

Tobias Bernard

# *Design and Evaluation of Spatial Interfaces in Virtual Reality*

Master thesis

Supervisor: Prof. Dr. Marc Alexa

Department of Computer Graphics  
TU Berlin

Copyright © 2018 Tobias Bernard

This thesis uses the Tufte-Latex template (<http://tufte-latex.googlecode.com>) licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

*Declaration*

I hereby declare that I have completed this thesis independently without any unauthorized external help using only the cited sources and references.

*Deklaration*

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

*Berlin, March 2018*

---





# *Contents*

*Abstract*      7

*Zusammenfassung*      9

*Introduction*      11

*Related Work*      13

*Background*      23

*Study*      27

*Results*      35

*Discussion*      45

*Conclusion*      47

*Bibliography*      49



# *Abstract*

In order to display more information than the screen can accommodate at any one time, graphical user interfaces often clip the content, *i.e.* they employ a virtual space larger than the screen, which users can interactively navigate. Because users can only see a small part of the clipped data space at any given time, it is harder for them to establish a mental model of the data. This can make interfaces difficult to navigate, especially with large amounts of content.

Many Virtual Reality applications today employ the same clipping patterns as traditional 2D interfaces. Our hypothesis is that by using all the possibilities afforded by the 3D environment (such as the user's ability to move and turn their head), VR applications can be made easier to navigate than is the case with clipping-based approaches.

To test this hypothesis we ran a study in which we compared three different interface types: *Spatial* (a grid), *Stacked* (a three-dimensional scrolling list), and *Clipped* (a clipped scrolling list). The content was sets of 20, 50, or 150 square cards with simple icons. We tested all conditions with *Monochrome* and colorful card backgrounds (10 different randomly assigned colors). For each of the 18 resulting conditions, participants were instructed to find a series of icons as quickly as possible.

Participants found the target icons significantly faster in *Spatial* conditions compared to *Stacked* and *Clipped*. Colorful backgrounds also improved the task performance significantly. We did not find a quantitative difference between *Stacked* and *Clipped*, but in our post-experiment questionnaire users expressed a significant preference for *Stacked*.



# Zusammenfassung

Zur Darstellung von großen Informationsmengen, welche nicht gleichzeitig auf den Bildschirm passen, wird in traditionellen grafischen Oberflächen häufig “clipping” (ausschneiden, beschneiden) verwendet. Dabei wird der Inhalt auf einer Fläche dargestellt, welche größer als der Bildschirm ist. Der sichtbare Ausschnitt des Bildes kann von Nutzern interaktiv verschoben werden. Da immer ein Teil des Inhalts sichtbar ist, fällt es Nutzern schwer, sich ein mentales Modell der Daten zu bilden, was die Navigation schwieriger machen kann.

Viele Virtual Reality Anwendungen verwenden dieselben “clipping” Patterns wie traditionelle 2D Oberflächen. Unsere Hypothese ist, dass durch die Nutzung des vollen Potentials der dritten Dimension (z.B. die Möglichkeit, sich im Raum zu bewegen) bessere Interaktionsmodelle für die Navigation von VR Umgebungen möglich sind.

Um diese Hypothese zu testen haben wir drei verschiedene Oberflächen in einer Studie verglichen: *Spatial* (ein Raster von Elementen), *Stacked* (eine dreidimensionale scrollbare Liste), und *Clipped* (eine “geclippte” scrollbare Liste). Die Oberflächen bestanden aus 20, 50, oder 150 quadratischen Karten einem Piktogramm. Alle Konditionen wurden in einer einfarbigen Variante und mit buntem Hintergrund der Karten (10 verschiedene zufällig zugewiesene Farben) getestet. Für jede der insgesamt 18 Konditionen bestand die Aufgabe der Teilnehmer darin, eine Reihe von Piktogrammen so schnell wie möglich zu finden.

Die Teilnehmer waren signifikant schneller in *Spatial* Konditionen im Vergleich zu *Stacked* und *Clipped*. Farbige Hintergründe hatten ebenfalls einen signifikanten positiven Effekt. Wir konnten keinen quantitativen Unterschied zwischen *Stacked* und *Clipped* feststellen, aber in einem Fragebogen nach dem Experiment drückten Teilnehmer eine signifikante Präferenz für *Stacked* aus.



# Introduction

When presenting large amounts of information on any medium, whether it's a book, a paper map, or a phone's screen, there are always cases where it isn't possible to show everything at the same time. Strategies for handling this problem are as old as civilization: Ancient scrolls employ a single, linear surface which is rolled up, books split the up information and lay it out on pages, which can be stacked, and paper maps are be folded in intricate ways <sup>1</sup>, in order to enable easy navigation in two dimensions (see figure 1).

In screen-based graphical user interfaces, these cases are usually handled by relying on clipping, *i.e.* by employing a virtual space on which all of the information is laid out, and cutting off the content at the edges of the screen. Users can then interactively navigate the virtual space by moving their viewport, *e.g.* by scrolling. This technique works, but it is sub-optimal, because it puts the burden of keeping track of the position in the virtual space on the user. Instead of being able to see the entire space they have to rely on their mental model of it <sup>2</sup>.

Unlike with screens, there are no natural boundaries for Virtual Reality (VR) interfaces, because the virtual environment takes up the entire field of vision. Additionally, users can move around in this environment while using an application. This presents the opportunity to use the larger space, as well as the third dimension, in the design of user interfaces.

However, there are few examples of VR interfaces that make use of these possibilities. Most VR applications today are either games or novelty apps such as 3D drawing programs. The interfaces with some complexity that do exist tend to mimic 2D interface patterns and employ clipping (see figure 2).

In this study we explored spatial interfaces which enable the presentation and navigation of large data sets using the unique possibilities that 3D space affords. While the limited real estate on screens necessitates clipping in most cases, this is not true for VR. The

<sup>1</sup> Stephan Angsüsser. Map folding techniques in the digital age. 2012

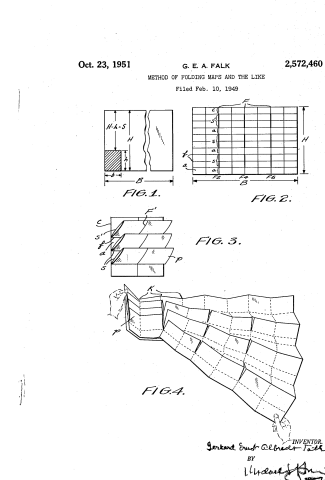


Figure 1: US Patent 2572460 "A United Method for Folding Maps and the Like", from 1951 by G. E. A. Falk, describes a technique for folding printed maps in such a way that they can be read without fully unfolding them.

<sup>2</sup> Benjamin B Bederson. The promise of zoomable user interfaces. *Behaviour & Information Technology*, 30(6):853–866, 2011



Figure 2: Application detail page in the Oculus Home store. Clicking the sections on the left changes the content of the center column with a fade animation.

potentially unlimited 3D space VR interface elements exist in allows us to keep them around, even when they aren't being used or looked at.

### *Research Question*

The primary research question is whether spatial interfaces can improve the usability of VR applications compared to clipping. Additionally, we explored the advantages and disadvantages of different spatial approaches and interaction modalities in VR.

To answer our question we tested three different interfaces displaying the same information, two of which were spatial, while one employs clipping. We measured the performance of these different interfaces in a quantitative experiment, and assessed the user experience using questionnaires.

### *Contributions*

We evaluated general-purpose patterns for displaying and navigating large amounts of information in VR, which can be used by others building information-dense VR applications in the future.



## Related Work

This chapter aims to provide an overview of the history and state of the art in VR research, the most important challenges with designing VR interfaces, and the various interaction techniques that have been developed to overcome them. It also surveys the use of 3D space in the interfaces of current-generation consumer VR systems.

### Virtual Reality

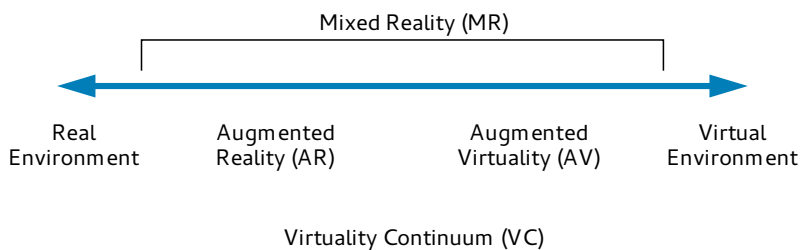
Both *Augmented Reality* and *Virtual Reality* use screens (or other display technologies) and positional tracking of the user's body in order to simulate virtual, spatial objects before the user's eyes. Azuma's *Survey of Augmented Reality* (1997) describes Augmented Reality as "3-D virtual objects are integrated into a 3-D real environment in real time"<sup>3</sup>. Virtual Reality, on the other hand, tends to be used to describe systems with few or no real-world objects, where users are fully immersed in an experience.

Milgram's reality-virtuality continuum spans from augmented reality (mostly real objects with some virtual ones), to virtual reality (mostly virtual elements)<sup>4</sup>. Between these two extremes there are many different types of *mixed reality* with different degrees of virtuality.

<sup>3</sup> Ronald T Azuma. A survey of augmented reality. *Presence: Teleoperators and virtual environments*, 6(4):355–385, 1997

<sup>4</sup> Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12): 1321–1329, 1994

Figure 3: A simplified representation of Milgram's reality-virtuality continuum.



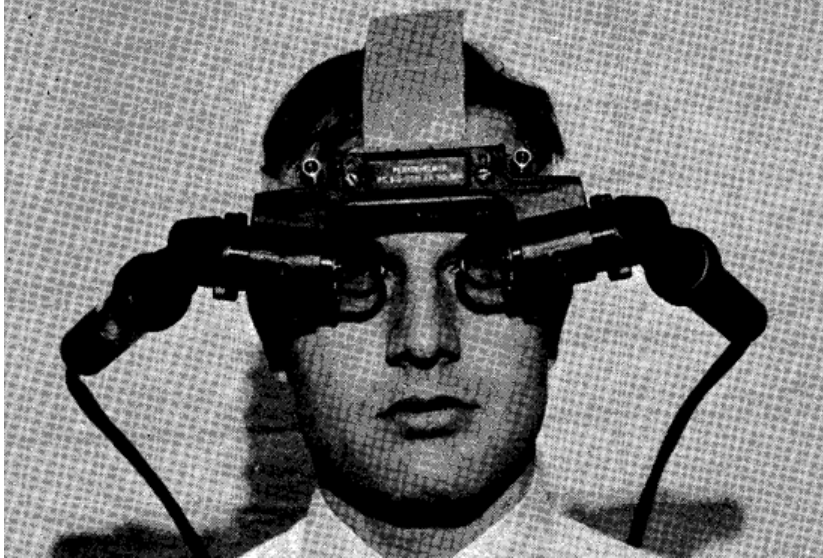


Figure 4: Ivan Sutherland's 1968 head-mounted display with miniature CRTs.

The earliest experiments with virtual and augmented reality date back as far as the 1960s <sup>5</sup>, but only starting in the late 1980s head-mounted displays became practical and widely available <sup>6</sup>. Though the technology was limited (both in terms of the display technology, and the graphics hardware used for rendering the virtual environments), many of the interaction techniques still used in current-generation VR were developed on these headsets.

Figure 5 shows the Forte VFX1, a head-mounted display released in 1995 <sup>7</sup>. It came with headphones and a handheld controller. The resolution was  $263 \times 230$  pixels per eye, and the field of view 35.5 degrees horizontally and 26.4 degrees vertically.

