VRTactileDraw: A Virtual Reality Tactile Pattern Designer for Complex Spatial Arrangements of Actuators

Oliver Beren Kaul[1], Andreas Domin[1], Michael Rohs[1], Benjamin Simon[1], and Maximilian Schrapel[1]

Leibniz University Hannover, Am Welfengarten 1, 30167 Hannover, Germany
<lastname>@hci.uni-hannover.de
http://www.hci.uni-hannover.de

Abstract. Creating tactile patterns on the body via a spatial arrangement of many tactile actuators offers many opportunities and presents a challenge, as the design space is enormous. This paper presents a VR interface that enables designers to rapidly prototype complex tactile interfaces. It allows for painting strokes on a modeled body part and translates these strokes into continuous tactile patterns using an interpolation algorithm. The presented VR approach avoids several problems of traditional 2D editors. It realizes spatial 3D input using VR controllers with natural mapping and intuitive spatial movements. To evaluate this approach in detail, we conducted a user study and iteratively improved the system. The study participants gave predominantly positive feedback on the presented VR interface (SUS score 79.7, AttrakDiff "desirable"). The final system is released alongside this paper as an open-source Unity project for various tactile hardware.

Keywords: Tactile Patterns · Tactile Pattern Design · Tactile Feedback · Design Tool · Design Interface · VR Tool · Spatial Input.

1 Introduction

Beyond haptic renderings, which may be realized using physics simulations (e.g., contact or impact forces), tactile patterns (TPs) can be used for abstract concepts such as eliciting emotions or guidance during navigation. However, the design of tactile patterns for such abstract concepts requires manual exploration of the design space. It is also interesting for non-technical people (e.g., for personalized touch sensations between remote humans). With the emergence of high-fidelity haptic feedback, the demand for interfaces that can be used to design TPs and effects rose as well. Several works appeared in the recent past [6, 13, 20, 33, 36, 42] but none of these approaches is designed for a high number of actuators that may be spatially oriented in more complex shapes than just a 2D grid.

Fig. 1. The final version of *VRTactileDraw* in action. Users wear an HTC Vive Pro VR headset and a tactile display [26] (center). A user draws symmetric strokes on the 3D model head in VR (left) and then replays the resulting pattern (right). During drawing, the user can feel the resulting tactile actuation.

This work introduces a pattern design interface for tactile feedback systems that feature many actuators in complex spatial arrangements around the body (e.g., [3, 4, 10, 14, 26, 29, 32, 41]). The need for a pattern designer for systems, including many arbitrarily distributed actuators on the human body, can be further motivated by the wide range of novel use cases the systems above enable. For example, full-body suits potentially enable the feeling of physical closeness to a remote person by "distantly touching or brushing" a model body in any desired location, effectively creating a real-time TP. Another example would be creating TPs for an immersive movie where viewers wear a tactile system when watching action-packed scenes and feel specifically designed effects on their body. Imagine feeling the shockwave of an explosion or a giant spider crawling up your spine and over your head before finally becoming visible in the movie scene from above.

In our prior work [24], an iterative design process was followed, which includes several design and implementation phases and two think-aloud studies with feedback from technical and non-technical users. The goal was to develop two variants of an intuitive TP designer (see Fig. 2):

- A *curve interface*, which behaves like an audio/video editor, allowing the user to modify the intensity-over-time curve of each actuator.
- A *drawing interface*, which allows the user to directly draw actuation strokes onto a body part, with interpolation between the actuators.

With the TP designer from our prior work [24], created patterns can be played while drawing. Heat maps provide a live visual representation of vibration intensity. Users who wear the tactile feedback system can simultaneously feel the created pattern. The two variants have different advantages and disadvantages, but we focused on improving the drawing interface, as users tended to prefer it

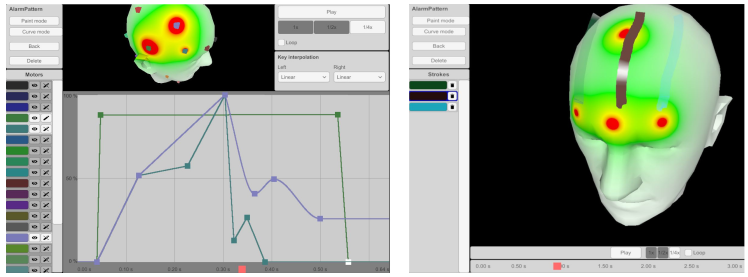


Fig. 2. Curve interface (left) and drawing interface (right) from prior work [24].

over the rather complex curve interface. Some of the most severe disadvantages of the drawing mode are related to the 2D user interface. In particular, the following tasks are difficult to perform with the 2D interface:

- Drawing a stroke on a non-flat body part from a 2D camera perspective as this leads to distortions.
- Moving the camera around the modeled body part while drawing a stroke.
- Adjusting the stroke intensity level while drawing a stroke.

These issues with the 2D interface led us to develop a VR user interface for the same purpose. A VR interface can address the above challenges. It offers increased spatial awareness, ease of moving around a 3D model by simply walking around, and a more direct spatial mapping when drawing. While developing the VR interface, we had the following research questions in mind:

- RQ1 - Usability: What kinds of interactions are suited best for designing tactile patterns in VR? How can the required interface functionality be made as simple and intuitive as possible, and what levels of and usability does the designed VR interface achieve?
- RQ2 - Comfort: How can the VR experience be made comfortable for the user, and what levels of and comfort does the designed VR interface achieve?

The final version of the resulting *VRTactileDraw* system is shown in Figure 1.

2 Background and Related Work

We first give an overview of tactile feedback systems without going into too much detail, as the specific actuator configurations and application areas are less relevant to this work. Then we discuss several TP editors and conclude with the specific prior work on which this paper is based.

2.1 Tactile Feedback

Early work on tactile displays appeared in 1957 by Geldard et al. [16] and was neatly summarized alongside newer work and general guidelines by Jones and

Sarter [23]. Their research review in the area concludes that different levels of vibrotactile intensity and frequency are hard to distinguish and even interfere with each other. Simultaneously, stimulus location and duration are easier to identify and can thus achieve a higher bandwidth of communicated information.

A variety of tactile feedback systems appeared after the initial steps in this domain with a large number of different use cases, including situational awareness, navigation and guidance, vision substitution, obstacle avoidance, notification, target acquisition, and others:

- vibrotactile belts: e.g. [15, 34, 45]
- vibrotactile systems on the arm, wrist, or hand: e.g. [4, 33]
- vibrotactile systems on the head: e.g. [8, 11, 26]
- vibrotactile systems on the back: e.g. [22, 38]
- full body suits: e.g. [3, 10, 14, 29, 32]

Except for full-body, head-worn, and vision substitution systems, most of the systems above feature a relatively low number of actuators in a simple configuration and require only moderate work to define meaningful TPs manually. However, there are systems with large numbers of actuators or complex actuator arrangements [3, 4, 10, 14, 26, 29, 32, 41], which pose an obvious challenge to the design of TPs. They currently require a significant amount of manual work by a pattern designer or algorithmic support to generate meaningful TPs due to their complexity. In such situations and without the support of a suitable interface, creating high-quality TPs is a daunting task. These use cases are likely to profit most from the proposed system, as it is expected to drastically reduce the amount of work needed for generating meaningful TPs. It enables fast prototyping and allows even non-technical users of the system to define their own TPs.

