaling of a teapot in two dimensions. In the beginning, the distances to all selectable vertices are calculated (Figure 1(a)). If the distance to a vertex falls below a certain threshold, this vertex is selected and the plane is activated (Figure 1(b)). As long as the 3D cursor is not too far away from the plane, the vertex is moved according to the cursor position projected onto the plane (Figure 1(c)). In this example, the cursor movement causes the teapot to scale. The operation ends when the distance of the 3D cursor to the plane exceeds the threshold (Figure 1(d)). In certain cases the constraint geometry could be a straight line instead of a plane. The respective constraint geometry depends on the application and must be defined by the developers of the 3D modeling tool. For example, when modeling solids of revolution, a plane spanned by the axis of symmetry and the selected point can be used (Figure 3).

The basic algorithm is described in Figure 2. The two states “State 1” and “State 2” correspond to the classification given by Buxton [52]. In addition to what is described above, not only vertices but also edges, or control points can be moved. Those elements are called object parts. In special cases, the object parts can also refer to whole meshes. In principle, multiple object parts can be selected. The manipulations of one object are then applied to all of them.

This interaction techniques tries to reduce complexity by constraining the possible movements to one (in case of a line) or two (in case of a plane) spatial degrees of freedom. The 3D manipulations can be performed in a manner analogous to manipulations in GUI-based systems, e.g. by using 3D widgets. This is supposed to allow transfer of existing knowledge to immersive environments, reducing the learning effort. By visualising the manipulation plane, we aim to let the user “feel” the constraints.

The most important advantage of 3Dtad is that selection gestures can be avoided. As described above, the recognition of gestures is error-prone and performing gesture (and their counterpart movements) involves much hand and finger movements which can lead to unexpected shifts of the interaction center. Furthermore, the applicable gestures are usually not communicated by the interface to the user and require a certain level of experience. With 3Dtad technique we tried to reduce those disadvantages and thus

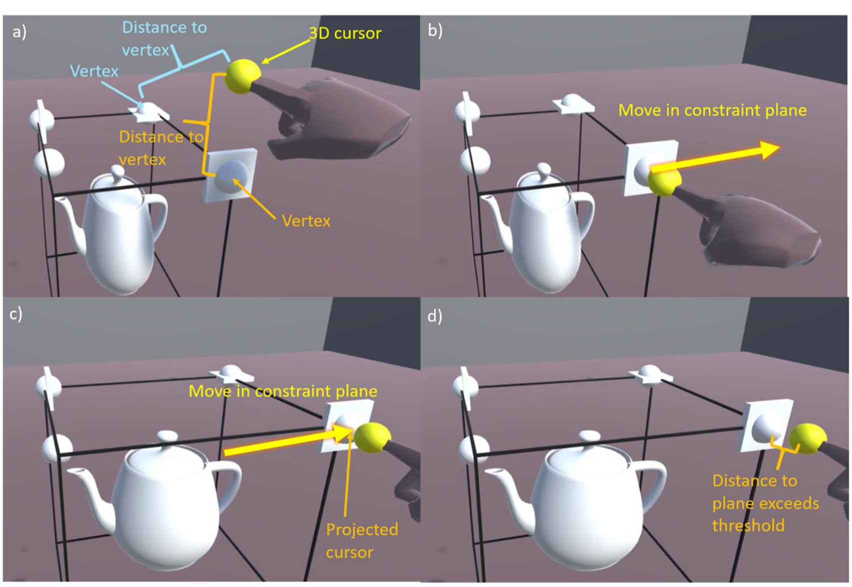

Figure 1: Constraint-based bare-hand immersive 3D modelling: 3Dtad with a plane as the constraint geometry.

```
State 1:

while no object parts have been selected

  if distance between 3D cursor and object part falls below threshold value
  {
    select object part
    define constraint geometry which includes object part
    begin modelling operation
    clear list
    goto State 2
  }

State 2:

while an object part is selected

  if the distance between 3D cursor and constraint geometry stays below threshold
  {
    add 3D cursor position to list
    project 3D cursor onto constraint geometry
    update modelling operation based on the projected 3D cursor
  }
  else
  {
    search pos in list where user starts to end operation
    project this pos onto constraint geometry
    finish modelling operation at projected pos
    deselect object part
    goto State 1
  }
```

Figure 2: 3Dtad algorithm. The commands in bold describe the determination of the end position. For more information, see Section 4.2.6.

expect it to increase performance and accuracy in modelling related selection actions compared to gesture based interactions.

Clearly, a complete modelling tool requires mode-switching interfaces [53] (e.g. menus) and a travel technique (camera control). A full investigation of 3Dtad would also have to include such interactions, but this is outside the scope of this paper.

4 Evaluation

Our evaluation of the 3Dtad technique was motivated by the following research questions:

- What are the main factors influencing gestureless 3D polygonal modelling?
- Does the 3Dtad technique offer more advantages than gesture-based 3D modelling approaches?

We evaluated our modelling technique in three steps:

1. Develop a first prototype of the 3Dtad input system and integrating it into an existing experimental 3D modelling system with a 2D GUI (i.e. proof of concept).
2. Conduct a user study based on a simplified application case (e.g. moving a sphere along a line) to identify the essential parameters and investigate the significant influencing factors.
3. Conduct a user study to explore the benefits of 3Dtad technique compared to gesture-based interactions using a more realistic use case (scaling an object in 3D space).

4.1 Step 1: integration of 3D touch-and-drag into Artist3D

To evaluate the new technique, we first integrated it into an experimental polygonal modelling system called Artist3D [54]. In this system, all modelling operations are restricted to planes or lines. The modelling operations and planes are implicitly selected when the user selects an edge or vertex. For example, the solids of revolution can be designed by starting with a cone. The user can insert a ring by touching a vertical edge and adjust the height and radius of the ring by moving the vertex with the cursor. In this case the movement is restricted to a modelling plane spanned by the vertical axis of the object and the vertex (Figure 3).

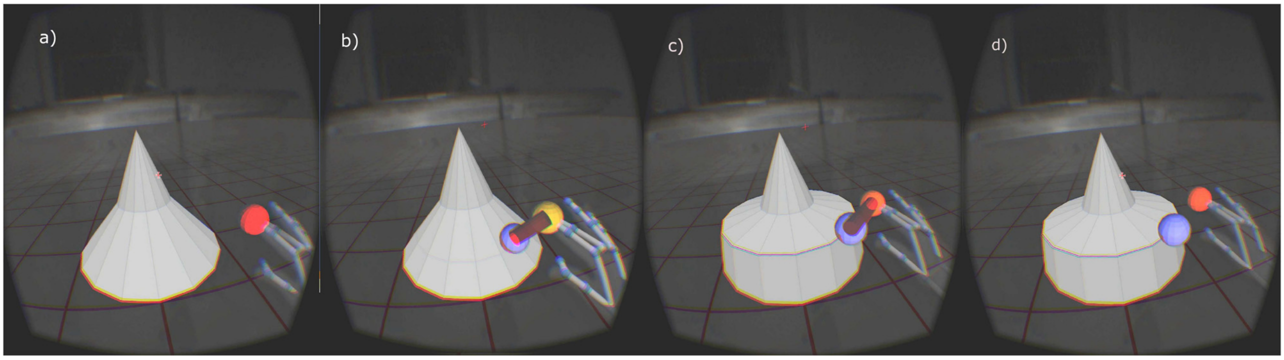

Figure 3: Constraint-based bare-hand immersive 3D modelling: 3Dtad in Artist3D. (a) The forefinger approaches the edge; (b) the ring is inserted, because the distance between the forefinger and the edge falls below the threshold value; (c) the radius and height of the ring change while the finger remains in the plane defined by the symmetric axis and the point; (d) the operation ends because the distance from the finger to the plane exceeds the threshold value.