<sup>5</sup> Ivan E Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 757–764. ACM, 1968

<sup>6</sup> Mark Billinghurst, Adrian Clark, Gun Lee, et al. A survey of augmented reality. *Foundations and Trends® in Human-Computer Interaction*, 8(2-3): 73–272, 2015

<sup>7</sup> VR Wiki - Forte VFX1. <http://vrwiki.wikispaces.com/Forte+VFX1>. Accessed: 2017-11-10



Figure 5: The Forte VFX1, an early head-mounted display for virtual reality.

The 1995 paper *User interface constraints for immersive virtual environment applications* gives an overview of the interface design constraints and challenges when building VR interfaces<sup>8</sup>. It stresses the importance of affordances (and signifiers), mappings, feedback, and constraints to designing usable interfaces<sup>9</sup>, especially in the context of immersive VR software. It mentions spatial boundaries that keep users from leaving the physical space designated for the VR experience, raycasting for object selection, and a number of other techniques that are still highly relevant in 2017.

### *Input Devices*

When it comes to interacting with VR objects, the fundamental problem is that the “natural” way to interact with 3D objects in the real world using our hands does not work. VR headsets simulate the visual appearance of virtual objects, but there is no equally flexible technology for simulating haptics<sup>10</sup>, so VR objects cannot be touched or manipulated like real-world objects. Different VR input technologies have different sets of trade-offs, but are ultimately all limited in this respect.

The simplest interaction mechanic, mostly employed by mobile VR systems on phones, is gaze-based. To trigger an action, the user has to look at an object and either wait a certain amount of time, visualized by a “fuse” timer, or press a button on a remote to confirm the action. This mechanic is of course very limited and only useful for simple experiences like media consumption. Some VR experiences can be used with traditional game controllers, which provide more degrees of freedom and better haptic feedback, but are also very limiting because they can only be used for indirect manipulation. They also require remembering the button layout, because the controller is not visible from within VR.



<sup>8</sup> Douglas A Bowman and Larry F Hodges. User interface constraints for immersive virtual environment applications. Technical report, Georgia Institute of Technology, 1995

<sup>9</sup> Don Norman. *The design of everyday things: Revised and expanded edition*. Basic Books (AZ), 2013

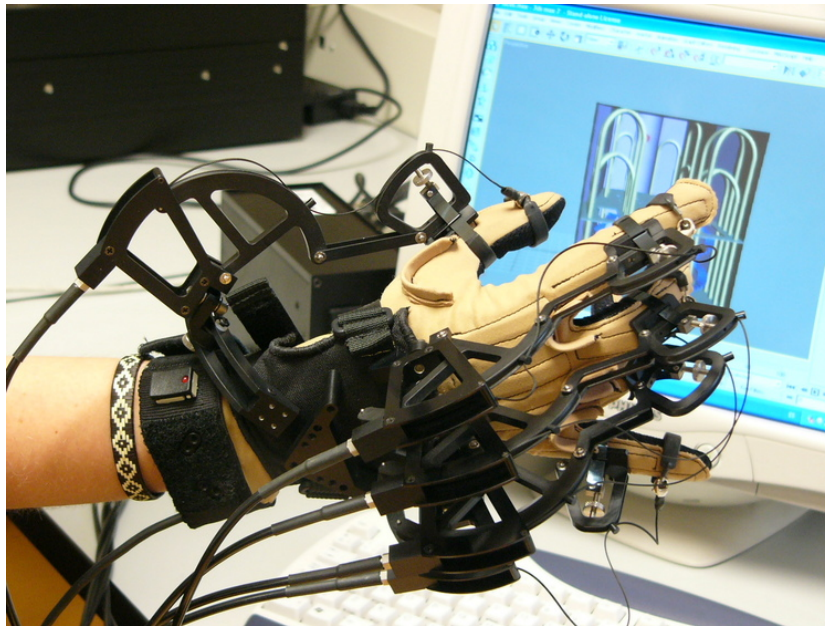
<sup>10</sup> Grigore C Burdea. Keynote address: haptics feedback for virtual reality. In *Proceedings of international workshop on virtual prototyping*. Laval, France, pages 87–96, 1999

Figure 6: How an HTC Vive controller is held, with the index finger on the trigger button (seen from the side and the top).

VR systems like the Oculus Rift or HTC Vive come with two controllers, one for each hand. They are tracked in 3D space, and are therefore visible in VR. Their shape and button layouts allow for some

natural hand movements, like grabbing or pointing to be detected. Figure 6 shows an HTC Vive controller. However, the current generation of controllers does not detect each finger separately, so this is limited to a few gestures. The spatial tracking of the controllers allows for some degree of direct manipulation of objects, e.g. moving the hand onto an object, grabbing it, and then throwing it across the room. There are limitations to this however, since there is no real haptic feedback, and objects offer no resistance to the user intersecting them with their bodies or controllers <sup>11</sup>.

Other systems, such as the Microsoft HoloLens, employ a gesture-based interaction model using hand tracking. This works by visually detecting hand positions and gestures using the cameras on the headset. It is also possible to add hand tracking to desktop VR systems such as the HTC Vive by adding a Leap Motion hand tracking system to it <sup>12</sup>. These systems have the advantage of not requiring additional hardware and allowing for more degrees of freedom. However, they lack any kind of haptic feedback and the gesture recognition can be unreliable, which is frustrating for users.



<sup>11</sup> Maria V Sanchez-Vives and Mel Slater. From presence to consciousness through virtual reality. *Nature Reviews Neuroscience*, 6(4):332–339, 2005

<sup>12</sup> Peter Wozniak, Oliver Vauderwange, Avikarsha Mandal, Nicolas Javahiraly, and Dan Curticapean. Possible applications of the leap motion controller for more interactive simulated experiments in augmented or virtual reality. In *Optics Education and Outreach IV*, volume 9946, page 99460P. International Society for Optics and Photonics, 2016

Figure 7: The CyberGrasp, consisting of the CyberGlove, a fabric glove containing sensors for the hand position, and an exoskeleton with cable-driven actuators to provide haptic feedback.

Another possible type of input device is using gloves with sensors to track hand movements and gestures. This technology is not used by any current-generation VR system, but there have been several commercially available devices implementing this concept. Some of these devices did not actually track the motion of the fingers, but merely registered when two fingertips touched, and could therefore



only recognize pinching gestures<sup>13</sup>. More advanced solutions like the CyberGlove are able to precisely track hand movements movements, including the bend on each individual finger. This system can be extended to provide haptic feedback using mechanical actuators. However, this requires more than just a glove, but an exoskeleton around the user's hand<sup>14</sup>. It is therefore not very practical for general-purpose systems, and mostly used for specialized industrial or military applications.

### Locomotion

The simplest way to navigate VR experiences is tracking body movement and mapping it 1-1 to the virtual space. Current-generation room scale VR is capable of doing this reasonably well. Thanks to a 90 Hz display refresh rate (in both Oculus and Vive) there is no perceivable lag from head movement in most cases. However, this type of locomotion limits the size virtual spaces to the size of the physical space (usually around  $3 \times 3$  meters for current-generation room scale VR systems). This means that the virtual spaces can not have arbitrary proportions, which is a problem for many kinds of VR experiences, such as open-world games, or architectural simulations. Thus, more flexible, but less intuitive types of locomotion are required in order to enable these kinds of experiences inside limited physical spaces.

One solution to this problem is "flying" around the virtual space by simply moving the camera without the user moving their body. The movement can be indirectly controlled e.g. by pressing buttons on a controller<sup>15</sup>. This is a very flexible locomotion mechanic that scales to any kind of room, but is problematic because it can cause motion sickness. Since the user is not actually moving, it results in conflicting visual and vestibular stimuli, leading to disorientation and nausea for many people<sup>16</sup>.

Another common solution is teleportation. In most implementations of this mechanic, the user can set their new position by pointing at a location somewhere in the virtual space<sup>17</sup>. They are then instantly "transported" to the new location, usually with little or no animations. This can be disorienting, as the user has to re-scan the space after the transitions, but since there is no movement between the start and end positions, it does not cause motion sickness. The feedforward of explicitly setting the new position in space is likely another factor that mitigates this. Figure 8 shows an example of what this looks like.

<sup>13</sup> Doug Bowman, Chadwick Wingrave, Joshua Campbell, and Vinh Ly. Using pinch gloves (TM) for both natural and abstract interaction techniques in virtual environments. 2001

<sup>14</sup> Yoseph Bar-Cohen. Haptic devices for virtual reality, telepresence, and human-assistive robotics. *Biol Inspired Intell Robots*, 122:73, 2003

<sup>15</sup> Douglas A Bowman, David Koller, and Larry F Hodges. Evaluation of movement control techniques for immersive virtual environments. Technical report, Georgia Institute of Technology, 1996

<sup>16</sup> Hironori Akiduki, Suetaka Nishiike, Hiroshi Watanabe, Katsunori Matsuoka, Takeshi Kubo, and Noriaki Takeda. Visual-vestibular conflict induced by virtual reality in humans. *Neuroscience letters*, 340(3):197–200, 2003

<sup>17</sup> Evren Bozgeyikli, Andrew Raij, Srinivas Katkooari, and Rajiv Dubey. Point & teleport locomotion technique for virtual reality. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, pages 205–216. ACM, 2016

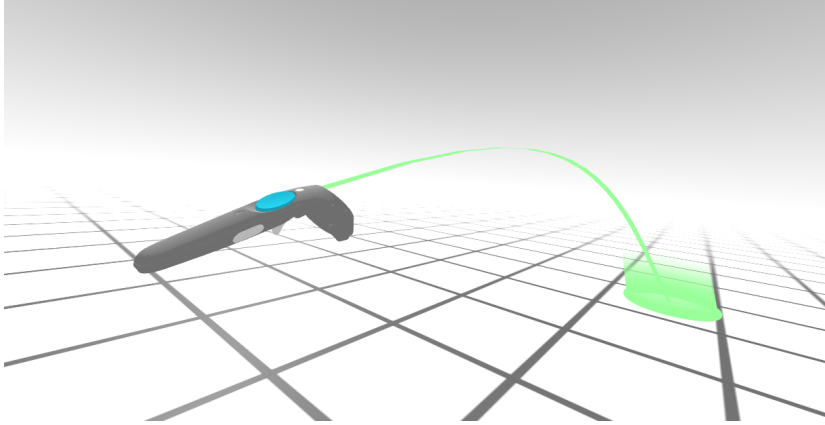


Figure 8: The teleportation mechanic, as implemented by A-Frame's teleport component. The green circle on the floor previews where the user will be teleported.

“Redirected Walking” is a technique that imperceptibly rotates the virtual environment to accommodate a larger virtual space<sup>18</sup>. This can be used to create the illusion of walking in a straight line, when in reality walking along an arc (see figure 9). However, since the rotation has to be slow in order for users not to notice it, this technique requires a large tracked space in order to accommodate arbitrary virtual environments.