Fig. 3. *HapticHead* for vibrotactile feedback around the head [26].

One system that benefits from the *VRTactileDraw* system is *HapticHead* [26]. *HapticHead* is a vibrotactile interface with a total of 24 actuators located around the head in a sphere-like arrangement (see Fig. 3). The actuator type we used in this work is a common coin-style Precision Microdrives model 312-101 [35] (12500 rpm at 3.3V, 12 mm coin type, 2.6 g normalized amplitude, 40 ms lag time, 132 ms rise time, and 285 ms stop time). Combined with the high update rate of the *HapticHead* system of 90 Hz, this actuator type can playback any tactile intensity, roughness, and rhythm within its limits (namely rise and stop time). Typical, predefined patterns such as sine or sawtooth are also possible.

2.2 Tactile Pattern Editors

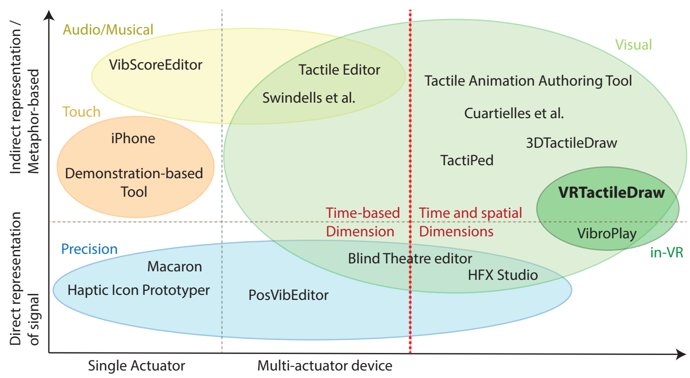


Fig. 4. Design space of TP editors after [33], modified to include *VRTactileDraw* and various other recent works.

Seifi et al. [42] evaluated three possible interfaces that allow users to define TPs for a single actuator in a Wizard of Oz study. They conclude that users want more control (intensity, roughness, and rhythm) over TPs than a simple choice among preset patterns. *Macaron* is a web-based TP editor for single actuator TP design [40]. Swindells et al. published an editor to merge simple haptics with audio and video channels [44].

Cuartielles et al. [6] briefly present four different approaches to TP editors and give recommendations on how a TP editor should allow interaction (touch-based, not overly complex, and allowing immediate replay of the created pattern). The "Blind Theatre Editor" (2009) mentioned in [6] is an approach to generate TPs for up to 64 haptic actuators. However, while it allows defining an intensity curve for each actuator individually on a timeline, like the curve mode in [24], it is also quite complex, shows just a rough indication of spatial actuator position, and can only be used by technicians [6].

The *posVibEditor* [36] aims to support TP design for multiple actuators and arbitrary waveforms but lacks in selecting a particular actuator from a large number of possibilities as the actuators are not visualized on the affected body part. Moreover, each actuator curve has to be defined individually, leading to a high degree of control on the generated pattern but is rather complex. The *bHaptics Designer* [3] by bHaptics Inc. is similar to the *posVibEditor*. It was released alongside a commercial tactile suit and accessorizes for VR gaming and also features grid and timeline editing of TPs using the mouse and keyboard [3].

TactiPEd [33] is a visual interface to easily prototype TPs for multiple actuators. [33] surveys several systems and approaches and discusses their strengths and weaknesses, including some of the ones above (see Fig. 4). The philosophy behind *TactiPEd* is closely related to this work, as the goal is to provide a simple visual interface to define TPs for multiple actuators. While [33] works for actuator arrangements of up to 6 to 8 actuators, such as in a wristband, it is unclear whether it scales to more complex spatial arrangements due to the missing spatial 3D mapping. The *Tactile Animation Authoring Tool*, presented with an algorithm to interpolate between actuators in grid displays, is similar to the algorithm used in this work. It allows users to draw strokes on a 2D grid representation of actuators, thus enabling rapid prototyping of TPs for 1D or 2D tactile grids [38]. It has not been released publicly nor validated in usability studies with standardized usability measures.

HFX Studio [7] is a haptic editor to design patterns for VR use cases with the human perceptual model in mind. The intention is to design haptic patterns for VR, but the actual pattern editing is performed outside of VR using Unity timelines with a mouse and keyboard. Thus, while the aim is related to the aims of *VRTactileDraw*, the interaction concept is very different. In principle, HFX Studio can design TPs of similar spatial complexity like the ones resulting from *VRTactileDraw*. However, the effort and time investments in designing such complex TPs are considerably higher than our approach. In Danieau et al.'s [7] study task of drawing a tactile arrow on a model, users took more than three minutes on average, while the same task takes just a few seconds with *VRTactileDraw*. Nevertheless, HFX studio is one of the most advanced haptic editors, as it aims for high precision and even allows defining effects for other modalities (like thermal and pressure) in addition to vibration.

VibroPlay [21] is the first short concept of an in-VR TP editor that allows direct manipulation of a pattern by touching actuators in a model (only supports binary on or off), which is distinct to our approach. The concept was neither fully documented nor tested in a usability study. Finally, *3DTactileDraw* [24], mentioned in the introduction, shows a first implementation of two possible user interfaces for designing TPs using strokes on a model for a large number of actuators in arbitrary configurations. The implemented curve interface (see Fig. 2, left) is closely related to many of the approaches above. The drawing interface (see Fig. 2, right) pursues a novel approach to define TPs. With the drawing interface, individual actuators no longer have to be defined separately. Instead, the user can draw a stroke on the model of a body part. An interpolation algorithm, first published in [25, 26] and similar to *TactileBrush* [22, 38], takes care of modeling the resulting TP. This drawing interface was preferred by most participants of the user studies that compared the curve interface, and the drawing interface [24]. However, due to the nature of a 2D user interface and mouse and keyboard input, this system has some inherent limitations. For this reason, we decided to realize the TP editor in a VR environment. Unlike most previous related works, the simple and intuitive design of *VRTactileDraw* focuses on novice users.