In Artist3D, modelling operations are normally performed using the standard 2D touch and drag technique with a mouse or touch screen. In the 3Dtad technique, as described in the previous section, modelling operations are started by approaching a vertex and finished by leaving the modelling plane or line. We successfully implemented an immersive version of this interaction technique in Artist3D using a Leap Motion device and a head-mounted display. While we found that the modelling operations worked well, the menus need to be revised and a navigation interface needs to be integrated in the next version of the system.

4.2 Step 2: user study for "moving a sphere along a line"

In principle, the 3Dtad technique works with any kind of constraint. For our first study we chose the simplest type of constraint, a straight line. The task was to move a sphere along this line.

4.2.1 Participants

Fourteen volunteers took part in the study. The participants were all men between the ages of 20 and 39. One participant was left-handed, eleven were right-handed and two were ambidextrous. Thirteen had experience with computer games and eight had experience with head-mounted displays. Participants were asked to complete a training phase in each session before performing a task 20 times with different start and end positions (Figure 4).

4.2.2 Setup

The test scenes were presented via a head-mounted display (HTC VIVE [55]). Participants had to select a sphere at the start position and move it to the target position, where they had to release it. For visual orientation, we displayed a room with simple textured walls and pillars (see Figure 4).

In our analysis of related work, we found that most studies that did not develop their own finger tracking device used either a Microsoft Kinect system or a Leap Motion finger tracking system. The Leap Motion system has relatively high accuracy, at least when it is firmly attached to a table [56]. We therefore included it in our setup. As controller-based interactions in immersive environments are currently the most widely used, we also included this technique as a reference. Therefore, we designed the following conditions employing three devices:

- Hand tracking with a stationary Microsoft Kinect v2 ("Kinect")
- Forefinger tracking with a Leap Motion device attached to the head-mounted display ("Leap Motion")
- Position tracking with a wireless VIVE controller ("Controller")

4.2.3 Tasks

In the three conditions, subjects were asked to move the sphere using 3Dtad, a pinch gesture ("pinch"), or a button click ("button"). The distances, positions, sizes and thresholds were adjusted before the test. The diameter of the sphere was 10 cm, and the distance between the centre of the sphere and the cursor had to be less than 3.5 cm to start

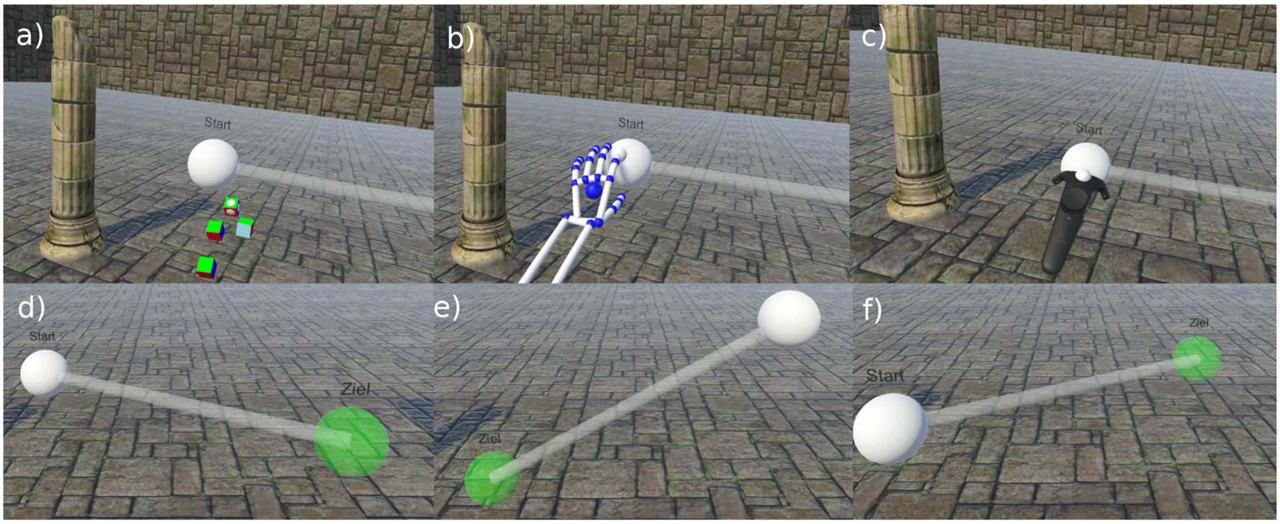

Figure 4: User test: Move a ball along a line. The user's initial views with hands and controller shown. (a) Kinect; (b) VIVE controller; (c) Leap Motion. (d–f) Three examples of tasks.

the operation. However, a maximum distance of 4 cm was allowed between the centre of the sphere and the cursor when performing the pinch gesture. When the target position (60 cm away) was reached, the task could be completed either by ending the gesture or by moving the cursor away from the line to a distance of at least 5 cm ("3Dtad").

We used the pinch gesture detection from the Leap Motion SDK V2 for Unity. A pinch gesture would be detected if the distance between the forefinger and the thumb ("Leap Motion") was less than 3 cm. When using Microsoft Kinect, the distance between the "thumb" and "hand" points in the skeleton was used, with a threshold of 9 cm. When using the controller, a button was used to control the operation. Because of the speed-accuracy trade-off [57], participants were asked to focus on execution time ("Fast") for the first ten trials, and on accuracy ("Accuracy") for the last ten trials.

As dependent variables, we measured the movement times (MTs) in the first ten trials and the deviations from the target position (error) at the end of the interaction in the second ten trials.

We tried enabling ("Visible") and disabling the display of the fingers, skeleton, or controller in the 3Dtad sessions, because although displaying the fingers or controllers could increase the sense of presence [58], it could also occlude the sphere, potentially hindering accurate interactions. In all variants, we displayed the cursor as a sphere.

At the end of the study, all subjects gave their subjective ratings of the different devices. We used a five-question questionnaire adapted from [59]. In addition, participants indicated whether they found the display of hands or controller models helpful.

Task difficulty can depend on both the direction of movement [60] and the difference between the start and end points in the user's line of sight [61]; therefore we decided to vary the trajectories. The following set of ten tasks was used in all sessions (coordinates are given in metres):

1. From the left $(-0.3, 0, 0)$ to the right $(0.3, 0, 0)$
2. From the right to the left
3. From the top left $(-0.26, 0.15, 0)$ to the bottom right $(0.26, -0.15, 0)$
4. From the bottom left $(-0.26, -0.15, 0)$ to the top right $(0.26, 0.15, 0)$
5. From the bottom right to the top left
6. From the top right to the bottom left
7. From the front left $(-0.26, 0, -0.15)$ to the back right $(0.26, 0, 0.15)$
8. From the back left $(-0.26, 0, 0.15)$ to the front right $(0.26, 0, -0.15)$
9. From the back right to the front left
10. From the front right to the back left

The origin of the coordinate frame was 30 cm below the user's eyes at a distance of 30 cm. To exclude the influence of handedness, the participants had the choice of using either their left hand or right hand for each trial.

The controller trials were designed to allow us to fundamentally investigate 3Dtad for accuracy and efficiency. The 3Dtad technique with both controller and finger tracking was used to analyse the differences caused by tracking. The Leap Motion trials allowed us to compare gesture-based interactions with gesture-free interactions.

We randomised the order of the three devices used: Kinect, Leap Motion and controller. Users performed both 3Dtad and pinch sessions.

4.2.4 Execution

All participants were asked to perform the tasks in different scenarios at least 120 times. Although each trial took only a few seconds, many participants experienced fatigue due to “gorilla arm syndrome”. When this occurred, we excused them from performing the Kinect scenario, which meant that only seven participants performed in this scenario.

The HTC VIVE, Leap Motion and Kinect devices all sent and received infrared light to and from space. We therefore assumed that the systems could interfere with each other and planned to exclude failed trials, which would mainly be caused by tracking errors.