<sup>18</sup> Sharif Razzaque, Zachariah Kohn, and Mary C Whitton. Redirected walking. In *Proceedings of EUROGRAPHICS*, volume 9, pages 105–106. Manchester, UK, 2001

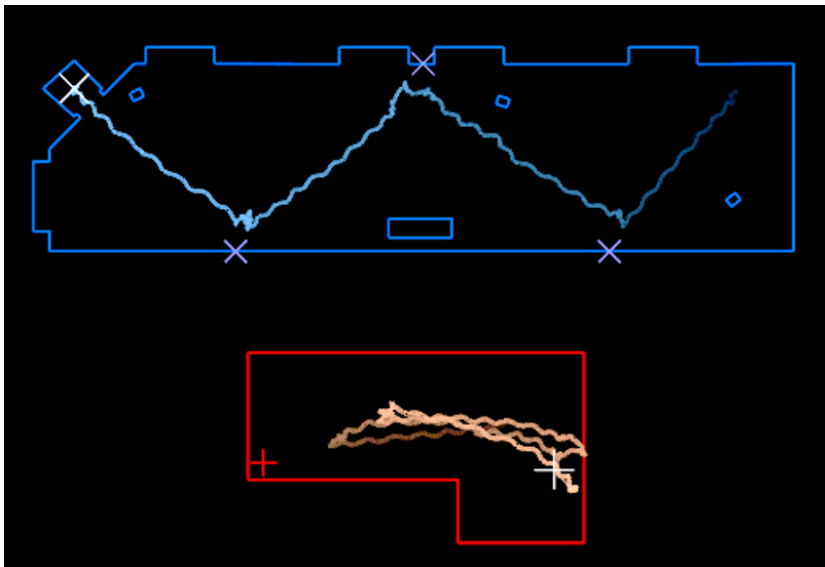


Figure 9: Walking paths in the virtual environment (above in blue) and the real room (below in red) from the study by Razzaque et al. When the user stands still, the environment slowly turns their view of the virtual environment, turning the zig-zag path in VR into a back-and-forth path in the real room.

Other approaches to this problem use specialized hardware to give users the illusion of physical travel. Omnidirectional treadmills allow users to walk in place without actually moving. This can be achieved using two perpendicular treadmills with a tracking system<sup>19</sup>, or a low-friction surface and special shoes<sup>20</sup>.

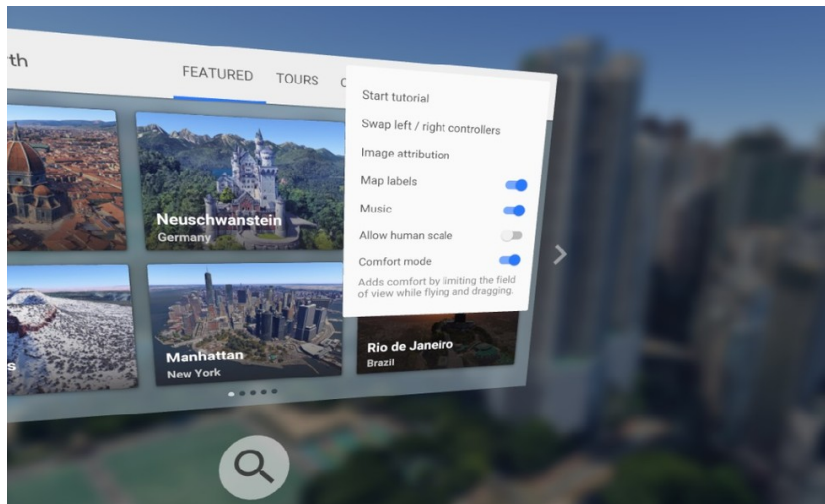
<sup>19</sup> Rudolph P Darken, William R Cockayne, and David Carmein. The omnidirectional treadmill: a locomotion device for virtual worlds. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 213–221. ACM, 1997

<sup>20</sup> Lawrence E Warren and Doug A Bowman. User experience with semi-natural locomotion techniques in virtual reality: the case of the virtuix omni. In *Proceedings of the 5th Symposium on Spatial User Interaction*, pages 163–163. ACM, 2017

### Selection & Manipulation

Basic direct manipulation of objects is possible with current-generation VR controllers, but sometimes indirect manipulation is necessary or preferable (e.g. because it is faster or more precise for a specific task). Bowman & Hodges describe a design metaphor for indirect manipulation interfaces in VR based on traditional desktop interface concepts such as widgets, menus, buttons, icons, and pointing <sup>21</sup>.

Most VR interfaces today employ patterns like these (see figure 10), which are translations of 2D interface elements to the 3D environment. Instead of a mouse pointer, raycasting cursors starting from a hand-tracked controller are often employed to select items. Clicks are emulated by button presses while an object is selected via raycasting. In this way, most traditional desktop interface elements, such as buttons, sliders, and switches can be used in 3D just as they would be in 2D.



<sup>21</sup> Douglas A Bowman and Larry F Hodges. Wimp (widgets, menus, and pointing) design tools for virtual environments. Technical report, Georgia Institute of Technology, 1994

Figure 10: A menu in Google Earth VR. The interface elements (tabs, menus, switches) are the exact same ones Google uses in their desktop and mobile products.

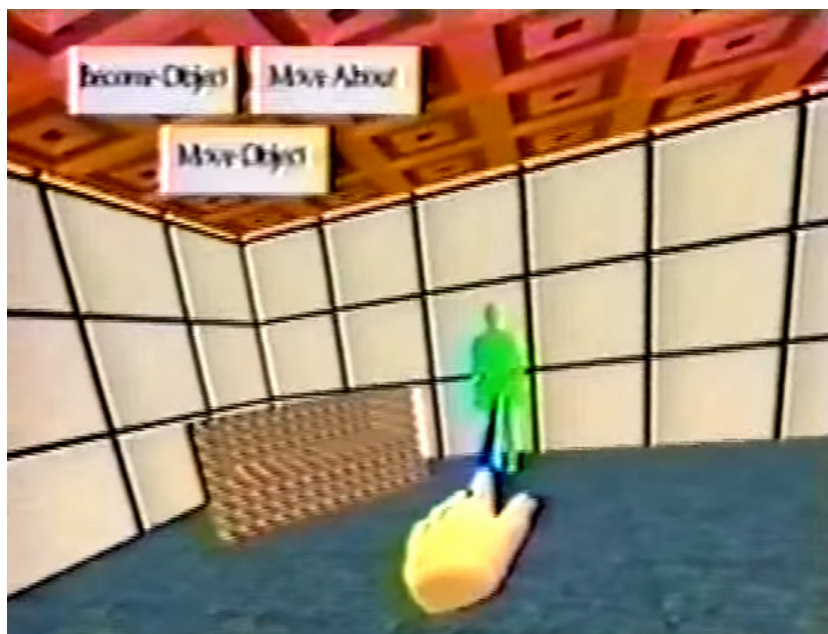
A different approach suggested by literature is using virtual “tools” to perform actions. Bowman defines these as “specialized objects, in some ways not part of the environment itself, but rather the representations of methods the user may employ to perform actions on or in the environment” <sup>22</sup>. Examples of virtual tools include guns or swords in games, paintbrushes in drawing applications, or minimaps of the environment to quickly teleport somewhere else. Wloka et al. present the Star Trek tricorder as a model for a multi-functional virtual tool, serving both as an input mechanism and a display <sup>23</sup>.

<sup>22</sup> Douglas A Bowman and Larry F Hodges. User interface constraints for immersive virtual environment applications. Technical report, Georgia Institute of Technology, 1995

<sup>23</sup> Matthias M Wloka and Eliot Greenfield. The virtual tricorder: a uniform interface for virtual reality. In *Proceedings of the 8th annual ACM symposium on User interface and software technology*, pages 39–40. ACM, 1995

Non-linear mapping of hand movement (also known as the “go-go” technique) has been proposed to make direct manipulation of far away objects less cumbersome <sup>24</sup>. Within a certain distance from the user, the virtual representation of the user’s hands is directly mapped to their real hands. After this threshold, the virtual hands are mapped in a non-linear fashion, allowing users to reach further than they otherwise could.

A 1997 study evaluated six different techniques for manipulating remote objects, including go-go and raycasting <sup>25</sup>. Based on the findings from this study the HOMER technique (Hand-centered Object Manipulation Extending Ray-casting) was developed, a hybrid between raycasting and direct manipulation. In this technique the user first selects their target object using ray-casting. Their virtual hand is then moved to the object, allowing them to directly manipulate it from a distance.



<sup>24</sup> Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in vr. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 79–80. ACM, 1996

<sup>25</sup> Doug A Bowman and Larry F Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 35–ff. ACM, 1997

Figure 11: The selection stage of Bowman’s HOMER technique. The element about to be selected is highlighted in green. After selection, the object can be moved in the environment using direct manipulation.

Traditional raycasting does not work the selection of objects that are partially or fully occluded. The Flexible Pointer technique tracks both hands to interactively construct a bezier-like curve, which can be used to reach around the objects obscuring the target <sup>26</sup>.

### *State of the Art VR Interfaces*

The current generation of VR Interfaces does not make much use of 3D space. Most interfaces are flat 2D panels, not very different

<sup>26</sup> Alex Olwal Steven Feiner. The flexible pointer: An interaction technique for selection in augmented and virtual reality. In *Proc. UIST’03*, pages 81–82, 2003



from traditional 2D interfaces in both form and behavior. They rely heavily on flat, clipped, scrollable areas, or simply pagination with fade transitions (see figure 12). The third dimension is often used for purely aesthetic purposes, e.g. as a three-dimensional wallpaper behind or around the interactive interface elements.

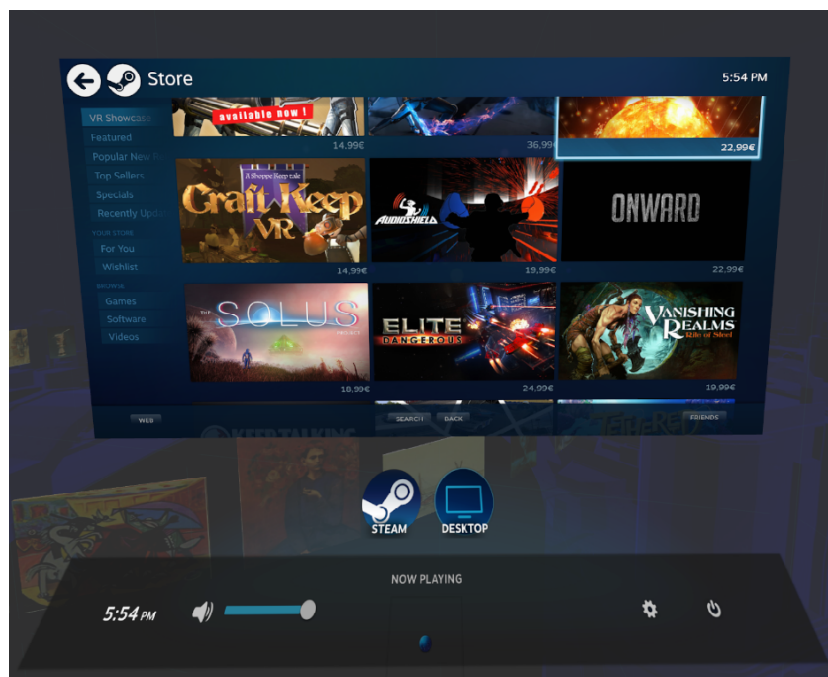


Figure 12: The SteamVR store interface is a single flat surface that behaves like a regular desktop app. On the bottom there is a taskbar to switch between open apps, and a system bar with time, volume and settings.

When interfaces do use 3D, it is often in a highly skeuomorphic context, mimicking real-world interactions such as picking up items and throwing them. This is at least in part due to the fact that the low resolution on the current generation of headsets makes interfaces high information densities impossible.

We found this to be true for both leading VR operating systems (Oculus Home and SteamVR), as well as most applications on these platforms. Oculus Home's interface makes some use of the 3D space by employing interface elements floating freely in space, while SteamVR's interface relies mostly on monolithic flat panels that behave exactly like a traditional screen. However, SteamVR has more consistent, spatial animations, while Oculus Home relies heavily on jarring cuts and fades.