3 Iterative Design Process and Implementation

We started by reviewing TP design interfaces from related work. We settled on using the Unity IDE for VR scene modeling with the HTC VIVE Pro, including the wireless add-on, as the basis of our system. The HTC VIVE has left and right-hand controllers, which are used simultaneously. These controllers feature several buttons, including an analog trigger button and a touchpad, which offer various possibilities for the design of VR interfaces.

3.1 Virtual Reality Interface Design

Apart from appropriate hardware, a suitable user interface design is needed for effective tool selection and to prevent so-called VR sickness (in particular nausea) from occurring [30]. Previous work on VR sketching showed prototypes of, e.g., a color selector menu [9]. We mostly followed the guidelines by Sherman et al. [43] for general guidelines, Alger [1] for recent VR interfaces, and Lin et al. [31] for measures to prevent VR sickness. Lin et al. [31] recommend using *independent visual backgrounds* to reduce VR sickness. With RQ2 in mind, we implemented this by using a low-poly background from the Unity asset store. We chose a background with few environmental features (e.g., mountains and trees) to prevent distraction and for performance reasons. The VR environment should be rendered at least as fast as the VR headset refresh rate to reduce the likelihood of VR sickness and nausea [43, 46]. Alger recommends using radial menus anchored on the controllers or hands themselves combined with a touchpad [1]. We found this a useful recommendation for implementing several different interface options for the HTC Vive controllers as we aimed to design our interface as simple and intuitive as possible (RQ1). The final mapping of interface functions to controller buttons is shown in Fig. 5. Because drawing in mid-air sometimes leads to inaccuracies, it is necessary to let users rapidly delete and re-draw a stroke in case of erroneous input.

3.2 Goals and Basic Features

We defined the following set of goals for the system, which originate from earlier work [24], VR interface design guidelines mentioned above, our research questions defined in the introduction, and pilot testing:

- **G1** Make the VR interface easy and intuitive to use so that non-technical and first-time VR users can still design their imagined TPs for a complex tactile interface on an experimental basis, without having to rely on external documentation or training.
- **G2** All settings within the interface should be within reach, and easily usable, comparable to *Tilt Brush* [17] and adhere to the design guidelines set by Alger [1].

We identified the following basic features of the VR TP design interface, which originate from pilot testing and adapting related work in VR [24]:

- Drawing strokes of varying intensity levels by "laser pointing" at a modeled body part.
- Instantly rendering the stroke that a user is currently drawing as a TP (the user should see real-time visual feedback and feel tactile feedback).
- Selecting existing strokes by pointing and clicking or through menu buttons.
- Replaying the entire pattern, which may consist of several strokes that may overlap in time.
- Changing the TP playback speed seamlessly within a certain range (e.g., 0.1-3.0 times the original speed).
- Jumping to a desired position within the TP and looping the TP.
- Changing the start time of a stroke relative to other strokes after drawing it.
- Deleting strokes and patterns.

3.3 Design and Implementation

Based on related work and the goals and essential functions outlined above, we started designing a VR interface for TPs from scratch. In our first sketching sessions, we came up with the concept of two radial controller menus as suggested by Alger [1], where one "draw" controller in the user's dominant hand would be responsible for the main functions (e.g., draw, delete stroke/entire pattern, return to the main menu). In contrast, the "select" controller in the user's non-dominant hand is mapped to secondary functions, such as stroke selection, pattern playback speed, start/pause, and current position in the pattern. Furthermore, we designed a simple main menu for choosing between the available patterns, specify global settings, and potentially delete patterns.

We used Unity v5.6.6f for the implementation. The project is designed so that it can easily be used with a variety of tactile interfaces.

We implemented the aforementioned basic features and made them accessible via two controller menus as shown in Figure 5. To quickly delete multiple strokes, we implemented a multi-tap for the "delete stroke" button so that users can delete strokes in rapid succession without a confirmation dialog. We decided not to implement an undo function for this feature, as it is easy to draw a new stroke in case of accidental deletion. Deleting all strokes at once is also possible by deleting the entire pattern in the main menu.

As shown in Fig. 6, we designed a head-up display (HUD) embedded in the upper center of the user's field of view. The HUD shows the current and total time in the pattern, an overlay of all strokes with their respective colors, the selected pattern speed, and whether pattern looping is turned on. It also seamlessly shows the current position on the timeline while changing the current time. This HUD makes it easy to use these functions without looking at the controller. We also decided to show the current stroke intensity percentage as a hint that stroke intensity is proportional to force applied to the trigger button (hidden while no stroke is being drawn).

Since the user can draw multiple strokes which have to be visually distinguished, we select colors for our strokes by picking colors from a color alphabet

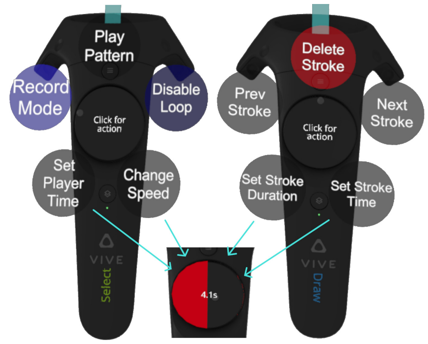


Fig. 5. Final mapping of the controller menu in the final prototype. Some functions were not yet present in the user study. Left: selection controller, right: drawing controller. Menu items can be selected by moving the thumb in the appropriate direction on the touch pad and then pressing down. Pressing down multiple times without fully lifting the thumb triggers the action repeatedly. Some menu items lead to an adjustable slider, which is confirmed by pressing down (bottom center).

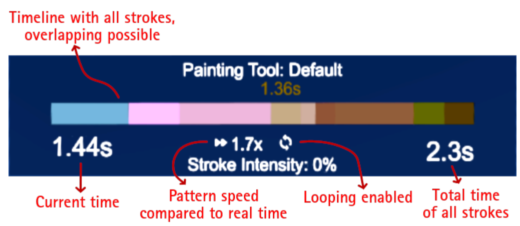


Fig. 6. First version of the head-up display with annotated elements.

designed by Green-Armytage [18]. It ensures a high contrast between the colors and works well on bright backgrounds.

We initially implemented the same interpolation algorithm for tactile actuation as in [24, 26], but finally settled on a different algorithm. The original algorithm was targeted explicitly at guidance, whereas in this work, each actuator's intensity depends on one or more drawn strokes. The original guidance algorithm would sometimes stimulate an actuator farther away to get a person to move their head in that direction. In contrast, the new algorithm always picks the actuators closest to the stroke. The new interpolation algorithm works as follows: For a single position on a stroke, we gather the $N = 3$ closest actuators. These are driven at an intensity proportional to their distance to the stroke position, normalized over all N actuators, and multiplied by the stroke's intensity at the current position (0..1). In case multiple stroke positions of different strokes affect an actuator at the same time, the results are added up per actuator and capped at the maximum intensity. This algorithm is less suitable for tactile interfaces that do not feature a dense layout of tactile actuators. For example, *HapticHead* has closely spaced actuators to take advantage of the tactile funneling illusion, which may cause users to perceive stimulations as smoother and in-between actuators [26, 27]. The interpolation algorithm can easily be replaced by a more appropriate algorithm for less dense arrangements.