We needed a minimum level of accuracy to complete the operation successfully. The projection of the moving sphere on the line at the end of the trial had to be within 3 cm of the target position. Therefore, failed trials occurred when the users did not finish the operations at the target or when the tracking failed during the operation. Initially, we detected failed trials automatically. Later, we verified the results by visually inspecting the recorded motion data.

Table 1 shows the differences in tracking quality between the three devices used in our research. The tracking quality of the Kinect scenario was so poor that we excluded the Kinect trials from further evaluation.

At the end of a 3Dtad trial, the user moved the 3D cursor away from the constraint geometry and thus away from the target (hereafter “final phase”). From the beginning of the final phase, the direction of movement should be almost perpendicular to the direction of the line. We used a threshold of 45° to identify the start of the final phase.

In any case, the cursor position was projected onto the line.

4.2.5 Results and discussion: efficiency

For the efficiency analysis, we used the trials in which the users focused on fast execution. We recorded 650 successful trials from our fourteen subjects (Table 2).

For the analysis, we recorded the 3D coordinates, the execution times (Table 2) and the accuracies at the position where the final phase started (Figure 1(c)).

We assumed that it would be more difficult for users to complete the task if the start and end points had different z-values. Therefore, we first examined the MTs of the trajectories using the example of our reference scenario

Table 1: Failed trials.

Device	Mode	Visible	Failed trials with focus on execution time		Failed trials with focus on accuracy	
Controller	Button	Yes	3/130	2 %	2/130	2 %
	3Dtad	Yes	32/130	25 %	17/130	13 %
	3Dtad	No	23/130	18 %	23/130	18 %
Kinect	Pinch	Yes	24/70	34 %	24/70	34 %
	3Dtad	Yes	37/70	53 %	22/66	33 %
	3Dtad	No	28/60	47 %	25/60	42 %
Leap Motion	Pinch	Yes	29/140	21 %	18/140	13 %
	3Dtad	Yes	39/140	28 %	43/140	31 %
	3Dtad	No	24/130	18 %	26/130	20 %

Table 2: Results: n is the number of successful trials; MT is the average movement time in seconds; and SD is the standard deviation.

Device	Mode	Visible	n	With final phase		Without final phase	
				MT	SD	MT	SD
Controller	Click	Yes	127	–	–	1.20	0.54
	3Dtad	Yes	98	1.56	0.62	1.27	0.58
		No	107	1.45	0.40	1.18	0.34
Leap Motion	Pinch	Yes	111	–	–	1.56	0.57
	3Dtad	Yes	101	1.77	0.80	1.41	0.60
		No	106	1.74	0.52	1.42	0.43

("Controller", "Button click", and "Fast"). However, a two-way ANOVA did not show any significant effect of the trajectories on the MTs in this situation ($p = 0.43$).

In the questionnaire eleven out of thirteen participants reported that displaying the controller was helpful, and thirteen out of fourteen participants said the same about displaying hands in the Leap Motion scenario. In contrast, we found that MTs were significantly shorter when the hands (MT = 1.67 s, SD = 0.44 s vs. MT = 1.83 s, SD = 0.81 s; $t(73) = -2.02$, $p < 0.047$) or the controller models (MT = 1.41 s, SD = 0.39 s vs. MT = 1.51 s, SD = 0.56 s; $t(84) = -2.12$, $p < 0.038$) were not displayed (see Table 2).[1] Therefore, it is useful to examine the tracking on the basis of the trials in which the controller or the hands were not displayed. This was only the case for the 3Dtad trials. As expected, interactions with the controller were significantly faster (MT = 1.43 s, SD = 0.34 s) than those with the Leap Motion device (MT = 1.73 s, SD = 0.52 s); $t(91) = -6.73$, $p < 10^{-7}$).

It was possible to compare all the input techniques on the basis of the trials in which the controller and the hands were displayed. All tasks in our experiment were dragging tasks. When the user selects using a pinch gesture or button press, the dragging task corresponds to a pointing task for both selecting and terminating the interaction [62]. The 3Dtad interaction requires the user not to exceed a certain distance from the line. This is closer to a steering task. In this respect, 3Dtad could lead to longer MTs [63]. Therefore, we compared the MTs of our reference scenario (controller with button clicks) with those of 3Dtad, excluding the final phase. We found that the average MTs of the reference scenario were significantly shorter than the average MTs with 3Dtad ("Visible", MT = 1.18 s, SD = 1.09 s vs. MT = 1.28 s, SD = 1.13 s; $t(94) = -2.07$, $p < 0.042$) in our setting.

The main question was how the MTs of the 3Dtad with finger tracking behaved in comparison with gesture-based interactions. As mentioned above, the MTs of 3Dtad were divided into the movement phase of the sphere on the line and the final phase. Only one MT was calculated for the gesture-based variant, although ending the gesture would certainly take some time. We found that the median MTs were shorter for 3Dtad without the final phase (MT = 1.47 s, SD = 0.66 s) than for gesture-based trials (MT = 1.49 s, SD = 0.52 s), but these values were not statistically significant ($t(77) = 0.32$; $p = 0.75$).

We observed that subjects had difficulty selecting the sphere with the pinch gesture. However, we were unable to measure this phenomenon because the MTs were only recorded from the beginning of successful selections.

4.2.6 Results and discussion: accuracy

For the accuracy analysis, we used only those trials in which the users focused on accuracy. We recorded 671 successful trials from our fourteen subjects (Table 3). Again, we examined our results in the case of 3Dtad at the beginning and at the end of the final phase.

Consistent with the efficiency results, the errors in the Leap Motion scenario were significantly lower when the hands were not displayed (ERR = 1.06 cm, SD = 0.74 cm vs. ERR = 1.37 cm, SD = 0.88 cm; $t(79) = -2.43$, $p < 0.018$).

We found that the error at the beginning of the final phase was significantly lower (3Dtad, "invisible"; ERR = 0.77 cm, SD = 0.58 cm) than at the end of the phase with the controller (ERR = 1.06 cm, SD = 0.68 cm; $t(106) = -3.5$, $p < 0.0007$). Therefore, from now on, we will only consider the accuracy at the beginning of the final phase. This has already been implemented in Figure 2.

We assumed that 3Dtad would be more accurate than a pinch gesture in the finger tracking scenario. However, we did not find a significant effect on accuracy (ERR = 1.19 cm, SD = 1.19 cm vs. ERR = 1.04 cm, SD = 0.76 cm; $t(184) = 1.07$, $p = 0.83$). To analyse this effect in more detail, we would probably need better tracking accuracy than the Leap Motion device currently offered in our scenario.

4.2.7 Conclusions from study 1

We found that interactions with the Vive Controller are significantly faster than the interactions with the Leap Motion device, probably because of stable tracking. Users welcomed the display of hand or controller models, although this degrades the accuracy and efficiency of the operations. Contrary to our expectations, we did not find any significant differences in accuracy between 3Dtad and pinch-gesture-based interactions. The differences in MTs could not be conclusively assessed because we did not obtain the selection times.

4.3 Step 3: scaling task

Our second user test was to examine the MTs, including selection times. In the previous study, users preferred the

1 In this and all other significance tests, we used two-tailed paired t-tests ($\alpha = 0.05$) with missing data handled by listwise deletion. Therefore, the MTs in the significance test differed slightly from those of all users in Table 2.

Table 3: *Error*: average distances at the end of the movement in centimeters; SD is the standard deviation.