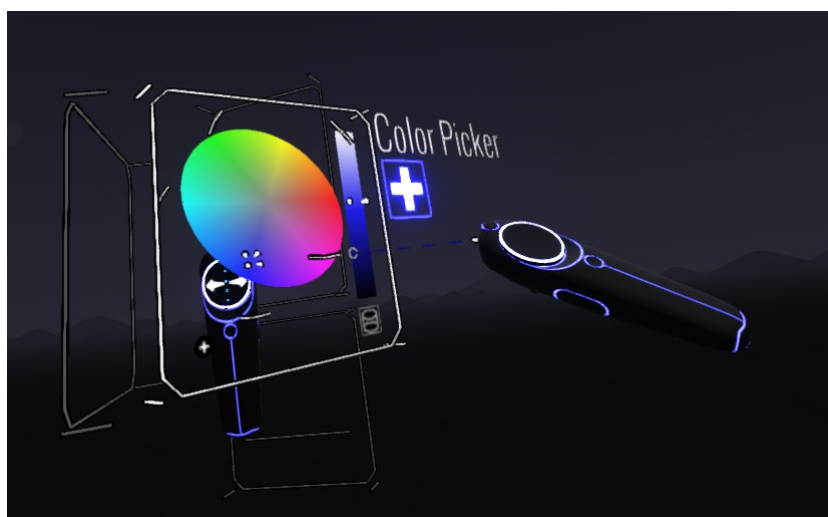
### *Controller-Mapped Interfaces*

One notable exception to the generally flat and traditional interfaces are the controls persistently mapped to the controllers in some

applications. Google Earth and TiltBrush are prominent examples of this pattern, which is a variation of Bowman's concept of virtual tools<sup>27</sup>.

In TiltBrush, the left hand has a persistent menu with three pages of settings, which are arrayed around the controller, facing outwards (see figure 13). They can be turned along the center axis with the left hand controller's touchpad, and the right hand controller is used to operate the menus. The contents of the two menu panels not facing the user is not visible, likely so as not to be visually distracting. As the panels turn, only the one facing the user shows The menus themselves are not as interesting as their containers, employing standard 2D interface patterns such as pagination and clipping.

In Google Earth, a miniature globe is fixed on top of the left hand controller (see figure 14). The right hand controller can be used to point at a position on the globe to teleport there. This kind of navigation with two hands feels very intuitive, because it only relies on direct hand movement in space, not abstract button mappings or external controls. This mechanic implements Stoakley's World In Miniature (WIM) concept<sup>28</sup>.



<sup>27</sup> Douglas A Bowman and Larry F Hodges. User interface constraints for immersive virtual environment applications. Technical report, Georgia Institute of Technology, 1995

<sup>28</sup> Richard Stoakley, Matthew J Conway, and Randy Pausch. Virtual reality on a wim: interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 265–272. ACM Press/Addison-Wesley Publishing Co., 1995

Figure 13: The color picker page on the TiltBrush menu. Only the page facing the user contains actual content, while the other two are only visible as wireframe outlines.

These examples show the potential of spatial interfaces in VR. They make use of the fact that users can see and move their hands, and move around the space. They make use of unique possibilities that 3D space offers in a functional way, to makes interfaces easier to navigate.



Figure 14: The miniature globe is fixed to the top of the left hand controller and increases in size when pointed at by the right hand controller. A little red pin shows the location hovered by the cursor. Pressing the trigger button teleports to that location.

# Background

## Technology

The VR ecosystem is currently (mid 2017) still in its infancy. We experimented with the Oculus Rift CV1 and the HTC Vive (both released in early 2016) and tested the headset and controller hardware, software ecosystem, and developer tools.

The installation and setup are very cumbersome for both headsets, especially on the software side, where we encountered many glitches, bugs, and usability problems.

The biggest issue with both headsets is the very low resolution of only  $1080 \times 1200$  pixels per eye. In addition, in the periphery of the display the picture is quite blurry. This means that refocusing the eyes to the sides to see content there is not possible, which is irritating. Text needs to be either very big, or very close in order to be legible, which makes any kind of information-dense application impractical<sup>29</sup>. Bowman's vision of Information-Rich Virtual Environments<sup>30</sup> is therefore still not feasible with current hardware.

Both the Rift and the Vive have room scale tracking that works reasonably well, but we found the Vive tracking to be more reliable. This is likely due to its trackers being mounted higher up and on opposite sides of the room. Walking around in the roughly 2x2m space feels natural, but the limitations of being able to move only within such a small space quickly become clear in scenarios like games with open-world environments, where it means that teleporting has to be the primary means of movement.

## Development

We developed our prototypes in WebVR using Mozilla's A-Frame<sup>31</sup> Framework. A-Frame is a Javascript library which enables building VR experiences using declarative components directly in HTML. Its entity-component structure makes it easy to prototype experiences by combining existing components, and defining new ones in Javascript. A-Frame runs completely standalone in the browser (though it does

<sup>29</sup> Anthony Elliott, Brian Peiris, and Chris Parnin. Virtual reality in software engineering: Affordances, applications, and challenges. In *Software Engineering (ICSE), 2015 IEEE/ACM 37th IEEE International Conference on*, volume 2, pages 547–550. IEEE, 2015

<sup>30</sup> Doug A Bowman, Chris North, Jian Chen, Nicholas F Polys, Pardha S Pyla, and Umur Yilmaz. Information-rich virtual environments: theory, tools, and research agenda. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 81–90. ACM, 2003

<sup>31</sup> A-Frame: A Javascript framework for building VR experiences on the web (<https://a-frame.io>)

require Steam or Oculus being set up on the computer in order to work with the Vive or Oculus headsets), and can be used with other Javascript libraries and standard web tooling. Though there are some performance and stability issues in some cases, the quality of the experiences is more or less on par with native VR for simple applications.

### *Prototypes*

In an initial exploration phase, we designed a number of different spatial interfaces and built prototypes to test their viability. The goal was to find novel ways of using three dimensions to make VR interfaces that don't require clipping to display large amounts of information.

#### *Stacked List Prototype*

This prototype is a scrollable 1-dimensional list of cards which uses z-depth to stack cards at both ends of a list (see figure 15). The content of these stacked cards is not visible, but their presence shows how many items there are above and below the currently visible cards, thereby intuitively communicating the position in the list to the user, without relying on external indicators such as scrollbars.

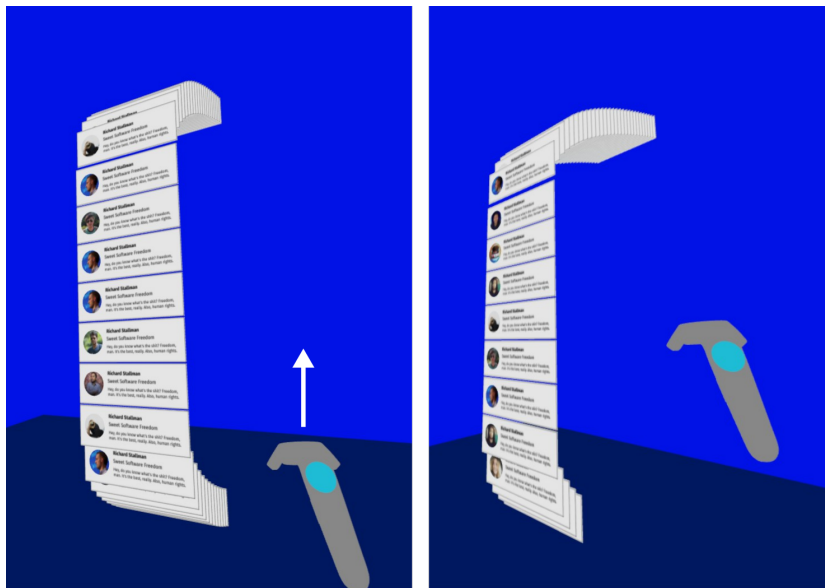


Figure 15: Illustration of the scrolling behavior: Lifting up the controller while pressing the trigger button moves the cards up in the list.

The list can be scrolled by keeping the trigger button on the controller pressed while moving it vertically. The scrolling direction

mimics the “natural” scrolling behavior on touchscreens and touchpads, whereby moving the controller up will move the elements upwards, thus scrolling down in the list.

The advantage of this approach is that it can accommodate many elements in a relatively small space. On the other hand, the usefulness of the stacked cards highly depends on the content. If the content is not very visually distinctive, individual cards are not recognizable while stacked.

This prototype was the inspiration for the *Stacked* interface type in our study.

### *Elevator Prototype*

In an attempt to use the available physical space as efficiently as possible, this prototype arrays information on a vertical 3D tube around the user (see figures 17 and 16). The information on the walls can be explored by walking around inside the tube, as well as changing the vertical position of the tube by interactively “scrolling” up and down, behaving as kind of elevator through the virtual space.

This prototype can accommodate very large amounts of information, and navigation is efficient and intuitive. However, being totally enveloped by the content can make orientation difficult due to the lack of stable landmarks. It also means that there is no possibility of getting a good overview by stepping back a bit. These problems could be addressed by only using having content along half of the tube. This could be combined with an overview mode, where the elevator moves back horizontally so users can see the tube from afar.

We found that most of the value in this kind of interface comes from being able to explore by simply looking around the space, rather than interacting. This is what inspired the *Spatial* interface type in our study.



Figure 16: The “elevator” prototype, seen from the outside



Figure 17: The “elevator” prototype showing paintings on the walls of the cylinder. The vertical position can be moved by pressing the trigger button and moving the controller vertically.



# Study

To evaluate the differences between spatial and clipping-based interfaces we conducted an empirical user study. We asked participants to perform the same task, finding icons in a list, using three different interfaces: A two-dimensional grid (*Spatial*), a scollable list that stacks overflowing elements in the z-dimension (*Stacked*), and a clipped, scrollable list (*Clipped*), illustrated in figure 18.

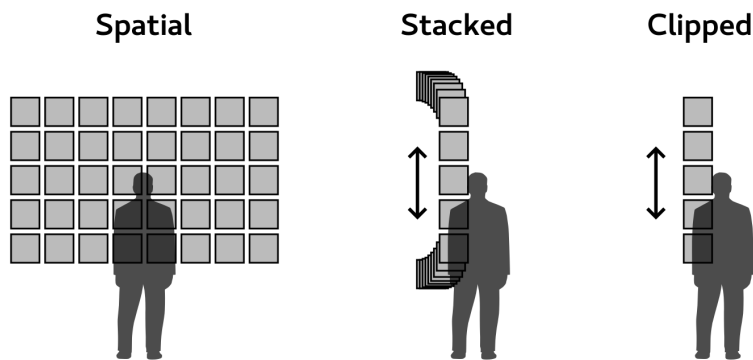


Figure 18: The three different experiment types. Each of these was used in 6 different configurations, with different numbers of items, *Colored* or *Monochrome*.

## Participants

We recruited 20 paid participants (8 female, 12 male), aged between 19 and 49 years (MDN = 24 years). Most of them were display workers, with an average of 5.9 hours per day working with computers (SD = 2.3 h). Most participants had some experience with Virtual Reality, but none of them had used it extensively.

## Apparatus

The study was conducted in a university lab room using an HTC Vive connected to a Windows 10 PC. Participants were instructed to stand in the middle of an area measuring about  $3m \times 3m$  at the center

of the room. They were given the Vive headset and one tracked hand controller. The room scale VR setup allowed them to freely move within this area and have their body and hand movement reflected faithfully inside VR (see figure 19).