4 User Study

We validated our system in a user study with 17 participants from different backgrounds and gathered feedback on possible improvements.

4.1 Design and Study Tasks

We chose the think-aloud method [47] for our study as it helps to expose usability flaws. In a think-aloud study, the user interface should be self-descriptive so that the user can work on a given task without any advice from the experimenter [47].

To make participants fully explore the possibilities of the interface, we designed a set of 10 tasks. Most of these tasks are open so that the participants may take various approaches and may reach different results.

Specifically, the tasks ask the participants to design a TP which:

1. asks the user to stop,
2. notifies the user of an up-leading staircase,
3. asks the user to turn right,
4. asks the user to crouch,
5. lets the user feel a growing tension,
6. asks the user to look up,
7. warns the user of a future earthquake,
8. asks the user to run forward,
9. lets the user feel a slow heartbeat, and
10. lets the user feel a simultaneous vibration left and right.

Tasks 1-8 represent general use cases of the tactile interface. Task 9 was chosen so that users experiment with the "Loop" feature and task 10 requires users to experiment with the "set stroke time" feature.

4.2 Implementation

We implemented the aforementioned counterbalanced tasks in the interface by embedding the textual instruction statically in the scene's background. Users are constantly reminded of what they are currently working on.

4.3 Procedure

After reading and signing an informed consent form and optionally a photographic release form, we explained the think-aloud study method [47] to the participant. For hygienic reasons, participants wore a balaclava under the *HapticHead* prototype, which played the TP. On top of that, they wore the HTC

Vive Pro, which rendered the VR scenes. The only other instruction to each participant was that they are supposed to go back to the main menu after finishing a task, possibly take a break and then start the next task.

Before starting the actual experiment, we tested each of the 24 actuators of the *HapticHead* individually to make sure there were no defects. The participant then started working on the first task after tapping the appropriate "new pattern" button in the main menu. A balanced Latin Square counterbalanced tasks to distribute order effects. In case the participant gave us consent, we also recorded the entire session on video for later analysis. All participants gave us consent for video recording, at least for internal uses. In the end, the participant filled out a final questionnaire containing several Likert scales and comment fields about his or her experiences. Since we wanted to measure the final usability (RQ1) and comfort (RQ2) of our system, we used *system usability scale* (SUS) [5] questions in the final questionnaire, and users also filled out an *AttrakDiff* questionnaire [19]. The participants received a bar of chocolate as a small sign of gratitude.

4.4 Participants

We invited a total of 17 participants (one woman and 16 men, 12 technical and 5 non-technical backgrounds, mean age 24 years, SD 3.4 years). Eight participants had prior experiences with VR headsets, and 11 frequently use game controllers. Six participants indicated drawing about once a month, five about once a year, and the others never performed any drawing activity.

4.5 Results

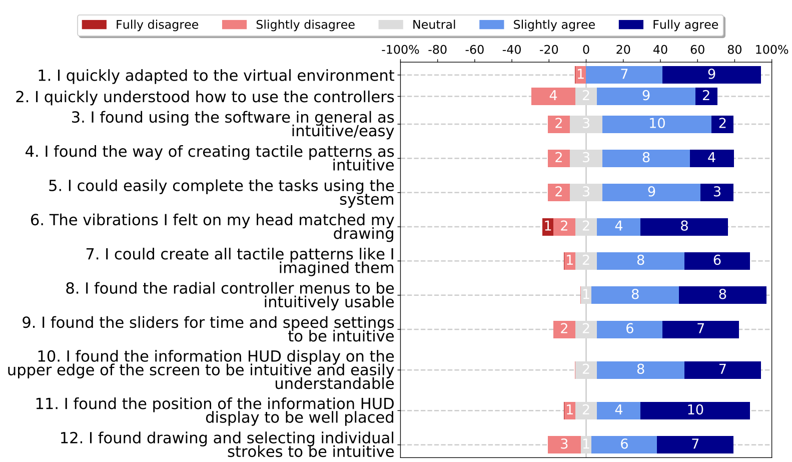


Fig. 7. Questionnaire on intuitiveness and interface design presented as a diverging stacked bar chart.

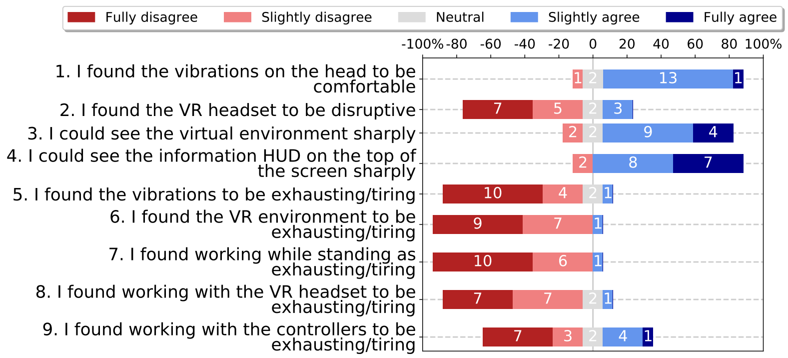


Fig. 8. Questionnaire on comfort presented as a diverging stacked bar chart.

The final questionnaire was split in SUS questions [5], questions on intuitiveness and the design of interface elements (Fig. 7), and the comfort of using the VR interface, *HapticHead*, and VR controllers (Fig. 8). The system usability scale (SUS) score of our system reached a mean of 79.7 (SD=11.2). The AttrakDiff questionnaire [19] was external as we used the official one[1]. Results of AttrakDiff are shown in Fig. 9.

Our participants generally designed quite different patterns, but some general similarities could be found among the responses as seen in Fig. 10.

4.6 Discussion

Generally, the VR TP designer was well received by the study participants. A SUS score of 79.7 is between "good" and "excellent" according to [2]. A meta-analysis of 5000 SUS evaluations showed that a system with a SUS score of 80.3 is better than 90 % of all evaluated systems [5, 37]. The AttrakDiff scores show a similarly positive result. The system is generally rated as desirable (Fig. 9), with a little higher pragmatic than hedonic quality (answers RQ1). However, the AttrakDiff result is influenced by two questions that generally penalize VR systems: Since the system is a solo VR experience, it is more isolating than connective (the real world is completely shut out), and it separates the user from the world and other people around, instead of bringing them closer together (see Fig. 9, right). Without these two categories, the already good AttrakDiff scores would likely be even higher.

