3DTactileDraw: A Tactile Pattern Design Interface for Complex Arrangements of Actuators

Oliver Beren Kaul, Leonard Hansing, Michael Rohs
Leibniz University Hannover (LUH)
Hannover, Germany
<lastname>@hci.uni-hannover.de

ABSTRACT

Creating tactile patterns for a grid or a 3D arrangement of a large number of actuators presents a challenge as the design space is huge. This paper explores two different possibilities of implementing an easy-to-use interface for tactile pattern design on a large number of actuators around the head. Two user studies were conducted in order to iteratively improve the prototype to fit user needs.

KEYWORDS

Tactile Patterns; Tactile Pattern Design; Tactile Feedback; Design Interface

INTRODUCTION

With the emergence of high-fidelity haptic feedback, the demand for interfaces that can be used to design haptic or tactile patterns (TPs) and effects rose as well. Several works appeared in the recent past [1, 2, 4, 10–12] but none of these approaches are designed for a high number of actuators that may be spatially oriented in more complex shapes than just a 2D grid. *HapticHead* [6–8] is a

CHI'19 Extended Abstracts, May 4–9, 2019, Glasgow, Scotland UK

ACM ISBN 978-1-4503-5971-9/19/05.
https://doi.org/10.1145/3290607.3313030

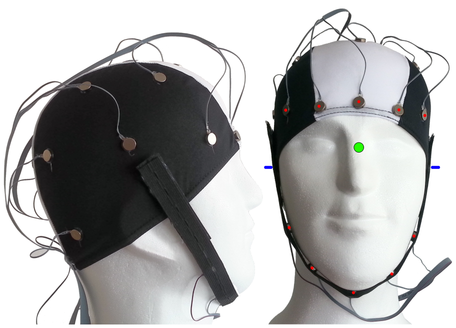
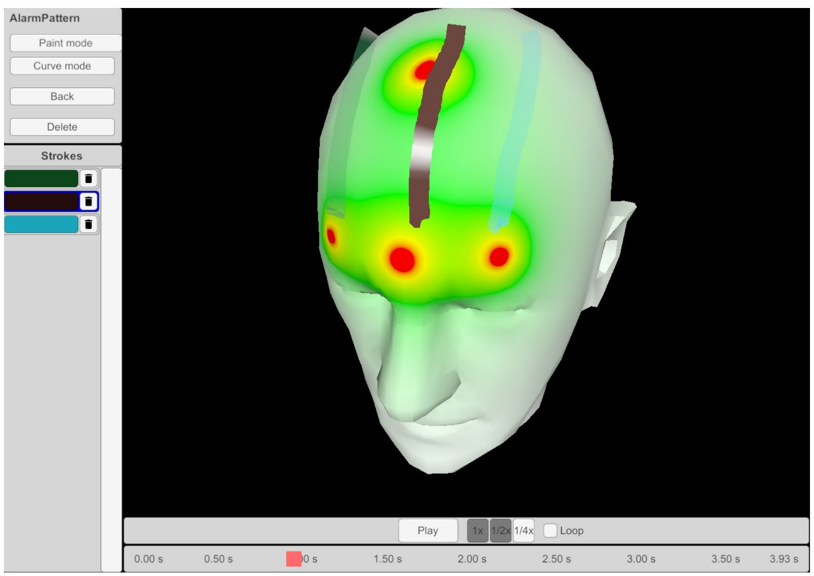


Figure 1: HapticHead, a vibrotactile interface around the head [8].