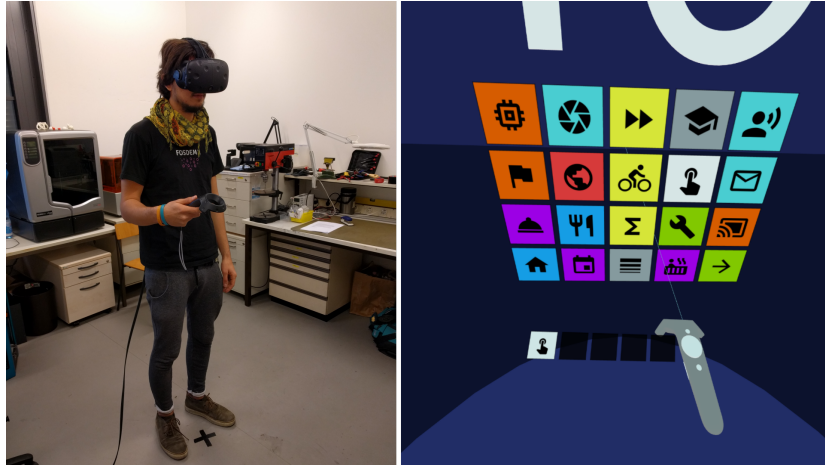


Figure 19: *Left*: Participant standing at the designated spot in the middle of the room, wearing the Vive Headset and holding the controller. *Right*: An experimental condition, as seen by the participant.

## Design

We used a within-subjects repeated measures design with three independent variables:

- *Interface type*: the three different interface types, *Spatial*, *Stacked*, and *Clipped*
- *Number of items*: 20, 50, or 150 icons
- *Colored*: Whether all icons have the same white background, or different, colored backgrounds (from a palette of 10 colors)

The resulting 18 conditions were counterbalanced using a Latin square.

## Interface Types

*Spatial* is making use of the physical space to show all icons in a single grid. They can be navigated by simply moving one's head, and don't require additional interaction. The field of view taken up by this interface type varies from 45 to 100 degrees (see figures 20 and 21), depending on the number of icons.

*Stacked* is a vertically scrolling single column of icons, where overflowing elements at the top and bottom of the column stack in

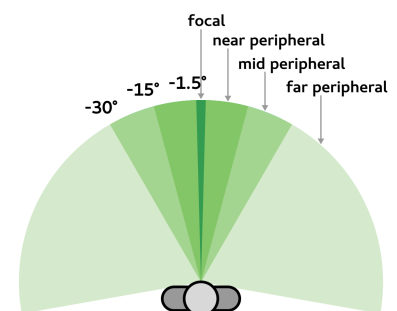


Figure 20: Diagram of the human field of view



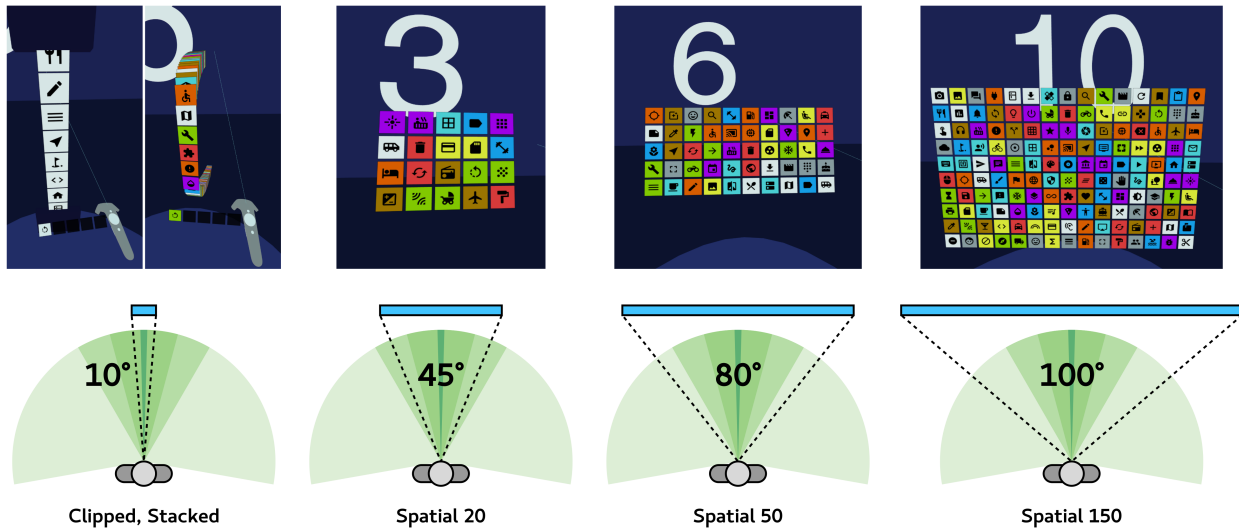


Figure 21: Field of view for the different interface types

the z-dimension. This means that all list elements are always present in the interface and are never completely hidden. Scrolling works by pressing and holding the trigger button and moving the controller up or down. This scrolls the list continuously, following the vertical movement of the controller.

*Clipped* is similar to *Stacked*, except overflowing items disappear instead of stacking. The list is “clipped” at the top and bottom. The scrolling interaction is the same as for *Stacked*.

Figure 22 shows how all three interface types look to participants in VR.

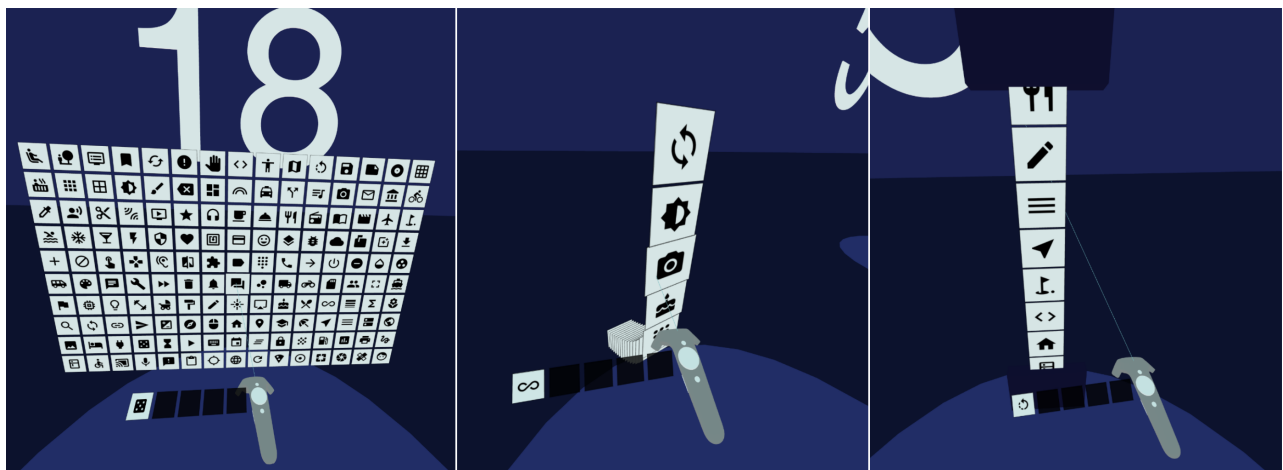


Figure 22: The three different interface types (left to right): *Spatial*, *Stacked*, and *Clipped*

### Number of Items

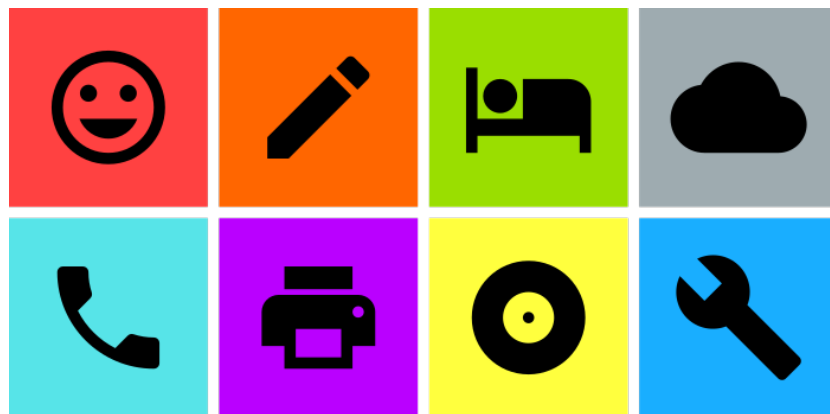
The three different list sizes (20, 50, and 150) were chosen because they allowed us to test a large part of the spectrum of list sizes that realistically appear in user interfaces. We see 150 as the upper bound to the size of arbitrarily ordered, completely unstructured lists that people can handle in a practical manner. In our pilot studies we tested conditions with more items, but participants were very frustrated by the long distances they had to scroll to find icons in these conditions.

### Colored/Monochrome

We included the *Colored* variable in order to understand the differences between *Clipped* and *Stacked* better. When all icons have the same background color, it is not possible to see what icons are on the “stacked” cards at both ends of the list. With different background colors, they can at least be differentiated when they are stacked, even though the actual icons are still obscured. We chose a palette of 10 colors that are each assigned to 10% of the total number of icons. In addition, we also tested all conditions in a *Monochrome* variant, where the background is white for all icons.

### Tasks

The interfaces in all conditions display a number of squares with simple monochrome icons. Participants had to find specific icons, one at a time. The icon set used is a subset of the Material Design icon set used by Android and other Google products (see figure 24).



*Initial:* First, participants had to find 5 icons (see figure 23), which were pseudo-randomly distributed across the entire set (one icon per



Figure 23: The 5 icons fixed to the left of the controller were used to guide the participants through the experiment. During each task, the current target icon was highlighted, the others were black. Selecting the target icon would reveal the next target icon on the controller, and obscure the previous one.

Figure 24: Examples of the icons used in the experiments

quintile). To select the icon, they had to point at it with the raycasting cursor starting from the top of their controller, and press the trigger button.

*Repeat:* After finding each icon once, participants had to find the same 5 icons again, but in randomized order. Each participant had to find and select a total of 180 icons (18 conditions  $\times$  (5 + 5) icons).

## Hypotheses

We conducted the experiment with the following hypotheses in mind:

- H1. *Spatial* outperforms *Clipped* and *Stacked*: *Spatial* allows participants to explore the entire set of icons by simply turning their heads, rather than using the controller, so we assumed it would this condition would be the fastest.
- H2. *Repeat* outperforms *Initial*: During the second round of each condition, participants only had to find icons that they'd found before, which we assumed would be faster due to spatial memory <sup>32</sup>.
  - H2.1. *Clipped* and *Stacked* perform similarly for *Initial*: Due to the similarities in structure and interaction, we assumed that there should be no difference between them in this case.
  - H2.2. *Stacked* outperforms *Clipped* for *Repeat Colored*: Since *Stacked* does not hide elements completely, but stacks them in space, we assumed participants would have a clearer sense of their position and find them more easily the second time.
- H3. *Colored* outperforms *Monochrome*: Since searching for a *Colored* icon requires only looking at icons that have the correct color, we assumed this would be faster than searching through all icons.

<sup>32</sup> Joey Scarr, Andy Cockburn, Carl Gutwin, et al. Supporting and exploiting spatial memory in user interfaces. *Foundations and Trends® in Human-Computer Interaction*, 6(1):1–84, 2013

## Procedure

Participants were given a demographic questionnaire, and then introduced to the VR setup. They put on the headset and were handed the controller. They then received instructions for how to use the interfaces in the experiment from a short tutorial in VR (see figure 25). The tutorial consists of two parts. In the first part the participants were shown how to select icons using the right controller's raycasting cursor.

In the second part they were shown how to scroll in *Clipped* and *Stacked* interfaces, by keeping the trigger button pressed and moving



Figure 25: The five initial steps of the experiment, including 2 tutorial and 3 example steps.

the controller vertically. Participants were told to perform the tasks *as quickly as possible without making any errors*.

After being introduced to the basic interactions, participants got 3 example conditions (one for each interface type). This allowed them to learn the mechanics and get comfortable with the different interfaces before the actual experiment.