Device	Mode	Visible	*n*	With final phase		Without final phase	
				Error	SD	*Error*	SD
Controller	Click	Yes	128	–	–	0.75	0.60
	3Dtad	Yes	113	1.08	0.77	0.73	0.62
		No	107	1.06	0.68	0.77	0.58
Leap Motion	Pinch	Yes	122	–	–	1.02	0.76
	3Dtad	Yes	97	1.36	0.80	1.14	1.13
		No	104	1.15	0.82	0.94	0.95

display of the hand and controller models. Therefore, we chose this variant even though it had a negative effect on the MTs. To obtain the selection times, we designed a scaling task consisting of several steps. This task was inspired by the 3D widgets in Artist3D for resizing boxes and scaling objects.

4.3.1 Participants

Fifteen volunteers (five women and ten men aged between 19 and 55) took part in this study; seven participants had experience of virtual reality and nine had experience of 3D manipulation.

4.3.2 Setup and tasks

Users scaled two dimensions of an object three times, with interactions restricted to three orthogonal planes (Figure 5). The widget had an initial size of 40 × 30 × 30 cm. It was placed at a height of 1.3 m. Users were recommended an optimal location, marked by a green cylinder on the floor. From this position, users could see the widget diagonally from the top left and reach all interaction points directly without having to move.

To complete the task, users had to move a point in a plane three times at a time. Again, they used either their fingers or a VIVE controller (Figure 6). Finger tracking was

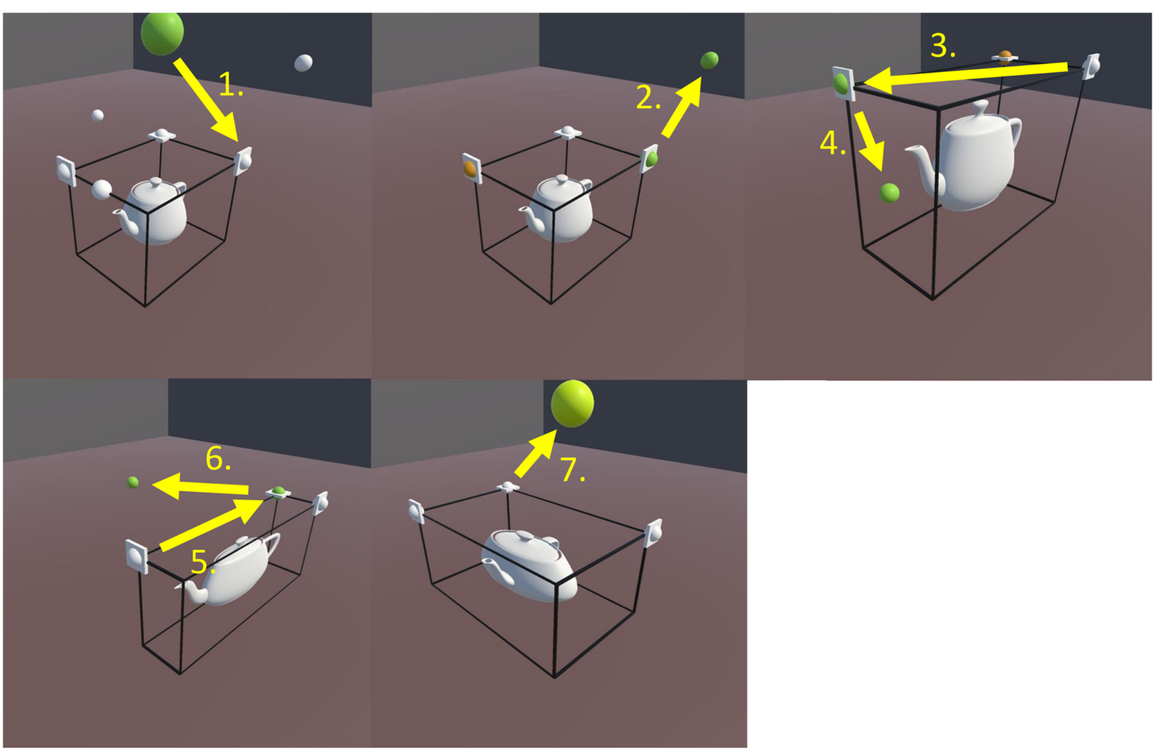

Figure 5: Scaling a teapot in three steps. Each sub-task is represented by a yellow arrow. The only manipulatable elements were the tree grey orthogonal planes at the corners of the cube widget.

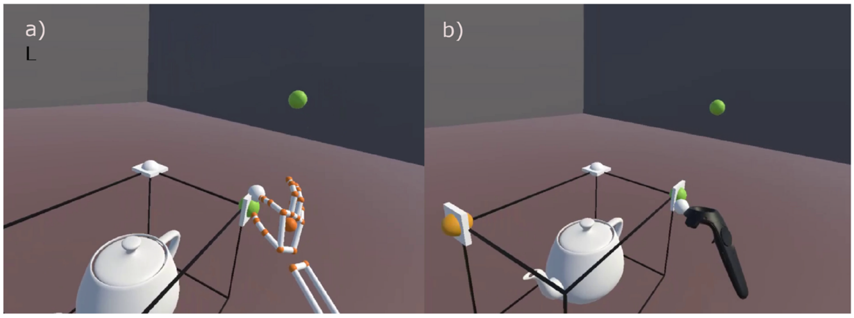

Figure 6: Scaling task (user's view). (a) Using Leap Motion; (b) using a controller.

implemented using a Leap Motion device and the Unity Leap Motion Core Assets version 4.

The interaction techniques used were 3Dtad, button click or pinch gesture. In all trials, users recorded their start and stop times by touching a large sphere (10 cm in diameter) hovering over the widget.

The 3D cursor was represented by a 4 cm diameter sphere, and the spheres to be selected ("interaction spheres") were 5 cm in diameter. The 3D cursor and the interaction sphere had to intersect at the beginning and at the end of an interaction, which corresponded to a required distance between the sphere centres of 4.5 cm. The position of the 3D cursor was projected perpendicular to the plane during the interaction and visualised by another interaction sphere. For the 3Dtad trials, this sphere and the 3D cursor had to intersect at all times. If this was no longer the case, the interaction was considered terminated. During scaling, the user had to maintain a distance of 4.5 cm and exceed this distance to exit the interaction. Pinch gestures were terminated again by moving the thumb and index finger 3 cm apart.

We measured the start time when selections were made to begin the scaling task and the end time. The selection times were thus included in the times between the scaling operations "approach" and "transition". The duration of the completion of the respective gesture or final phase was included in the scaling subtask. The mental preparation times were not investigated further, as we assumed that they were likely to be the same in all scenarios.

The trajectories of the seven subtasks can be found in Table 4.

Table 4: Scaling task: trajectories of subtasks; *d* is the distance in meters.

No.	Subtask	From	To	*d*
1	Approach	(0, 1.7, 0)	(0.2, 1.36, −0.16)	0.39
2	Scales *x*, *y*	(0.2, 1.36, −0.16)	(0.5, 1.6, −0.16)	0.38
3	Transition	(0.5, 1.6, −0.16)	(−0.2, 1.6, 0.15)	0.77
4	Scales *x*, *z*	(−0.2, 1.6, 0.15)	(−0.2, 1.36, 0)	0.28
5	Transition	(−0.2, 1.36, 0)	(0.5, 1.36, 0)	0.7
6	Scales *x*, *z*	(0.5, 1.36, 0)	(0.2, 1.36, 0.5)	0.58
7	Finish	(0.2, 1.36, 0.5)	(0, 1.7, 0)	0.64

4.3.3 Execution

We asked users to concentrate on completing the task as quickly as possible. We only wanted to study error-free interactions. Because this task required users to rotate their hand to reach points, it was to be expected that the tracking quality of the Leap Motion device would be affected by visual occlusion. Due to the necessary execution of several subtasks in a row, we expected increasing error rates.

Therefore, in this study, we asked users to perform a best possible trial. They had to do at least ten trials, but they could keep going until they felt they had the best possible trial. We recorded all the trials and only the trial with the shortest total duration was included. Because of this approach, we also excluded an explicit training phase.