There are possible solutions to make users feel less isolated from the world and other people. For example, it is possible to put the entire experience into an AR context instead of VR, using a device like the Microsoft HoloLens. Doing this would make users feel more connected to the real world and further bring them closer to other people. Multiple users could collaboratively work together on a single pattern: All participating users could feel the pattern they collaboratively created. Patterns could be discussed immediately. New ideas could be brainstormed, explored, and refined together.

[1] http://attrakdiff.de/index-en.html – accessed September 04, 2020

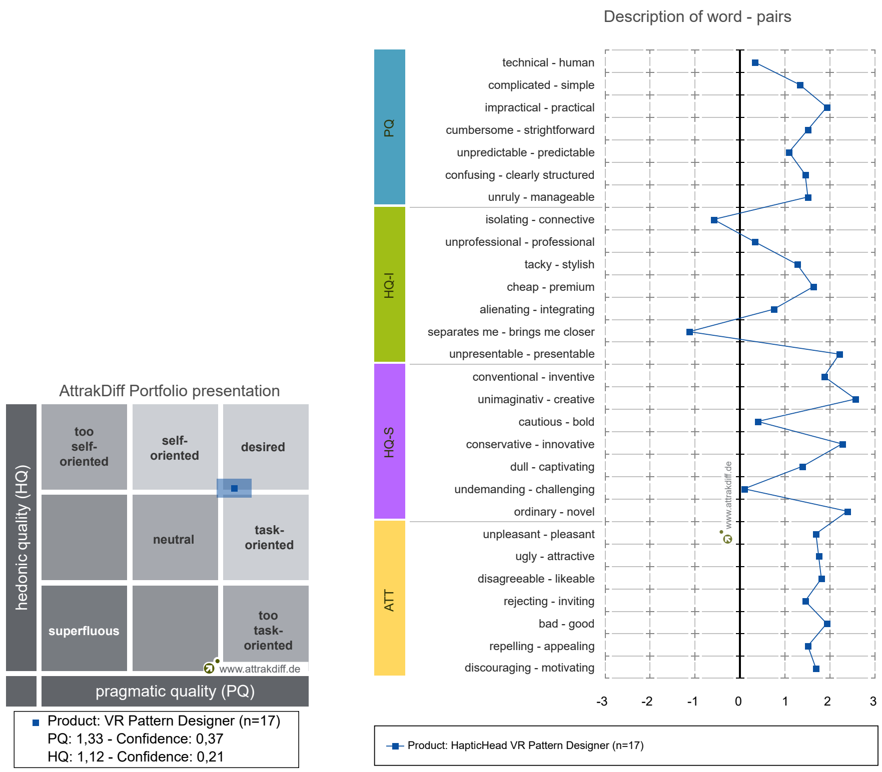
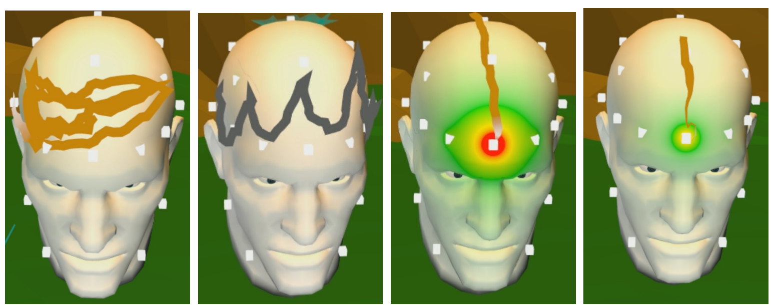


Fig. 9. AttrakDiff: Portfolio of results (left) and Diagram of word pairs (right).

Fig. 7 shows that the overwhelming majority of the participants liked the concept, implementation, and intuitiveness of the design (answers RQ1). Nonetheless, the following analysis concentrates on the few negative responses we got in the questionnaires.

P2 strongly disagreed that the vibrations he felt on his head were matching his drawing. This participant tried to specifically target individual actuators and insert pauses of no actuation, which was difficult with the prototype and chosen interpolation algorithm. It incorporated the $N = 3$ closest actuators instead of just one. While it is possible to add pauses with the “set stroke time” function, this is not as intuitive as a “record mode,” which this participant suggested instead of stopping the time while a stroke is not being drawn. P8 was the other participant who was mostly negative on the questions shown in Fig. 7 and answered positively on some of the tiring questions in Fig. 8. At 33 years, P8 was the oldest participant and did not like the concept of vibrations on the head.

Regarding comfort, Fig. 8 shows that the majority of users were happy with the design and software prototype (answers RQ2). Some participants could not adjust the VIVE Pro headset in such a way that they could sharply see the VR scene and interface. While we made sure that the headset was adequately adjusted at the beginning of the study, it may shift slightly on the head during the study, which leads to the display surface not being in focus. Also, three participants did not like the VR headset with *HapticHead* below it and described it as disruptive or annoying. Nevertheless, only one of our 17 participants would have preferred working with a standard 2D UI instead of a VR interface. Five of the participants agreed that working with the controllers was exhausting or tiring. We can relate this to the clunkiness and weight of the VIVE controllers (203 g each). Future systems might offer different controller types, e.g., Valve index controllers[2], which naturally fit around the hands without needing a permanent firm hand grip.



(a) Task 7, P6 (b) Task 7, P16 (c) Task 8, P7 (d) Task 8, P17

Fig. 10. Example TP designs by our participants. Task 7 was to design a pattern to notify the user of an earthquake, while task 8 encouraged the user to run forward. The heat map in (c) and (d) shows the stroke's start and intensity.

Another interesting finding is the appropriateness, complexity, and similarity of the TPs designed by our participants. While appropriateness is difficult to measure as it is highly individual, pattern complexity varied strongly between participants. Some used only a single short stroke, while others used multiple strokes of varying intensity for the same task. Not all of the designed patterns were similar, but certain tasks had a considerable level of agreement. Fig. 10 shows a number of example designs. The task to design a TP for notifying the user of an earthquake evoked mostly high-intensity patterns, which spread over multiple regions of the head. Other tasks, e.g., the task to encourage the user to run forward, produced simpler lower intensity patterns. An interesting avenue for future research may be a study on how users define and recognize their own TPs and how the TP designs of different users differ in certain relevant aspects (e.g., intensity, complexity, and head region).

In terms of usability comparison vs. related work, we cannot directly compare our system's usability against other TP editors, as no other work we know of published standardized usability measures such as SUS or AttrakDiff. In terms of performance, our system is faster for rapid prototyping of complex TPs than TP editors based on timelines or intensity curves for individual actuators. For

[2] https://www.valvesoftware.com/en/index/controllers – accessed August 20, 2020

example, HFX Studio [7], as one of the most sophisticated TP editors, conducted a study in which participants had to draw a tactile arrow on the back of a torso model. Their users took more than three minutes on average to complete the task, while the same task using *VRTactileDraw* is as simple as drawing an arrow in any drawing application and only takes a few seconds. The only other work using a similar algorithmic interpolation approach as *VRTactileDraw*, the *Tactile Animation Authoring Tool* [38], is likely to have a similar TP prototyping performance, but only in the case of a 1D or 2D tactile grid. However, we are not aware of a public release of the tool or any usability evaluation with standardized usability measures.