Then they performed the tasks for all 18 conditions, counterbalanced using a Latin square. After each condition, they answered a questionnaire (inside the VR environment) with four qualitative statements assessing their experience with a given condition. Figure 26 is a screenshot of what this questionnaire looked like to participants. They rated each of the statements by “clicking” one of the 7 squares using the same raycasting pointer mechanic used during the experimental conditions. Participants rated each of the statements on a 7-point Likert scale (1 = not true at all, 7 = very true).

These are the four statements:

1. I could memorize the position of the items.
2. I was overwhelmed by the number of items.
3. I found the layout efficient to navigate.
4. I could easily find the item I was looking for.

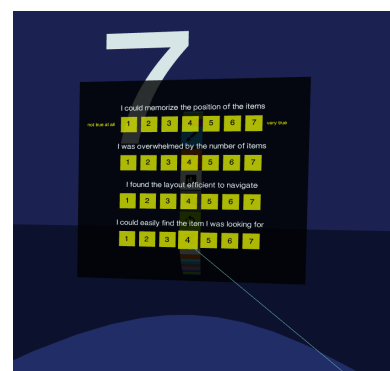


Figure 26: The qualitative questionnaire participants filled in after each of the 18 conditions.

After the experiment, which lasted about 45 minutes, participants were given a questionnaire rating the three interface types (both *Colored* and *Monochrome*) on a 7-point Likert scale (1 = very negatively, 7 = very positively).

### *Data Collection*

For each condition, we logged the time participants took to select each icon. We also logged selection errors (selection of non-target icons), scroll distances for *Clipped* and *Stacked* conditions, and controller movement in the room throughout the experiments.

For each individual icon that had to be selected during the experiment, we logged the following variables:

- The time it took to select select in milliseconds
- The scrolled distance in meters (only for conditions that require scrolling)
- Controller movement in meters
- Selection errors (selection of non-target icons)

Of these variables, only selection time was really necessary for our hypotheses, but we chose to log the additional variables to get a more complete picture and find potential problems in our data more easily.



# Results

This chapter presents the qualitative and quantitative results of our study, including the metrics measured during the experiment, the VR questionnaires after each condition, and the questionnaire at the end of the experiment.

## Quantitative Results

Of the quantitative variables measured during the experiment, *Selection Time* is the most important. It is measured as the time interval between when the target icon is shown on the controller and when it is selected. All our main hypotheses are about quantitative performance, which is assessed using *Selection Time*. Figure 27 gives an overview of our results for the different interface types.

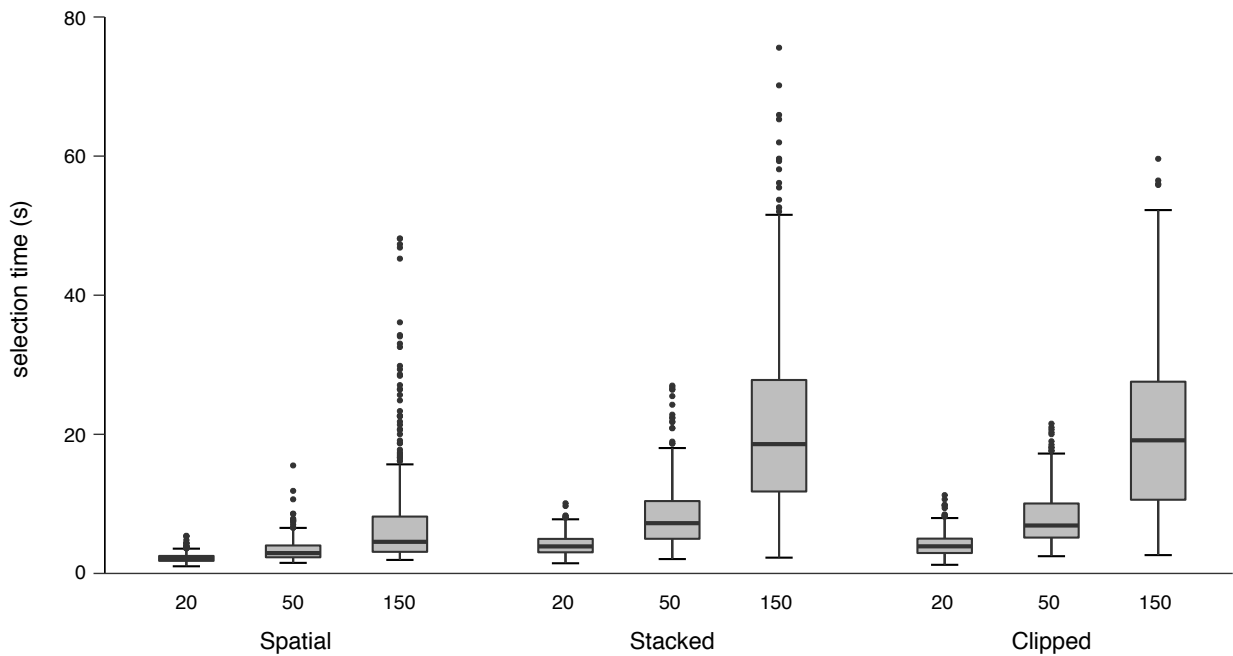


Figure 27: Selection time for all interface types and list sizes

### Summary of Results

Our primary hypotheses were that in *Spatial* conditions, it would take less time to select icons (H1), and that finding icons the second time would be faster than the first time (H2). Both of these were confirmed. In addition, we hypothesized that *Clipped* and *Stacked* should perform similarly for the initial finding of icons (H2.1), which was confirmed, and that *Stacked* should outperform *Clipped* in *Repeat Colored* conditions (H2.2), which was not confirmed. Lastly, we theorized that finding *Colored* icons would be faster than *Monochrome* ones (H3), which was confirmed.

### Data Processing

The data recorded by the experimental software was saved in csv files, aggregated, and cleaned of invalid data points in Excel. 35 out of 3600 trials had 1 or 2 errors, which means that 99.03% were valid. Then we computed the average and standard deviation for all conditions. We did not find any systematic error across conditions.

After removing trials with errors, 44 trials were excluded for having an average larger or smaller than 3 standard deviations than the average of the specific condition. This left us with 3521 (97.81%) valid trials. This is the data we used in our analysis, which we performed using the open source statistics package JASP<sup>33</sup>.

<sup>33</sup> JASP: Open Source Statistical Software (<https://jasp-stats.org>)

### H1: *Spatial* outperforms *Stacked* and *Clipped*

We hypothesized that the selection time would be significantly shorter for *Spatial* conditions compared to both *Stacked* and *Clipped* conditions. Our results confirmed that this is true in both cases (see figure 28). We performed an individual one-way ANOVA ( $\alpha = .05$ ) on the dependent variable selection time. We found a significant main effect from the interface type ( $F(5, 3515) = 98.92, p < .001$ ). A Tukey post-hoc test revealed that the selection time in *Spatial* conditions was significantly lower than in both *Stacked* ( $p < .001$ ) and *Clipped* ( $p < .001$ ) conditions. The mean selection time for *Spatial* conditions was 4.32s ( $SD = 5.05s$ ), for *Stacked* conditions it was 11.25s ( $SD = 10.89s$ ), and for *Clipped* 10.92s ( $SD = 10.09s$ ).

### H2: *Repeat* outperforms *Initial*

Our second hypothesis was that on average, selections should be faster for icons that have already been found previously (*Repeat*), compared to icons the participants have not seen before in a given condition (*Initial*). This was confirmed by our results. We performed

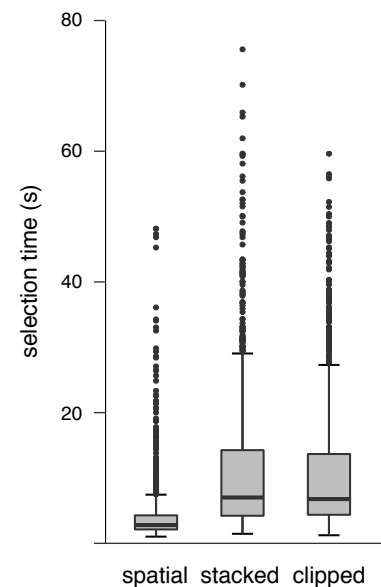


Figure 28: Selection time for the three different experiment types



an individual one-way ANOVA ( $F(\alpha = .05)$ ) on the dependent variable selection time and found a significant main effect from whether conditions were *Initial* or *Repeat* ( $F(1,3519) = 47.27, p < .001$ ). A Tukey post-hoc test revealed that *Repeat* icons were found significantly faster than *Initial* ones ( $p < .001$ ). The mean selection time for *Initial* icons was 9.94s ( $SD = 10.49s$ ), while for *Repeat* icons it was 7.74s ( $SD = 8.47s$ ).

### H2.1: Clipped and Stacked perform similarly for Initial

We hypothesized that for *Initial* icons there should not be any performance difference between *Stacked* and *Clipped*, since participants had not scrolled the list yet and could therefore not rely on spatial memory and the partially visible icons to find their target icons more quickly. We performed an individual one-way ANOVA ( $F(2,1774) = 159.6, \alpha = .05$ ) on the dependent variable selection time using only *Initial* trials, and found a significant main effect from the interface type ( $F(2,1774) = 139.9, p < .001$ ). A Tukey post-hoc test did not reveal a significant difference between *Stacked* and *Clipped* ( $p = .631$ ). The mean selection time for *Stacked* icons was 10.17s ( $SD = 9.81s$ ), while for *Clipped* icons it was 9.75s ( $SD = 9.00s$ ).

### H2.2: Stacked outperforms Clipped for Repeat Colored

The reason why we included *Colored* conditions was because we assumed this could impact the difference in performance between *Stacked* and *Clipped*. When all icons have the same white background, they are not distinguishable at all while stacked. When the backgrounds are colored, it's possible to recognize stacked icons to a limited degree by their color, and see patterns and sequences of colors even if the icons are not directly visible.

Our assumption was that this would result in better performance for *Stacked* compared to *Clipped* in *Repeat Colored* conditions, but this was not confirmed. We performed an individual one-way ANOVA ( $\alpha = .05$ ) on the dependent variable selection time using only *Repeat* trials (see figure 29) and found a significant main effect ( $F(2,882) = 81.92, p < .001$ ) from interface type. However, a Tukey post-hoc test revealed no significant difference between *Stacked* and *Clipped* for *Repeat Colored* icons ( $p = .95$ ). The mean selection time for *Stacked* icons was 9.45s ( $SD = 8.11s$ ), while for *Clipped* icons it was 9.27s ( $SD = 8.90s$ ).

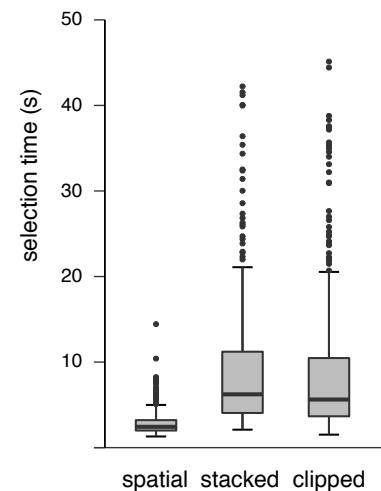


Figure 29: Selection time for different experiment types (only *Repeat Color* icons)

### *H<sub>3</sub>: Colored outperforms Monochrome*

Our last hypothesis was that *Colored* conditions should perform better than *Monochrome* ones on average, because the colors help reduce the number of icons that have to be scanned to only the ones with the right color. This was confirmed by an individual one-way ANOVA ( $\alpha = .05$ ) on the dependent variable selection time, which showed a significant main effect from color ( $F(1, 3519) = 30.79, p < .001$ ). A Tukey post-hoc test revealed that the performance of *Colored* icons was significantly better than *Monochrome* ones ( $p < .001$ ). The mean selection time for *Colored* icons was 7.94s ( $SD = 8.09s$ ), while for *Monochrome* icons it was 9.72s ( $SD = 10.81s$ ). Such a significant result is interesting because we had assumed that colors could potentially be distracting, and therefore impact performance negatively.