As we are interested in developing a hybrid modelling system in the medium term, we decided to include a non-immersive 2D variant in the test, in which the user had to perform the tasks using a mouse. The 3D scene was displayed in full screen (Figure 5). The mouse positions were raycasted onto the three interaction planes.

Normally, users move the mouse only a few centimetres when performing the drag-and-drop operations. For comparability, we decided to set the mouse speed to a low level (2 out of 11 in the Windows 10 settings menu). As a result, the mouse paths on the table were only about three times smaller than the interaction paths in the virtual world (Table 5, column "In 3D" vs. column "On table").

All subjects completed questionnaires (NASA-TLX [64]) at the end of the study. We randomised the order of the four runs with the Leap Motion and Vive controllers. The mouse runs were performed last for convenience.

Table 5: Distances in centimetres in the scaling task. The results of the follow-up test are shown in the two right columns.

Test	Step 3			Follow-up test	
Subtask	In 3D	On screen	On table	On table 50 dpi	On iPad Pro
1	39	8.5	17	16	8.1
2	38	6.3	13	11	6.0
3	77	14.0	31	25	13.1
4	28	6.5	13	13	6.2
5	70	8.9	20	15	8.4
6	58	8.1	19	15	7.7
7	64	5.6	12	11	5.3
Total	374	57.9	125	106	54.8

Table 7: Median MTs in seconds for subtasks.

Subtask	Controller		Leap Motion		2D mouse
	Button	3Dtad	Pinch	3Dtad	
1	1.32	1.34	1.95	1.74	2.84
2	0.35	0.35	0.39	0.59	0.45
3	1.49	1.50	2.57	1.99	2.70
4	0.39	0.35	0.54	0.75	0.68
5	1.66	1.66	2.29	2.01	2.85
6	0.29	0.43	0.45	0.78	0.66
7	0.77	0.84	0.95	0.91	1.07
Total	6.28	6.45	9.14	8.78	11.24

4.3.4 Results and discussion

The MTs are shown in Table 6. Although we only evaluated the best trials of the users, we still found many unnecessary actions such as pinching gestures or clicks that did not lead to the selection of a vertex or the termination of interactions outside the target area. Again, users had problems performing the pinch gesture.

As in the first study, interaction with the Vive controller (3Dtad; $\mathrm{MT} = 7.07$ s, $\mathrm{SD} = 2.46$ s) was significantly faster than finger tracking with Leap Motion (3Dtad; $\mathrm{MT} = 10.52$ s, $\mathrm{SD} = 5.14$ s; $t(14) = -3.34$, $p < 0.005$).

However, our main focus was on the comparison between the two finger-tracking methods. The gesture-based approach ($\mathrm{MT} = 9.76$ s, $\mathrm{SD} = 3.38$ s) was on average slightly faster than the 3Dtad method ($\mathrm{MT} = 10.52$ s, $\mathrm{SD} = 5.14$ s), but the differences were not significant ($t(14) = -0.74$, $p > 0.46$).

As we only took into account users' best attempts, individual users' errors affected the results. For this reason, we have concentrated on the medians in the following.

The duration of the start of the pinch gesture was included in the approach/transition movement; therefore, we examined the MTs of the subtasks (Table 7).

Table 6: Results: MT is the average MTs in seconds, and SD is the standard deviation. AUA is the average number of detected unnecessary actions.

	Controller		Leap Motion		2D
	Button	3Dtad	Pinch	3Dtad	Mouse
MT	6.91	7.07	9.75	10.52	13.02
SD	3.23	2.46	3.47	5.14	3.79
AUA	0.2	0.6	1.6	0.6	0.4

Table 7 also shows that the median MTs of the subtasks with the finger tracking 1, 3, and 5 were substantially larger in the pinch sessions, whereas the MTs of subtasks 2, 4, and 6 were larger in the 3Dtad sessions. The extra effort required to perform the pinch gesture and to finish 3Dtad seems to be about the same. This aspect is illustrated by the visualisation of the temporal processes using a "virtual" test user, which summarizes the median MTs of the subtasks (Figure 7).

A surprising result was that in our test setup, using the 2D mouse resulted in significantly longer MTs ($\mathrm{MT} = 13.02$ s, $\mathrm{SD} = 3.79$ s) than using the Vive Controller with 3Dtad in the immersive variant ($\mathrm{MT} = 6.07$ s, $\mathrm{SD} = 2.46$ s; $t(14) = -8.3$, $p < 10\mathrm{E} - 6$). Users described the mouse scenario as "unrealistic", "unfamiliar", "more difficult due to the low mouse speed", and "more exhausting". The subjectively greater effort and higher frustration when using the

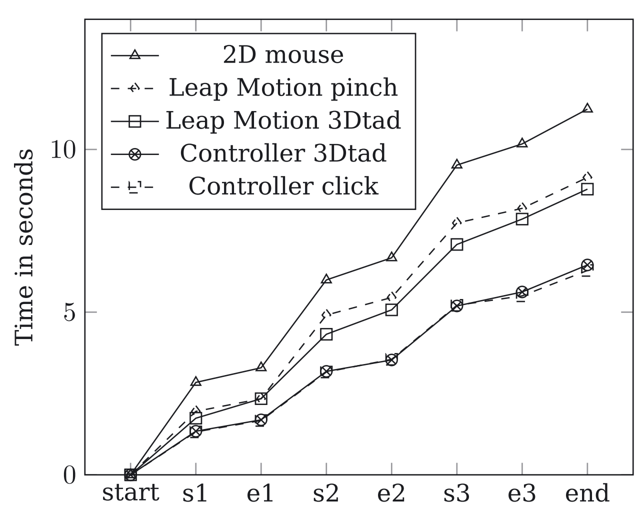

Figure 7: Subtasks of a virtual median test user, si and ei refer to the start and the end of a scaling sub-task. The courses for "Controller 3Dtad" and "Controller click" are very close to each other and are difficult to distinguish.

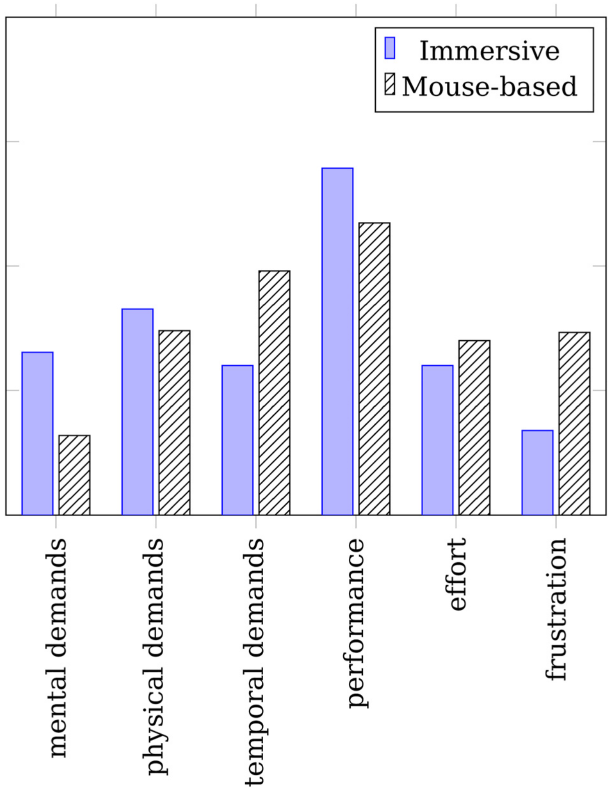

Figure 8: NASA task load index: immersive 3D interaction versus mouse-based interactions.

mouse can also be traced from the user-filled NASA-TLX questionnaires (Figure 8). Therefore, we decided to run a follow-up test using a mouse having different sensitivities.