4.7 Improvements After the User Study

We implemented the following features that our participants suggested:

- *Set stroke duration* function: Allows changing the total duration of an individual stroke after drawing it with visual feedback (changing bar length) in the HUD. Eight participants suggested this.
- *Symmetric mirror drawing mode*: Strokes on one side of the model are simultaneously drawn on the other side of the model (symmetric relative to the mid-sagittal plane). This was requested by five participants and generalizes to other systems beyond *HapticHead* due to the human body's symmetry.
- *Show actuator positions* option: This option in the main menu shows the positions of all actuators so that users can consider actuator locations while drawing patterns. Four participants requested this.
- *Record mode*: While pushing a record menu button on the select (left-hand) controller, time moves on even while not drawing a stroke compared to the regular mode, in which pattern recording time stops when no stroke is being drawn. A red recording icon on the HUD shows whether the record mode is active at a given time. A single participant suggested this function. We still found it to be a worthwhile addition.

With these changes, there are two drawing modes: *normal* and *mirror*. Switching between these modes is performed by the VIVE controller grip button. Otherwise, most of the new functions can be reached through the controller menus' final mapping (see Fig. 5).

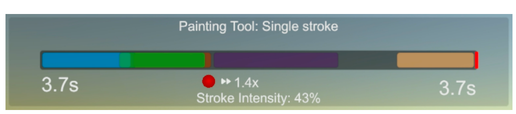


Fig. 11. Final version of the HUD. Recording mode is currently active and a stroke is being drawn.

We also polished the overall look and feel of the HUD by adding a vertical red time indicator, choosing appropriate background transparency, and hiding elements that are not strictly necessary in a given context (see Fig. 11).

Even though related work [38, 39] found that users might want to manipulate created paths after initial creation, only three participants would have liked an option to manipulate stroke intensity after creation. No participant suggested altering the strokes spatially. Thus, we chose not to include an option to manipulate strokes, as the current implementation allows rapid deletion and re-drawing of a stroke, which is only marginally slower than selecting a different tool and manipulating a stroke.

5 Limitations

While our software prototype already received predominantly positive feedback, we did not conduct a follow-up study of the changes made after the user study. Since we implemented features that the participants suggested and that we considered valuable, we estimate the SUS and AttrakDiff scores to improve with these changes. We aim for the final verification of this hypothesis in future work.

Nine of our 17 participants in the user study had no prior experience with VR systems. Thus a novelty effect [28] cannot be precluded. Even if we had recruited only participants with prior VR experience, a novelty effect could still occur as the TP editor itself would be novel to them. Thus, the ratings of our TP editor are likely positively biased. The novelty effect wears off over time [28], so a long-term study in future work may procure more accurate ratings.

6 Open Source Release and Extension to Other Prototypes

The user study was conducted with a tactile display from prior work [25, 26]. However, the prototype can easily be extended to support different body-worn tactile feedback systems, like tactile vests or full-body suits [3, 4, 10, 14, 26, 29, 32, 41]. We provide an open-source release[3] of *VRTactileDraw*. The interface is implemented in a modular, extensible way so that it is easy to replace the main components to fit specific requirements. In order to use the *VRTactileDraw* editor with a different tactile output system, a developer has to perform the following steps, further specified in the documentation:

- Replace the provided models of the head and human body with a targeted body part model (e.g., chest).
- Create copies of the *Actuator Prefab* model, assign unique ids, and position the actuators according to the tactile system's specification. This is all done inside the Unity editor.
- Implement a script to drive the new tactile system, including the system-specific communication protocol. The *RaspberryCommandSender* class may serve as an example.

[3] VRTactileDraw open-source release: https://github.com/obkaul/VRTactileDraw

With the open-source release, we provide a documented framework for developers to use on their own tactile prototypes. The original software prototype for *HapticHead* is included in the framework. Besides, we provide other examples of tactile displays: two different tactile vests, including the respective 3D model of a complete human body.

Fig. 12. *VRTactileDraw* driving a 52-actuator tactile vest mockup, and the users' view inside an HTC Vive

Figure 12, left, shows *VRTactileDraw* in action on a virtual tactile vest with 52 actuators around the torso and over the shoulders. The VR software components are fully implemented, but the tactile output system is currently a mockup only.

Fig. 13. *VRTactileDraw* driving the MultiWave prototype vest [41]: A wearable vibrotactile vest containing 76 actuators (left) and the user's view inside an HTC Vive (right). The user is replaying a TP and simultaneously feels vibrations as indicated by the heatmap.

Figure 13 shows *VRTactileDraw* driving a fully implemented tactile vest system [41]. *MultiWave* is an FPGA-based controller for multiple tactile actuators connected to a tactile vest prototype, which is intended for mobile haptic feedback in virtual or augmented reality [41]. It consists of 76 actuators controlled by *MultiWave* via Bluetooth or WiFi. *VRTactileDraw* can be used to generate TPs for *MultiWave* and the tactile vest, e.g., to realize navigation for visually impaired people and to provide orientation feedback and effects in VR environments.

7 Conclusion and Future Work

We present the first full-fledged design, implementation, and evaluation of a TP editor where the actual TP editing process is entirely conducted inside a virtual environment. VR interfaces are well suited for designing TPs for complex spatial arrangements of tactile actuators. Users can freely move around and generally have a better understanding of spatial relationships than in 2D UIs (see [12]). Our system is also more scalable in terms of the number of actuators and spatial actuator configurations than traditional timeline-based TP pattern design UIs. In traditional timeline-based TP editors, a designer has to define an intensity curve for each actuator. Simultaneously, in our approach, an interpolation algorithm takes care of calculating intensity curves of all actuators in real-time based on simple strokes drawn by potentially novice users.

VRTactileDraw is generally easy to use and understand, especially when considering the complexity of the created patterns generated out of simple strokes. Thus, we can answer the research questions defined in the introduction as follows: The presented VR interface with its minimalistic environment and self-explanatory interactions is indeed highly suitable to design tactile patterns as it allows rapid prototyping, the SUS, and AttrakDiff scores indicate a desirable system and most of our users were able to design TPs without any prior knowledge freely and rated it as generally intuitive (see Figure 7 and 9, answers RQ1), and it was highly accepted amongst our study participants in terms of comfort (see Figure 8, answers RQ2).