### *Qualitative Results*

We collected qualitative data about the user experience of our prototypes using two questionnaires: One per-condition questionnaire that participants filled in inside the VR environment, right after each condition, and a paper questionnaire after the experiment was concluded. The qualitative data confirms the findings of the quantitative study for the most part, as participants preferred *Spatial* conditions over *Stacked* and *Clipped* ones, and *Colored* ones over *Monochrome*. However, there are also some interesting differences, such as the preference for *Stacked* over *Clipped* (even though there is no measurable quantitative performance difference).

### *Post-Condition Questionnaire*

After each of the 18 conditions, participants rated four qualitative statements assessing their experience with a given condition on a 7-point Likert scale. For the most part, the results from these questionnaires match our findings from the quantitative data.

*“I could memorize the position of the items”*: An individual one-way ANOVA ( $\alpha = .05$ ) on the dependent variable rating shows a significant main effect from interface type ( $F(2, 345) = 24.5, p < .001$ ), number of icons ( $F(2, 345) = 46, p < .001$ ), but not color ( $F(1, 345) = 0.029, p = .86$ ).

A Tukey post-hoc test revealed that icon positions in *Spatial* conditions were seen as significantly easier to memorize than in *Stacked* ( $p < .001$ ) and *Clipped* ( $p < .001$ ), while there was no significant difference between *Stacked* and *Clipped* ( $p = .61$ ). We also found a significant differences between all sizes of the conditions, between 20 and 50 ( $p = .049$ ), 20 and 150 ( $p < .001$ ), and 50 and 150 ( $p < .001$ ).

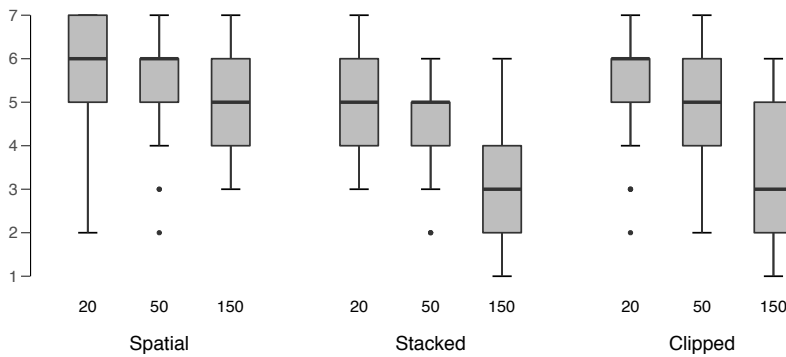


Figure 30: Responses to “I could memorize the position of the items” split by interface type and number of icons (Likert scale, 1=not true at all, 7=very true)

This shows that participants found *Spatial* conditions significantly easier to memorize, but didn’t perceive a difference between *Colored/Monochrome* and *Stacked/Clipped* conditions. The number of icons was also a significant factor, especially for *Stacked* and *Clipped*, as figure 30 shows.

“I was overwhelmed by the number of items”: An individual one-way ANOVA ( $\alpha = .05$ ) on the dependent variable rating shows a significant main effect from interface type ( $F(2, 345) = 7.97, p < .001$ ), number of icons ( $F(2, 345) = 192.78, p < .001$ ), but not color ( $F(1, 345) = 3.03, p = .083$ ).

A Tukey post-hoc test revealed that participants found *Spatial* conditions less overwhelming than *Stacked* ( $p < .001$ ), but there was no difference between *Spatial* and *Clipped* ( $p = .64$ ). *Clipped* was rated to be less overwhelming than *Stacked* ( $p = .024$ ). We also found a significant differences between all sizes of the conditions, between 20 and 50 ( $p < .001$ ), 20 and 150 ( $p < .001$ ), and 50 and 150 ( $p < .001$ ). The fact that participants found *Spatial* conditions least overwhelming is interesting, as we had assumed that especially in larger conditions, showing all elements at once would be overwhelming. As figure 31 shows this is not the case.

“I found the layout efficient to navigate”: An individual one-way ANOVA ( $\alpha = .05$ ) on the dependent variable rating shows a significant main effect from interface type ( $F(2, 345) = 31.12, p < .001$ ), number of icons ( $F(2, 345) = 51.63, p < .001$ ), and color ( $F(1, 345) = 7.88, p = .005$ ).

A Tukey post-hoc test revealed that *Spatial* conditions were seen as significantly more efficient to navigate than in *Stacked* ( $p < .001$ ) and *Clipped* ( $p < .001$ ), while there was no significant difference between *Stacked* and *Clipped* ( $p = 1$ ). *Colored* conditions were rated more efficient than *Monochrome* ones ( $p = .005$ ). We also found a significant differences between all sizes of the conditions, between 20 and 50

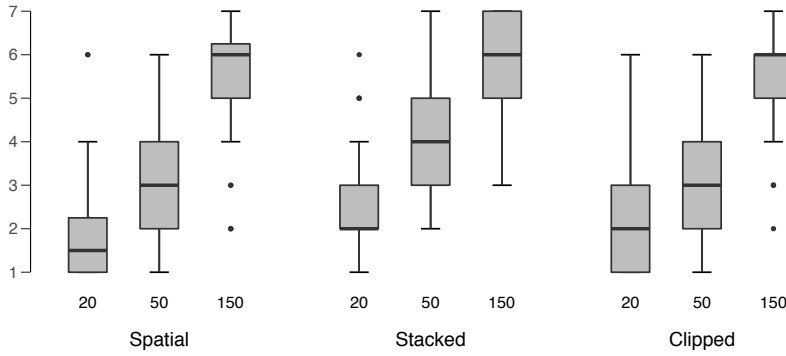


Figure 31: Responses to “I was overwhelmed by the number of items” split by interface type and number of icons (Likert scale, 1=not true at all, 7=very true)

( $p = .006$ ), 20 and 150 ( $p < .001$ ), and 50 and 150 ( $p < .001$ ). This shows that participants found *Spatial* conditions significantly more efficient to navigate, but didn’t perceive a difference between *Stacked* and *Clipped* conditions. The number of icons was also a significant factor, as is clearly visible in figure 32.

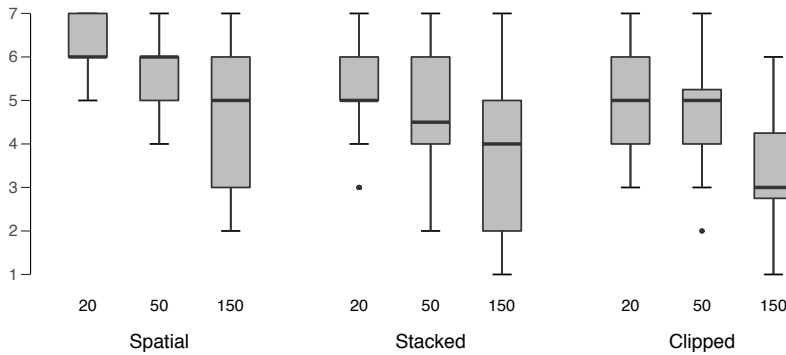


Figure 32: Responses to “I found the layout efficient to navigate” split by interface type and number of icons (Likert scale, 1=not true at all, 7=very true)

“I could easily find the item I was looking for” An individual one-way ANOVA ( $\alpha = .05$ ) on the dependent variable rating shows a significant main effect from interface type ( $F(2, 345) = 21.6, p < .001$ ), number of icons ( $F(2, 345) = 131.35, p < .001$ ), and color ( $F(1, 345) = 8.24, p = .004$ ).

A Tukey post-hoc test revealed that icons in *Spatial* conditions were seen as significantly more easy to find than in *Stacked* ( $p < .001$ ) and *Clipped* ( $p < .001$ ), while there was no significant difference between *Stacked* and *Clipped* ( $p = .72$ ). *Colored* conditions were rated more highly than *Monochrome* ones ( $p = .004$ ). We also found a significant differences between all sizes of the conditions, between 20 and 50 ( $p < .001$ ), 20 and 150 ( $p < .001$ ), and 50 and 150 ( $p < .001$ ). This shows that participants found icons easiest to find in *Spatial* and

*Colored* conditions, but didn't perceive a difference between *Stacked* and *Clipped* conditions (see figure 33). The number of icons was also a significant factor.

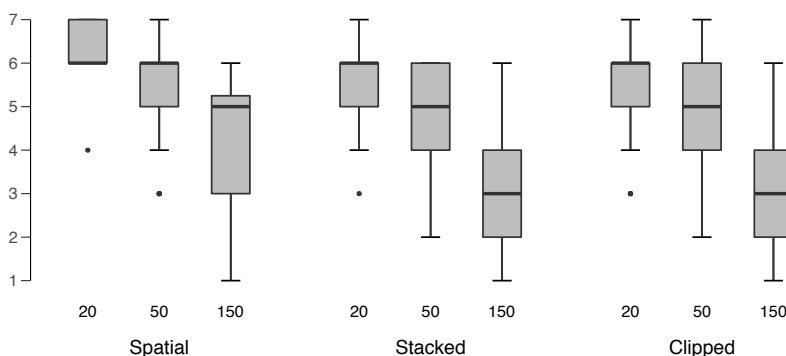


Figure 33: Responses to “I could easily find the item I was looking for” split by interface type and number of icons (Likert scale, 1=not true at all, 7=very true)

### Post-Experiment Questionnaire

After participants completed the experiment, they filled in a short paper questionnaire assessing their experience with the different interfaces. For each combination of interface type and color, they gave a rating on a 7-point Likert scale (1 = very negatively, 7 = very positively).

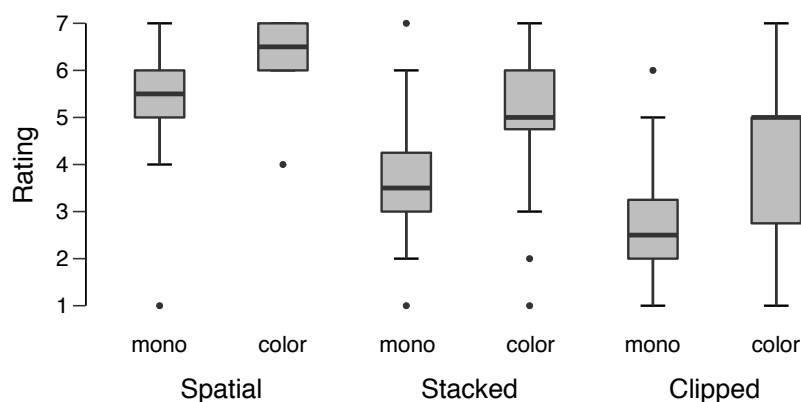


Figure 34: Post-experiment ratings for different experiment types, *Colored* and *Monochrome*

The results mirror the other quantitative and qualitative outcomes, in that *Spatial* and *Colored* conditions were rated most highly. An individual one-way ANOVA ( $F_{\alpha} = .05$ ) on the dependent variable rating shows a significant main effect from interface type ( $F(2, 114) = 30.55, p < .001$ ) and color ( $F(1, 114) = 23.19, p < .001$ ). A Tukey post-hoc test revealed that participants preferred *Spatial* to both *Stacked*

( $p < .001$ ) and *Clipped* ( $p < .001$ ). It also shows that they preferred *Colored* to *Monochrome* ( $p < .001$ ).