4.3.5 Follow-up test

We used a gaming mouse (Speedlink Prime Z-DW) whose software allowed us to explicitly set the resolution. We chose resolutions that were as close as possible to the user's optimal settings (200 dpi, 800 dpi, 1600 dpi), as well as a variant that would lead to similar long mouse movements on the table as in the previous test (see Table 5). We also tested a touch-based interaction on an iPad Pro (Figure 9).

Six subjects (four men and two women, aged between 12 and 55) participated in this study. All subjects performed the tests in the same order (50 dpi, 200 dpi, 800 dpi, 1600 dpi, iPad Pro). We expected the latter tests to produce better results due to learning effects. Our aim was to get the best possible results for 2D interaction.

The 50 dpi setting gave similar results to the mouse used in the previous study. The best MTs were achieved with a sensitivity of 800 dpi. However, interactions with the mouse were then only about 10 % faster than those with the controller ("button"). The mouse paths were only about one centimetre per subtask. The MTs on the iPad Pro were slightly shorter, but also reached at least 75 % of the immersive Vive controller variant (Table 8).

Table 8: Median MTs in seconds for subtasks in the follow-up test.

Test	Step 3		Follow-up test (dpi)				iPad Pro
Subt.	Button	Mouse	50	200	800	1600	
1	1.32	2.84	2.68	1.76	1.35	1.60	1.18
2	0.35	0.45	0.49	0.43	0.33	0.40	0.18
3	1.49	2.70	2.71	1.62	1.30	1.44	1.20
4	0.39	0.68	0.45	0.34	0.42	0.42	0.32
5	1.66	2.85	2.70	1.60	1.50	1.37	1.25
6	0.29	0.66	0.42	0.32	0.28	0.33	0.08
7	0.77	1.07	1.14	0.66	0.58	0.53	0.50
Total	6.28	11.24	10.59	6.73	5.76	6.09	4.71

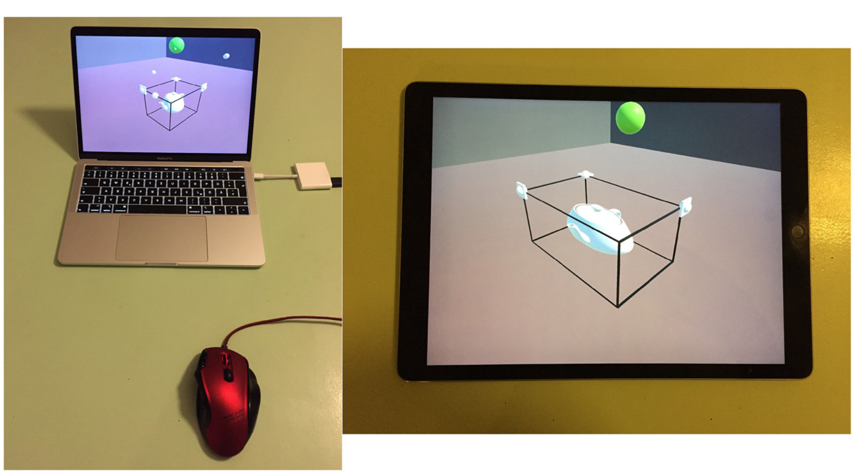

Figure 9: Experimental setup for scaling a teapot with a mouse and an iPad Pro.

5 Conclusions

We discussed the possibilities of developing hybrid 3D modelling tools (immersive/GUI) based on 3D widgets. We presented a new modelling technique called “3D touch-and-drag”, which allowed users to start a modelling operation by approaching a vertex. The operation was completed by removing the 3D cursor from the modelling plane or line (constraint geometry). We found that this technique could be easily integrated into an existing experimental 3D modelling system (Artist3D) in an immersive scenario.

In our first study, we found that using a controller resulted in significantly lower MTs than finger tracking. This could be due to the low reliability and accuracy of the Leap Motion device, which in our case was attached to a head-mounted display. We assumed that 3D interactions using the 3Dtad might be more accurate than using pinch gestures, but we did not find a significant effect on accuracy. We found that 3Dtad was about as efficient as using pinch gestures in our test scenarios. However, all of our test subjects were new to this type of interaction and should be considered novices. Pinch gestures were probably easier to understand as a concept than 3Dtad. In this respect, we are planning further experiments to investigate learning effects over longer periods of time. It may be that expert users can benefit more from the advantages of 3Dtad.

In our second study, we surprisingly found that interaction in the virtual environment was more efficient than in the 2D mouse and screen environment. A follow-up test showed that with optimal mouse settings, interaction times were slightly lower than in the immersive variant. However, the MTs depended on the length of the isomorphic trajectories, so reducing the box size in our user tests could improve the performance of the immersive version. Nevertheless, the results of this follow-up test are put into perspective by the fact that only 6 people participated in this test, compared to 15 participants in Study 2.

From the results of our studies we derive the findings that 3Dtad cannot outperform controller-based interaction both in accuracy and efficiency of modelling interactions. Furthermore no significant differences in these categories become visible when compared to gesture-based interaction. This is probably not an exceptional success, but it shows that 3Dtad can at least be used as an alternative to gesture-based modelling techniques, for example where gesture-based interaction is not appropriate. We are therefore optimistic that we will be able to offer future users an immersive user interface that uses constraints more efficiently than the classic 2D GUI.

Furthermore, as the 3Dtad technique allows controller-free 3D manipulation of objects without gestures, it allows haptic feedback to be integrated into 3D manipulation, which is likely to increase efficiency and accuracy. In the future, we plan to investigate the potential efficiency gains from passive or active haptic feedback. For example, encounter-type haptic displays [65] could be used to haptically represent the constraint geometry.

The weight of the tool tends to have a negative effect on MTs [57]; therefore, we expect uninstrumented finger tracking to be more efficient in principle than using a controller. This would need to be investigated with alternative finger tracking systems in future tests.

Acknowledgement: The authors would like to thank the participants of the study.
Author contributions: All the authors have accepted responsibility for the entire content of this submitted manuscript and approved submission.
Research funding: None declared.
Conflict of interest statement: The authors declare no conflicts of interest regarding this article.

References

1. Lee G., Eastman C. M., Taunk T., Ho C.-H. Usability principles and best practices for the user interface design of complex 3D architectural design and engineering tools. *Int. J. Hum. Comput. Stud.* 2010, *68*, 90–104.
2. Allan Bier E. Skitters and jacks: interactive 3D positioning tools. In *Proceedings of the 1986 Workshop on Interactive 3D Graphics (Chapel Hill, North Carolina, USA) (I3D ’86)*; ACM: New York, NY, USA, 1987, pp. 183–196.
3. Conner B. D., Snibbe S. S., Herndon K. P., Robbins D. C., Zeleznik R. C., van Dam A. Three-dimensional widgets. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics (Cambridge, Massachusetts, USA) (I3D ’92)*; ACM: New York, NY, USA, 1992, pp. 183–188.
4. Houde S. Iterative design of an interface for easy 3-D direct manipulation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Monterey, California, USA) (CHI ’92)*; ACM: New York, NY, USA, 1992, pp. 135–142.
5. Zeleznik R. C., Herndon K. P., Robbins D. C., Huang N., Meyer T., Parker N., Hughes J. F. An interactive 3D toolkit for constructing 3D widgets. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques (Anaheim, CA) (SIGGRAPH ’93)*; ACM: New York, NY, USA, 1993, pp. 81–84.
6. Keefe D. F., Acevedo Feliz D., Moscovich T., Laidlaw D. H., LaViola J. J. Jr. CavePainting: a fully immersive 3D artistic medium and interactive experience. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics (I3D ’01)*; ACM: New York, NY, USA, 2001, pp. 85–93.
7. Ponto K., Tredinnick R., Bartholomew A., Roy C., Szafir D., Greenheck D., Kohlmann J. SculptUp: a rapid, immersive 3D