Apart from extending our system to other tactile prototypes, as mentioned in the previous section, another direction for future work was already hinted at in the discussion: Multiple users could collaboratively design TPs, either in the same VR environment and simultaneously experience the created patterns or in an AR variant that shows the natural environment, other users, as well as the designed TPs. This collaboration would make users feel less isolated from the world and their colleagues, facilitate discussion about and refine patterns, and probably lead to a better overall result. However, it would require that each user is equipped with a tactile feedback device.

These collaborating users would not even need to be at the same physical location but could be represented by avatars and work together at a distance, sharing the created patterns. Furthermore, one user could draw a tactile stroke on a body part, while another user would simultaneously feel the tactile feedback. This would allow for the exploration of novel use cases compared to simple TP design, like feeling the touch of another person, hugs, "cuddles," and possibly experiencing a faraway person's emotions. Future work may also investigate the effects of drawing TPs from a third-person perspective and experiencing the synchronous tactile sensation on the process of designing tactile patterns and the outcome.

All the use cases mentioned above of extending our concept to multiple users are another considerable advantage of our in-VR (or potentially AR) concept over traditional TP designers. These are not easily extended to multiple collaborating users working in the same environment.

Bibliography

[1] Alger, M.: Visual Design Methods for Virtual Reality. Personal Website (September), 98 (2015), http://aperturesciencellc.com/vr/VisualDesignMethodsforVR_MikeAlger.pdf

[2] Bangor, A., Kortum, P., Miller, J.: Determining what individual SUS scores mean. Journal of Usability Studies **4**(3), 114–123 (2009)

[3] BHaptics: bHaptics Tactot (2020), https://www.bhaptics.com/tactsuit/

[4] Bloomfield, A., Badler, N.I.: A Low Cost Tactor Suit for Vibrotactile Feedback A Low Cost Tactor Suit for Vibrotactile Feedback. Technical Reports (CIS) (January), 7 (2003)

[5] Brooke, J.: SUS : A Retrospective **8**(2), 29–40 (2013)

[6] Cuartielles, D., Göransson, A., Olsson, T., Stenslie, S.: Developing Visual Editors for High-Resolution Haptic Patterns. The Seventh International Workshop on Haptic and Audio Interaction Design pp. 42–44 (2012), http://www.forskningsdatabasen.dk/en/catalog/2389362226

[7] Danieau, F., Guillotel, P., Dumas, O., Lopez, T., Leroy, B., Mollet, N.: HFX Studio: Haptic editor for full-body immersive experiences. Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST (2018). https://doi.org/10.1145/3281505.3281518

[8] de Jesus Oliveira, V.A., Brayda, L., Nedel, L., Maciel, A.: Designing a Vibrotactile Head-Mounted Display for Spatial Awareness in 3D Spaces. IEEE Transactions on Visualization and Computer Graphics **23**(4), 1409–1417 (apr 2017). https://doi.org/10.1109/TVCG.2017.2657238, http://ieeexplore.ieee.org/document/7829406/

[9] Deering, M.F.: HoloSketch: A Virtual Reality Sketching / Animation Tool. ACM Transactions on Computer-Human Interaction (TOCHI) **2**(3), 220–238 (1995). https://doi.org/10.1145/210079.210087

[10] Delazio, A., Nakagaki, K., Lehman, J.F., Klatzky, R.L., Sample, A.P., Hudson, S.E.: Force jacket: Pneumatically-actuated jacket for embodied haptic experiences. Conference on Human Factors in Computing Systems - Proceedings **2018-April**, 1–12 (2018). https://doi.org/10.1145/3173574.3173894

[11] Dobrzynski, M.K., Mejri, S., Wischmann, S., Floreano, D.: Quantifying Information Transfer Through a Head-Attached Vibrotactile Display: Principles for Design and Control. IEEE Transactions on Biomedical Engineering **59**(7), 2011–2018 (jul 2012). https://doi.org/10.1109/TBME.2012.2196433, http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6189750

[12] Dünser, A., Steinbügl, K., Kaufmann, H., Glück, J.: Virtual and augmented reality as spatial ability training tools. ACM International Conference Proceeding Series **158**, 125–132 (2006). https://doi.org/10.1145/1152760.1152776

[13] Enriquez, M.J., MacLean, K.E.: The hapticon editor: A tool in support of haptic communication research. In: Proceedings - 11th

Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, HAPTICS 2003. pp. 356–362. IEEE Comput. Soc (2003). https://doi.org/10.1109/HAPTIC.2003.1191310, http://ieeexplore.ieee.org/document/1191310/

[14] Erp, J.B.F.V., Veen, H.A.H.C.V.: A Multi-purpose Tactile Vest for Astronauts in the International Space Station. Proceedings of eurohaptics pp. 405–408 (2003)

[15] Flores, G., Kurniawan, S., Manduchi, R., Martinson, E., Morales, L.M., Sisbot, E.A.: Vibrotactile Guidance for Wayfinding of Blind Walkers. IEEE Transactions on Haptics **8**(3), 306–317 (jul 2015). https://doi.org/10.1109/TOH.2015.2409980

[16] Geldard, F.A.: Adventures in tactile literacy. American Psychologist **12**(3), 115–124 (1957). https://doi.org/10.1037/h0040416, http://content.apa.org/journals/amp/12/3/115

[17] Google LLC: Tilt Brush by Google (2016), https://www.tiltbrush.com/

[18] Green-Armytage, Paul: A Colour Alphabet and the Limits of Colour Coding. JAIC - Journal of the International Colour Association **5**(0) (2010), http://jaic.jsitservices.co.uk/index.php/JAIC/article/view/19

[19] Hassenzahl, M., Burmester, M., Koller, F.: AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität. pp. 187–196. Vieweg+Teubner Verlag (2003). https://doi.org/10.1007/978-3-322-80058-9_19, http://link.springer.com/10.1007/978-3-322-80058-9_19

[20] Hong, K., Lee, J., Choi, S.: Demonstration-based vibrotactile pattern authoring. In: Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction - TEI '13. p. 219. ACM Press, New York, New York, USA (2013). https://doi.org/10.1145/2460625.2460660

[21] Huang, D.Y., Chan, L., Jian, X.F., Chang, C.Y., Chen, M.H., Yang, D.N., Hung, Y.P., Chen, B.Y.: VibroPlay: Aaorind three-dimensional spatial-temporal tactile effects with direct manipulation. SA 2016 - SIGGRAPH ASIA 2016 Emerging Technologies (2016). https://doi.org/10.1145/2988240.2988250

[22] Israr, A., Poupyrev, I.: Tactile brush. In: Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11. p. 2019. ACM Press, New York, New York, USA (may 2011). https://doi.org/10.1145/1978942.1979235