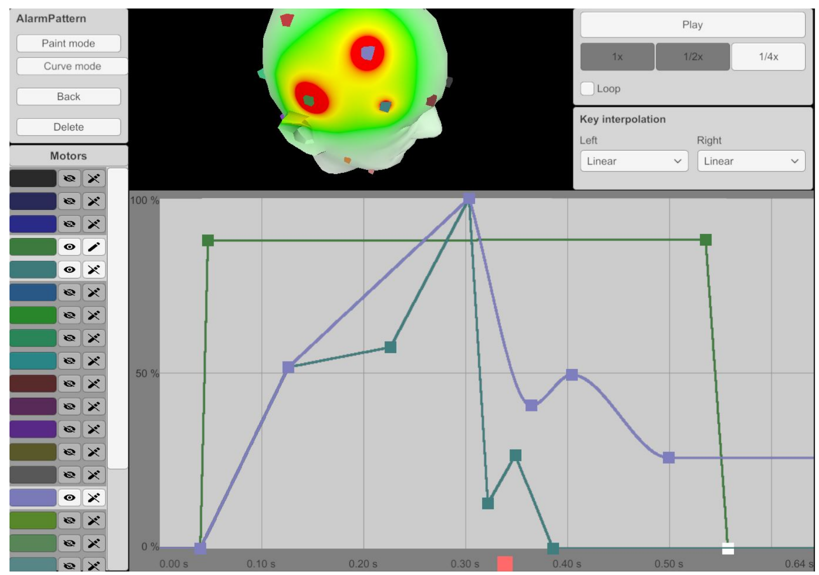
vibrotactile display around the head consisting of a bathing cap with a chin strap and a total of 24 vibrotactile actuators (see Fig. 1). It can be used in 3D guidance and localization scenarios in both virtual (VR) and augmented reality (AR) with relatively high precision and low task completion times. This work introduces a pattern design interface for tactile interfaces like HapticHead that feature a large number of actuators in complex arrangements around the body.

In an iterative design process including several design and implementation phases as well as two think-aloud user studies with feedback from potential users (technical and non-technical backgrounds alike) we designed two possible variants of an intuitive design interface (see Fig. 2):

- A *paint interface* which allows the user to directly draw actuation strokes onto an affected body part (Fig. 2a).
- A *curve interface* which behaves similar to video editors, allowing to modify the intensity-over-time curve of each actuator directly (Fig. 2b).



(a) Paint Interface



(b) Curve Interface

Figure 2: The implemented interface types: (a) the paint interface and (b) the curve interface.

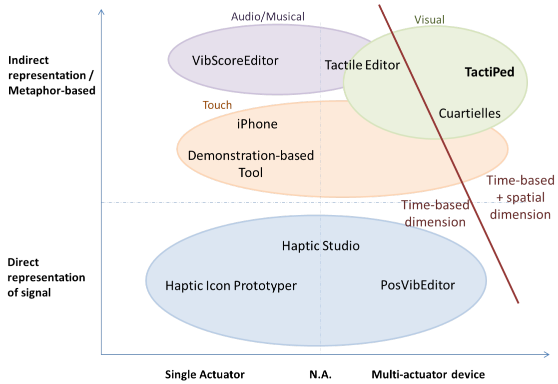


Figure 3: Design space of tactile pattern editors by Panëels et al. [10], used with permission.

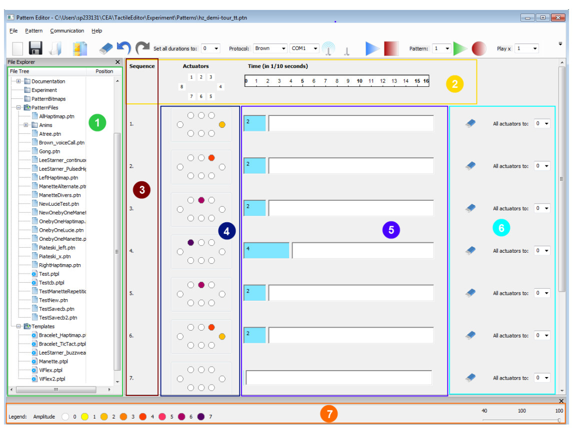


Figure 4: TactiPEd tactile editor by Panëels et al. [10], used with permission.

Both interfaces allow the instant and simultaneous play of the created patterns. While the implemented interface demonstrates its features specifically for the HapticHead prototype, it could easily be adapted to cover other tactile feedback systems with a large number of actuators such as vests, gloves, or even full body-suits.