modeling environment. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)*, 2013, pp. 199−200.
8. Wiese E., Israel J. H., Meyer A., Bongartz S. Investigating the learnability of immersive free-hand sketching. In *Proc. ACM SIGGRAPH/Eurographics Symposium on Sketch-Based Interfaces and Modeling SBIM*; ACM SIGGRAPH and the Eurographics Association, 2010, pp. 135−142.
9. Israel J. H., Wiese E., Mateescu M., Zöllner C., Stark R. Investigating three-dimensional sketching for early conceptual design — results from expert discussions and user studies. *Comput. Graph.* 2009, *33*, 462−473.
10. Stark R., Adenauer J., Israel J. H. Virtual reality technologies for creative design. In *CIRP Design 2012*; Springer: London, 2013, pp. 125−135.
11. Shneiderman B., Fischer G., Czerwinski M., Resnick M., Myers B., Candy L., Edmonds E., Eisenberg M., Giaccardi E., Hewett T., Jennings P., Kules B., Nakakoji K., Nunamaker J., Pausch R., Selker T., Sylvan E., Terry M. Creativity support tools: report from a U.S. National science foundation sponsored workshop. *Int. J. Hum. Comput. Interact.* 2006, *20*, 61−77.
12. Stuerzlinger W., Wingrave C. A. The value of constraints for 3D user interfaces. In *Virtual Realities: Dagstuhl Seminar 2008*; Springer: Vienna, 2008, pp. 203−223.
13. Guinness D., Jude A., Poor G. M., Dover A. Models for rested touchless gestural interaction. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction (Los Angeles, California, USA) (SUI '15)*; ACM: New York, NY, USA, 2015, pp. 34−43.
14. Arora R., Habib Kazi R., Grossman T., Fitzmaurice G., Singh K. SymbiosisSketch: combining 2D & 3D sketching for designing detailed 3D objects in situ. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*; Association for Computing Machinery: Montreal, QC, Canada, 2018, pp. 1−15.
15. Bornik A., Beichel R., Kruijff E., Reitinger B., Schmalstieg D. A hybrid user interface for manipulation of volumetric medical data. In *Symposium on 3D User Interfaces (3DUI)*; IEEE: Alexandria, VI, USA, 2006, pp. 29−36.
16. Graf H., Wundrak S., Stork A. The hybrid desktop within engineering environments. In *International Conference on Human-Computer Interaction (HCI International)*; Erlbaum: Las Vegas, Nev., 2005. CD−ROM.
17. Vinayak, Ramani K., Lee K. Jr., Jasti R. zPots: a virtual pottery experience with spatial interactions using the Leap motion device. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (Toronto, Ontario, Canada) (CHI EA '14)*; ACM: New York, NY, USA, 2014, pp. 371−374.
18. Osawa N. Enhanced hand manipulation methods for efficient and precise positioning and release of virtual objects. *IEICE Trans. Info Syst.* 2008, *E91-D*, 2503−2513.
19. Quek F., McNeill D., Bryll R., Duncan S., Ma X.-F., Kirbas C., McCullough K. E., Ansari R. Multimodal human discourse: gesture and speech. *ACM Trans. Comput. Hum. Interact.* 2002, *9*, 171−193.
20. Petruschat J. Essentials. In *Form+zweck, How to Handle Hands?* Vol. 18, 2001, pp. 24−43.
21. Arora R., Habib Kazi R., Anderson F., Grossman T., Singh K., Fitzmaurice G. Experimental evaluation of sketching on surfaces in VR. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*; ACM Press: New York, USA, 2017, pp. 5643−5654.
22. Wang Y., MacKenzie C. L. The role of contextual haptic and visual constraints on object manipulation in virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (The Hague, The Netherlands) (CHI '00)*; ACM: New York, NY, USA, 2000, pp. 532−539.
23. Hoffman H. G. Physically touching virtual objects using tactile augmentation enhances the realism of virtual environments. In *Proceedings. IEEE 1998 Virtual Reality Annual International Symposium (Cat. No.98CB36180)*, 1998, pp. 59−63.
24. Jung T. Controlling operation of a 3D tracking device WO2017080579 (A1), 2017. https://patents.google.com/patent/WO2017080579A1/en.
25. Jung T., Bauer P. 3D touch-and-drag: gesture-free 3D manipulation with finger tracking. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2018, pp. 589−590.
26. Clark J. H. Designing surfaces in 3-D. *Commun. ACM* 1976, *19*, 454−460.
27. Butterworth J., Davidson A., Hench S., Olano M. T. 3DM: a three dimensional modeler using a head-mounted display. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics (Cambridge, Massachusetts, USA) (I3D '92)*; ACM: New York, NY, USA, 1992, pp. 135−138.
28. Bowman D. A., Wineman J., Hodges L. F., Allison D. Designing animal habitats within an immersive VE. *IEEE Comput. Graph. Appl.* 1998, *18*, 9−13.
29. Zhong Y., Müller-Wittig W. Solid modelling through constraint-based manipulations in a virtual reality environment. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*; *Volume 1: 22nd Computers and Information in Engineering Conference*, 2002, pp. 21−28.
30. Piekarski W., Thomas B. H. Interactive augmented reality techniques for construction at a distance of 3D geometry. In *Proceedings of the Workshop on Virtual Environments 2003 (Zurich, Switzerland) (EGVE '03)*; ACM: New York, NY, USA, 2003, pp. 19−28.
31. Shen Y., Ong S. K., Nee A. Y. C. Collaborative design in 3D space. In *Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry (Singapore) (VRCAI '08)*; ACM: New York, NY, USA, 2008, p. 6. Article 29.
32. Keefe D., Zeleznik R., Laidlaw D. Drawing on air: input techniques for controlled 3D line illustration. *IEEE Trans. Visual. Comput. Graph.* 2007, *13*, 1067−1081.
33. Wacker P., Wagner A., Voelker S., Borchers J. Physical guides: an analysis of 3D sketching performance on physical objects in augmented reality. In *Proceedings of the Symposium on Spatial User Interaction (Berlin, Germany) (SUI '17)*; Association for Computing Machinery: New York, NY, USA, 2018, pp. 25−35.
34. Nishino H., Utsumiya K., Korida K. 3D object modeling using spatial and pictographic gestures. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (Taipei, Taiwan) (VRST '98)*; ACM: New York, NY, USA, 1998, pp. 51−58.
35. Matsumiya M., Takemura H., Yokoya N. An immersive modeling system for 3D free-form design using implicit surfaces. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (Seoul, Korea) (VRST '00)*; ACM: New York, NY, USA, 2000, pp. 67−74.
36. Kim H., Albuquerque G., Havemann S., Fellner D. W. Tangible 3D: hand gesture interaction for immersive 3D modeling. In *Proceedings of the 11th Eurographics Conference on Virtual*

Environments (Aalborg, Denmark) (EGVE'05); Eurographics Association, Aire-la-Ville: Switzerland, Switzerland, 2005, pp. 191–199.