However, unlike the quantitative results, these results also show significant differences between *Stacked* and *Clipped*. In the same analysis we also found a significant preference for *Stacked* over *Clipped* ( $p = .015$ ). This shows that participants perceived *Stacked* and *Clipped* differently, despite the fact that there was no significant difference in performance in our experiment.

### *Participant Comments*

The comments we got from participants, both verbally during the experiment, and in the comments field on our post-experiment questionnaire, match our other findings for the most part. The most prevalent comment was that color made it much easier to find items because it narrowed down the search to only icons of the right color. This was mentioned by almost all participants, though one participant expressed finding it confusing in *Spatial* conditions.

*"Color helps find icons easily (even if you don't remember the location). Black/White is difficult to find."*

- participant 17

Many participants also expressed a preference for *Spatial* conditions, stating that they found it faster and easier to find items by looking around rather than scrolling a list.

Several participants also commented on their preference for *Stacked* over *Clipped*. This is consistent with the results from the Likert scale ratings, which also show a preference for *Stacked*. Multiple comments also mentioned being particularly annoyed by large *Clipped* conditions, because the position in the list was not visible.

*"Clipped gives no indication where in the list I am (at 20% or at 80%)."*

- participant 9

One participant mentioned feeling like they were equally efficient with *Clipped* compared to *Stacked*, but *Clipped* required more mental work because they had to rely entirely on their memory.

A few participants had problems with clicking items in scrolling conditions (*Clipped* and *Stacked*), which they mentioned both verbally and in comments. They were all able to complete the tasks, but it may have had a small impact on their performance in these conditions.

Several participants also mentioned feeling like *Stacked* conditions were laggy and unresponsive compared to *Clipped*. This may also have had an impact on their task performance. Interestingly, this did

not seem to impact their preference for *Stacked* over *Clipped*. This may be connected to the fact that they liked how it looked (*“Stacked looks more cool”* - participant 11), and the aforementioned sense of position in the list that *Stacked* provides.

*“Clipped scrolls more smoothly than Stacked.”*

- participant 13





## Discussion

Our results show that spatial interfaces have the potential to make VR interfaces more efficient and pleasurable to navigate. *Spatial* conditions were outperforming other conditions in both the quantitative and qualitative parts of our study. This confirms our initial assumptions, and shows the potential of using 3D space more fully in VR interfaces. Though we did not find any significant performance advantage in *Stacked* over *Clipped*, the fact that participants preferred it and mentioned being able to have a better sense of position within the list does mean that this interface type could potentially be promising, and should be explored further.

### Limitations

Due to performance problems with our implementation, *Stacked* conditions were laggy than *Clipped* ones, rendering at lower frame rates and in some cases dropping frames aggressively. This happened to different degrees for different participants. It is hard to estimate the impact this had on the results, but it should be taken into account when discussing them. It is unlikely that a more performant implementation would have changed the results so drastically in favor of *Stacked* that the difference would have been significant.

Another flaw in our implementation was that scrolling and clicking were done using the same controller button (the right hand trigger). We chose this interaction because we wanted it to be easy to get started with by mimicking the one-finger interaction on touch screens that we assumed all participants would be familiar with. Holding the trigger and moving it further than a certain threshold from the original position would scroll, releasing it without moving it would click. This mechanic worked well for most of our participants, but for a few of them it was difficult to click without moving the controller past the threshold. This meant that they would have to try clicking an icon several times before being able to select it, because they moved the controller too much before releasing and triggered a scroll instead of a click. This got better after the first few conditions,

as participants learned to click more quickly and move the controller less. It is therefore unlikely that it impacted the results significantly.

### *Future Work*

All the interfaces in our experiment had a spatial model. The difference between *Clipped* and the other two conditions is that the spatial model is implicit. Just like on screens, elements that are not currently visible still exist in the user's mental model of the interface, they are just not currently visible. This may be part of the reason why we did not find a significant difference in performance between *Stacked* and *Clipped*. It would therefore be interesting to add interfaces with no spatial model (e.g. a paginated list with no animations) to the experiment. The difficulty in this case would be that the interactions could not be the same (scrolling is spatial by definition), which would make them harder to compare. However, given that pagination with non-spatial page transitions is a common navigation pattern in current-generation VR interfaces, it would be a very relevant comparison.

Another area that could be further explored is different spatial arrangements for items, like our *Stacked* condition. We found that users liked it, but some mentioned that they could not see the elements very well from the front, which negates part of the benefit of this layout. Different layouts, like stacking cards on top and bottom instead of behind the list, could be explored in order to address this problem.

In order to have more context on people's scanning and navigation techniques, it would be interesting to run a similar study while tracking participants' eye movements<sup>34</sup>. This could also help to understand the influence of color more clearly, as most participants mentioned that they found it helpful, while a few expressed that they found it distracting.

All of our participants had relatively little experience with VR. They used our interfaces for the first time during the experiment, and though we did have a set of tutorials introducing them to the interfaces, the novelty of it all was likely a factor in how participants performed in the experiments. It would therefore be interesting to perform a more long-term study, where participants use interfaces multiple times over a longer period. In order to understand whether the performance difference between *Spatial* and the other conditions changes over time, as participants get more comfortable with the more interaction-heavy *Stacked* and *Clipped* conditions.

<sup>34</sup> Stuart K Card. Visual search of computer command menus. *Attention and performance X: Control of language processes*, pages 97–108, 1984

## *Conclusion*

When a new medium emerges, it tends to emulate the constraints and idiosyncrasies of previous mediums at first. For example, early web pages were modeled after printed media, and therefore did not really make use of the new possibilities that screens afforded over print (e.g. dynamic, responsive layouts). As a medium matures, creators learn to use it more effectively, and it comes into its own as a separate medium rather than an extension of existing media. As Virtual and Augmented Reality technology advances and moves closer to the mainstream, we will likely see it undergo this transition as well.

Currently, VR interfaces are emulating traditional 2D interfaces for the most part. As the technology improves and designers become more familiar with the medium, they will increasingly want to make use of the fact that users can move their bodies in 3D space as part of their interactions with interfaces. This is the inherent advantage of VR compared to screen-based 2D interfaces, and it is why we believe that spatial interfaces are a very promising research area.

In this thesis we explored different types of spatial interfaces in Virtual Reality. We surveyed the relevant literature, the most common interaction techniques, and the current state of the technology. We then built a number of prototypes that explore different approaches to spatial VR interfaces, and conducted a study comparing them to a simple clipped list. The results show that spatial interfaces can help make interfaces more efficient and pleasurable to use, but it is dependent on the approach. We also found that even when there is no measurable performance difference, people preferred interfaces with an explicitly visible spatial model over ones that hide it. These results are encouraging, but further experimentation will be necessary to develop robust, scalable interface patterns for organizing information spatially. The dynamic nature of VR as a medium enables a myriad of new ways to display, structure, and navigate information that are yet unexplored.



# Bibliography

VR Wiki - Forte VFX1. <http://vrwiki.wikispaces.com/Forte+VFX1>. Accessed: 2017-11-10.

Hironori Akiduki, Suetaka Nishiike, Hiroshi Watanabe, Katsunori Matsuoka, Takeshi Kubo, and Noriaki Takeda. Visual-vestibular conflict induced by virtual reality in humans. *Neuroscience letters*, 340(3):197–200, 2003.

Stephan Angsüsser. Map folding techniques in the digital age. 2012.

Ronald T Azuma. A survey of augmented reality. *Presence: Teleoperators and virtual environments*, 6(4):355–385, 1997.

Yoseph Bar-Cohen. Haptic devices for virtual reality, telepresence, and human-assistive robotics. *Biol Inspired Intell Robots*, 122:73, 2003.

Benjamin B Bederson. The promise of zoomable user interfaces. *Behaviour & Information Technology*, 30(6):853–866, 2011.

Mark Billinghurst, Adrian Clark, Gun Lee, et al. A survey of augmented reality. *Foundations and Trends® in Human-Computer Interaction*, 8(2-3):73–272, 2015.

Doug Bowman, Chadwick Wingrave, Joshua Campbell, and Vinh Ly. Using pinch gloves (TM) for both natural and abstract interaction techniques in virtual environments. 2001.

Doug A Bowman and Larry F Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 35–ff. ACM, 1997.

Doug A Bowman, Chris North, Jian Chen, Nicholas F Polys, Pardha S Pyla, and Umur Yilmaz. Information-rich virtual environments: theory, tools, and research agenda. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 81–90. ACM, 2003.

Douglas A Bowman and Larry F Hodges. Wimp (widgets, menus, and pointing) design tools for virtual environments. Technical report, Georgia Institute of Technology, 1994.

Douglas A Bowman and Larry F Hodges. User interface constraints for immersive virtual environment applications. Technical report, Georgia Institute of Technology, 1995.

Douglas A Bowman, David Koller, and Larry F Hodges. Evaluation of movement control techniques for immersive virtual environments. Technical report, Georgia Institute of Technology, 1996.

Evren Bozgeyikli, Andrew Raij, Srinivas Katkoori, and Rajiv Dubey. Point & teleport locomotion technique for virtual reality. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, pages 205–216. ACM, 2016.

Grigore C Burdea. Keynote address: haptics feedback for virtual reality. In *Proceedings of international workshop on virtual prototyping. Laval, France*, pages 87–96, 1999.

Stuart K Card. Visual search of computer command menus. *Attention and performance X: Control of language processes*, pages 97–108, 1984.

Rudolph P Darken, William R Cockayne, and David Carmein. The omni-directional treadmill: a locomotion device for virtual worlds. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 213–221. ACM, 1997.

Anthony Elliott, Brian Peiris, and Chris Parnin. Virtual reality in software engineering: Affordances, applications, and challenges. In *Software Engineering (ICSE), 2015 IEEE/ACM 37th IEEE International Conference on*, volume 2, pages 547–550. IEEE, 2015.

Alex Olwal Steven Feiner. The flexible pointer: An interaction technique for selection in augmented and virtual reality. In *Proc. UIST'03*, pages 81–82, 2003.

Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.

Don Norman. *The design of everyday things: Revised and expanded edition*. Basic Books (AZ), 2013.

Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in vr. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 79–80. ACM, 1996.

Sharif Razzaque, Zachariah Kohn, and Mary C Whitton. Redirected walking. In *Proceedings of EUROGRAPHICS*, volume 9, pages 105–106. Manchester, UK, 2001.

Maria V Sanchez-Vives and Mel Slater. From presence to consciousness through virtual reality. *Nature Reviews Neuroscience*, 6(4):332–339, 2005.

Joey Scarr, Andy Cockburn, Carl Gutwin, et al. Supporting and exploiting spatial memory in user interfaces. *Foundations and Trends® in Human–Computer Interaction*, 6(1):1–84, 2013.

Richard Stoakley, Matthew J Conway, and Randy Pausch. Virtual reality on a wim: interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 265–272. ACM Press/Addison-Wesley Publishing Co., 1995.

Ivan E Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 757–764. ACM, 1968.

Lawrence E Warren and Doug A Bowman. User experience with semi-natural locomotion techniques in virtual reality: the case of the virtuix omni. In *Proceedings of the 5th Symposium on Spatial User Interaction*, pages 163–163. ACM, 2017.

Matthias M Wloka and Eliot Greenfield. The virtual tricorder: a uniform interface for virtual reality. In *Proceedings of the 8th annual ACM symposium on User interface and software technology*, pages 39–40. ACM, 1995.

Peter Wozniak, Oliver Vauderwange, Avikarsha Mandal, Nicolas Javahiraly, and Dan Curticapean. Possible applications of the leap motion controller for more interactive simulated experiments in augmented or virtual reality. In *Optics Education and Outreach IV*, volume 9946, page 99460P. International Society for Optics and Photonics, 2016.