[23] Jones, L.A., Sarter, N.B.: Tactile displays: Guidance for their design and application. Human Factors **50**(1), 90–111 (feb 2008). https://doi.org/10.1518/001872008X250638, http://journals.sagepub.com/doi/10.1518/001872008X250638

[24] Kaul, O.B., Hansing, L., Rohs, M.: 3DTactileDraw: A Tactile Pattern Design Interface for Complex Arrangements of Actuators. In: Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems - CHI EA '19. pp. 1–6. ACM Press, New York, New York, USA (2019). https://doi.org/10.1145/3290607.3313030

[25] Kaul, O.B., Rohs, M.: HapticHead: 3D Guidance and Target Acquisition through a Vibrotactile Grid. In: Proceedings of the 2016 CHI Conference

Extended Abstracts on Human Factors in Computing Systems - CHI EA '16. pp. 2533–2539. ACM Press, New York, New York, USA (may 2016). https://doi.org/10.1145/2851581.2892355

[26] Kaul, O.B., Rohs, M.: HapticHead: A Spherical Vibrotactile Grid around the Head for 3D Guidance in Virtual and Augmented Reality. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17. pp. 3729–3740. ACM Press, New York, New York, USA (2017). https://doi.org/10.1145/3025453.3025684

[27] Kaul, O.B., Rohs, M., Simon, B., Demir, K.C., Ferry, K.: Vibrotactile Funneling Illusion and Localization Performance on the Head. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems - CHI '20. pp. 1–13. Association for Computing Machinery (ACM), Honolulu, Hawaii, USA (apr 2020). https://doi.org/10.1145/3313831.3376335

[28] Koch, M., Von Luck, K., Schwarzer, J., Draheim, S.: The Novelty Effect in Large Display Deployments-Experiences and Lessons-Learned for Evaluating Prototypes. In: ECSCW 2018 - Proceedings of the 16th European Conference on Computer Supported Cooperative Work. European Society for Socially Embedded Technologies (EUSSET) (2018), https://dl.eusset.eu/handle/20.500.12015/3115

[29] Konishi, Y., Hanamitsu, N., Outram, B., Kamiyama, Y., Minamizawa, K., Sato, A., Mizuguchi, T.: Synesthesia Suit. pp. 499–503. Springer, Singapore (nov 2018). https://doi.org/10.1007/978-981-10-4157-0_84, http://link.springer.com/10.1007/978-981-10-4157-0_84

[30] LaViola, J.J., J., J.: A discussion of cybersickness in virtual environments. ACM SIGCHI Bulletin **32**(1), 47–56 (jan 2000). https://doi.org/10.1145/333329.333344, http://portal.acm.org/citation.cfm?doid=333329.333344

[31] Lin, J.J.W., Abi-Rached, H., Kim, D.H., Parker, D.E., Furness, T.A.: A Natural Independent Visual Background Reduced Simulator Sickness. Proceedings of the Human Factors and Ergonomics Society Annual Meeting **46**(26), 2124–2128 (sep 2002). https://doi.org/10.1177/154193120204602605, http://journals.sagepub.com/doi/10.1177/154193120204602605

[32] Lindeman, R.W., Page, R., Yanagida, Y., Sibert, J.L.: Towards full-body haptic feedback: The design and deployment of a spatialized vibrotactile feedback system. Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST pp. 146–149 (2004)

[33] Paneels, S., Anastassova, M., Strachan, S., Van, S.P., Sivacoumarane, S., Bolzmacher, C.: What's around me Multi-actuator haptic feedback on the wrist. In: 2013 World Haptics Conference, WHC 2013. pp. 407–412 (2013)

[34] Pielot, M., Krull, O., Boll, S.: Where is my team? Supporting Situation Awareness with Tactile Displays. In: Proceedings of the 28th international conference on Human factors in computing systems - CHI '10. p. 1705. ACM Press, New York, New York, USA (apr 2010). https://doi.org/10.1145/1753326.1753581

[35] Precision Microdrives: Precision Microdrives 312-101 (2017), https://www.precisionmicrodrives.com/product/312-101-12mm-vibration-motor-3mm-type

[36] Ryu, J., Choi, S.: posVibEditor: Graphical authoring tool of vibrotactile patterns. In: 2008 IEEE International Workshop on Haptic Audio visual Environments and Games. pp. 120–125. IEEE (oct 2008). https://doi.org/10.1109/HAVE.2008.4685310, http://ieeexplore.ieee.org/document/4685310/

[37] Sauro, J.: A practical guide to the system usability scale: Background, benchmarks & best practices. Measuring Usability LLC (2011)

[38] Schneider, O.S., Israr, A., MacLean, K.E.: Tactile Animation by Direct Manipulation of Grid Displays. In: Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15. pp. 21–30. ACM Press, New York, New York, USA (2015). https://doi.org/10.1145/2807442.2807470

[39] Schneider, O.S., Maclean, K.E.: Improvising design with a Haptic Instrument. IEEE Haptics Symposium, HAPTICS pp. 327–332 (2014). https://doi.org/10.1109/HAPTICS.2014.6775476

[40] Schneider, O.S., MacLean, K.E.: Studying design process and example use with Macaron, a web-based vibrotactile effect editor. IEEE Haptics Symposium, HAPTICS **2016-April**, 52–58 (2016). https://doi.org/10.1109/HAPTICS.2016.7463155

[41] Schrapel, M., Loewen, S., Rohs, M.: MultiWave: A Fully Customizable Mobile Function Generator For Haptic Feedback In VR. Contest (2019)

[42] Seifi, H., Anthonypillai, C., MacLean, K.E.: End-user customization of affective tactile messages: A qualitative examination of tool parameters. In: 2014 IEEE Haptics Symposium (HAPTICS). pp. 251–256. IEEE (feb 2014). https://doi.org/10.1109/HAPTICS.2014.6775463, http://ieeexplore.ieee.org/document/6775463/

[43] Sherman, W.R., Craig, A.B.: Understanding virtual reality : interface, application, and design (2018), https://books.google.de/books?hl=en&lr=&id=D-OcBAAAQBAJ

[44] Swindells, C., Pietarinen, S., Viitanen, A.: Medium fidelity rapid prototyping of vibrotactile haptic, audio and video effects. IEEE Haptics Symposium, HAPTICS pp. 515–521 (2014). https://doi.org/10.1109/HAPTICS.2014.6775509

[45] Tsukada, K., Yasumura, M.: ActiveBelt: Belt-Type Wearable Tactile Display for Directional Navigation, vol. 3205 (2004), http://link.springer.com/chapter/10.1007/978-3-540-30119-6_23

[46] Unity Technologies: Unity - Manual: VR overview (2019), https://docs.unity3d.com/Manual/VROverview.html

[47] Van Someren, M.W., Barnard, Y.F., Sandberg, J.A.C.: The think aloud method: a practical approach to modelling cognitive processes (1994)