37. Dave D., Chowriappa A., Kesavadas T. Gesture interface for 3D CAD modeling using Kinect. *Comput. Aided Des. Appl.* 2013, *10*, 663–669.
38. Jang S.-A., Kim H.-i., Woo W., Wakefield G. *AiRSculpt: A Wearable Augmented Reality 3D Sculpting System*; Springer International Publishing: Cham, 2014; pp. 130–141.
39. Jang S.-A., Wakefield G., Lee S.-H. Incorporating kinesthetic creativity and gestural play into immersive modeling. In *Proceedings of the 4th International Conference on Movement Computing (London, United Kingdom) (MOCO '17)*; ACM: New York, NY, USA, 2017, p. 8. Article 17 http://doi.acm.org/10.1145/3077981.3078045.
40. Weichel C., Lau M., Kim D., Villar N., Gellersen H. W. MixFab: a mixed-reality environment for personal fabrication. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems (Toronto, Ontario, Canada) (CHI '14)*; ACM: New York, NY, USA, 2014, pp. 3855–3864.
41. Schkolne S., Pruett M., Schröder P. Surface drawing: creating organic 3D shapes with the hand and tangible tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Seattle, Washington, USA) (CHI '01)*; ACM: New York, NY, USA, 2001, pp. 261–268.
42. Zhang F., Liu Z., Cheng Z., Deussen O., Chen B., Wang Y. Mid-air finger sketching for tree modeling. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, 2021, pp. 826–834.
43. Groenewald C., Anslow C., Islam J., Rooney C., Passmore P., Wong W. Understanding 3D mid-air hand gestures with interactive surfaces and displays: a systematic literature review. In *Proceedings of the 30th International BCS Human Computer Interaction Conference: Fusion! (Poole, United Kingdom) (HCI '16)*; BCS Learning & Development Ltd.: Swindon, UK, 2016, p. 13. Article 43.
44. Brown M. A., Stuerzlinger W., Mendonça Filho E. J. The performance of un-instrumented in-air pointing. In *Proceedings of Graphics Interface 2014 (Montreal, Quebec, Canada) (GI '14)*; Canadian Information Processing Society: Toronto, Ont., Canada, 2014, pp. 59–66. http://dl.acm.org/citation.cfm?id=2619648.2619659.
45. Vuibert V., Stuerzlinger W., Cooperstock J. R. Evaluation of docking task performance using mid-air interaction techniques. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction (Los Angeles, California, USA) (SUI '15)*; ACM: New York, NY, USA, 2015, pp. 44–52.
46. Jude A., Poor G. M., Guinness D. Grasp, grab or pinch? Identifying user preference for in-air gestural manipulation. In *Proceedings of the 2016 Symposium on Spatial User Interaction (Tokyo, Japan) (SUI '16)*; ACM: New York, NY, USA, 2016, p. 219.
47. Dudley J. J., Schuff H., Kristensson P. O. Bare-handed 3D drawing in augmented reality. In *Proceedings of the 2018 Designing Interactive Systems Conference (Hong Kong, China) (DIS '18)*; ACM: New York, NY, USA, 2018, pp. 241–252.
48. Ricca A., Chellali A., Otrnane S. The influence of hand visualization in tool-based motor-skills training, a longitudinal study. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, 2021, pp. 103–112.
49. Coninx K., Van Reeth F., Fleracker E. A hybrid 2D/3D user interface for immersive object modeling. In *Proceedings Computer Graphics International*, 1997, pp. 47–55.
50. Mine M., Yoganandan A., Coffey D. Making VR work: building a real-world immersive modeling application in the virtual world. In *Proceedings of the 2Nd ACM Symposium on Spatial User Interaction (Honolulu, Hawaii, USA) (SUI '14)*; ACM: New York, NY, USA, 2014, pp. 80–89.
51. Microsoft. *Hololens*, 2017. https://www.microsoft.com/en-us/hololens (accessed May 25, 2017).
52. Buxton W. A three-state model of graphical input. In *Proceedings of the IFIP TC13 Third Interational Conference on Human-Computer Interaction (INTERACT '90)*; North-Holland Publishing Co.: Amsterdam, The Netherlands, The Netherlands, 1990, pp. 449–456. http://dl.acm.org/citation.cfm?id=647402.725582.
53. Bhaskar Surale H., Matulic F., Vogel D. Experimental analysis of barehand mid-air mode-switching techniques in virtual reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (Glasgow, Scotland Uk) (CHI '19)*; ACM: New York, NY, USA, 2019, p. 14. Article 196.
54. Jung T. Three-dimensional modeling interface for augmented realities. In *Proceedings of Kultur und Informatik: Augmented Reality*; vwH Verlag: Berlin, Germany, 2016, pp. 79–87. http://www.artist3d.de.
55. HTC. *VIVE*, 2017. https://www.vive.com (accessed May 25, 2017).
56. Guna J., Jakus G., Pogacnik M., Tomazic S., Sodnik J. An analysis of the precision and reliability of the Leap motion sensor and its suitability for static and dynamic tracking. *Sensors* 2014, *14*, 3702–3720.
57. Fitts P. M. The information capacity of the human motor system in controlling the amplitude of movement. *J. Exp. Psychol.* 1954, *4*, 381–391.
58. Slater M., Usoh M., Steed A. Depth of presence in virtual environments. *Presence Teleoperators Virtual Environ.* 1994, *3*, 130–144.
59. Moerman C., Marchal D., Grisoni L. Drag'n Go: simple and fast navigation in virtual environment. In *2012 IEEE Symposium on 3D User Interfaces (3DUI)*, 2012, pp. 15–18.
60. Murata A., Iwase H. Extending Fitts' law to a three-dimensional pointing task. *Hum. Mov. Sci.* 2001, *20*, 791–805.
61. Ware C., Balakrishnan R. Reaching for objects in VR displays: lag and frame rate. *ACM Trans. Comput.-Hum. Interact.* 1994, *1*, 331–356.
62. Scott MacKenzie I., Sellen A., Buxton W. A. S. A comparison of input devices in element pointing and dragging tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (New Orleans, Louisiana, USA) (CHI '91)*; ACM: New York, NY, USA, 1991, pp. 161–166.
63. Accot J., Zhai S. Beyond Fitts' law: models for trajectory-based HCI tasks. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (Atlanta, Georgia, USA) (CHI '97)*; ACM: New York, NY, USA, 1997, pp. 295–302.
64. Hart S. G., Staveland L. E. Development of NASA-TLX (task Load index): results of empirical and theoretical research. In *Human Mental Workload*; Hancock P. A., Meshkati N., Eds.; *Advances in Psychology*; North-Holland, Vol. 52, 1988, pp. 139–183. http://www.sciencedirect.com/science/article/pii/S0166411508623869.
65. Mercado V., Marchai M., Lécuyer A. Design and evaluation of interaction techniques dedicated to integrate Encountered-type haptic displays in virtual environments. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2020, pp. 230–238.

Bionotes

Thomas Jung
HTW Berlin, University of Applied Sciences,
Berlin, Germany
Thomas.Jung@HTW-Berlin.de
Thomas.Jung@HTW-Berlin.de

Thomas Jung is professor of computer Science at HTW Berlin. His research focuses on computer graphics, mixed reality and human-computer interaction, especially in the field of 3D interaction in the context of geometric modeling. He is currently working in a project on participatory urban development (BMBF Inspirer).

Johann Habakuk Israel
HTW Berlin, University of Applied Sciences,
Berlin, Germany
JohannHabakuk.Israel@HTW-Berlin.de

Johann Habakuk Israel is a Professor of Applied Computer Sciences at the HTW Berlin. His main research areas are immersive sketching and modelling, tangible interaction, human-computer interaction, and virtual reality. He is currently involved in projects on participatory urban development (BMBF Inspirer), body schema therapy using VR (BMBF ViTraS) and virtual paleantology (Cluster of Excellence “Matters of Activity”).

Patrick Bauer
HTW Berlin, University of Applied Sciences,
Berlin, Germany
patrick.jean.bauer@googlemail.com

After his studies of Applied Computer Science at HTW Berlin, M.Sc. Patrick Bauer started his career as an agile software developer at Mayflower GmbH, where he mainly works on website architecture. His personal and technological interests focus on virtual and augmented reality systems as well as human-computer interaction using modern input devices.

Ruben Ahlhelm
HTW Berlin, University of Applied Sciences,
Berlin, Germany
rahlhelm@live.de

Ruben Ahlhelm has a Master of Science in Applied Computer Science from HTW Berlin. During his studies he focused on mobile applications and 3D modeling in a virtual environment. He currently works as an Android developer at agricultural AI company PEAT.