BASICS AND RELATED WORK

Early work on tactile displays appeared in 1957 by Geldard et al. [3] and was neatly summarized alongside newer work and general guidelines by Jones and Sarter [5]. In their review of research in the area they conclude that different levels of vibrotactile intensity and frequency are hard to distinguish and even interfere with each other, while stimulus location and duration are easier to identify and can thus achieve a higher bandwidth of communicated information.

The HapticHead prototype itself was introduced in prior work [8] and will not be fully described in this work as the specific arrangement of actuators and the actuator type is not central to this work. Essentially, HapticHead is a vibrotactile interface with a total of 24 actuators around the head in a sphere-like arrangement (see Fig. 1).

Seifi et al. [12] evaluated three possible interfaces which allow users to define vibrotactile patterns 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 of presets. Hong et al. define the term "demonstration-based authoring" as a method where a user taps on a screen and a system generates an accompanying TP which represents the user's taps [4]. The Hapticon Editor by Enriquez and MacLean [2] can be used to define a special waveform as a TP for a single actuator. VibScore, a musical score like design interface for TPs was introduced by Lee et al. [9]. The accompanying VibScoreEditor can be used to design a TP for a single actuator using musical notes.

Cuartielles et al. [1] developed a visual interface for designing TPs in a similar iterative fashion as this paper but they focused on only six actuators arranged in a simple way and did not properly visualize the feedback on the respective body part. The posVibEditor by Ryu and Choi [11] is closely related to this work, as their editor also aims to support TP design for multiple actuators and arbitrary waveforms. However, the posVibEditor lacks in selecting the correct actuator from a large number of possibilities and each actuator has to be defined individually.

Panëels et al. [10] published TactiPEd, a visual interface to easily prototype tactile patterns for multiple actuators. In their work they also summarized several other approaches and their strengths and weaknesses, including some of the aforementioned ones (see Fig. 3). TactiPEd is also closely related to this work as they aimed to provide an easy visual interface to define TPs for multiple actuators (see Fig. 4). However, while their interface still works for simple actuator arrangements with up to 6 or 8 actuators such as a wristband, it clearly lacks when trying to define TPs for more complex arrangements such as HapticHead due to the missing 3D aspect when selecting actuators and having to define each actuator individually.

ITERATIVE DESIGN PROCESS AND SOFTWARE IMPLEMENTATION

We started out with the TP design interfaces from related work in mind and wanted to improve upon them in terms of making it possible to define TPs for a much larger number and more complex arrangement of actuators. They should still be easily selectable as well as visually showing the affected body part and visualizing the generated tactile feedback as a heatmap. We were also influenced from commercial video editing software (Adobe Premiere) and modeling software (Blender) in terms of the choice of keyboard shortcuts for zooming and moving in the time domain and how to work with multiple actuators in a single timeline. Furthermore, we wanted our pattern design software, which we named 3DTactileDraw, to be accessible to non-technical users who are interested in TP design.

With these requirements in mind, we implemented a first prototype of 3DTactileDraw and subsequently did a first small user study with technical users in order to find the biggest weaknesses in our design and further improve it. A second, larger user study with users of non-technical background revealed more weaknesses and feature requests and led to our final prototype shown in Fig. 2.

We decided to implement two different kinds of interfaces (see sidebar and Fig. 2) due to trade-offs between usability and the ability to define each actuator individually.

Paint interface

The implemented paint interface is shown in Fig. 2a. In the paint interface we wanted to make it as easy as possible to generate a tactile pattern, especially for non-technical users. With a click on the model head, a user starts a new stroke. The user can then move the mouse on and around the head (the head turns by pressing the WASD keys). The generated pattern is captured in real-time and the user can add additional strokes to the same pattern which can have a selectable starting position with the red timeline slider on the lower part of the screen. Strokes are translated into vibrotactile actuation through a simple interpolation algorithm which actuates the four actuators closest to the stroke, depending on their distance in a linear fashion. The interface visually features a simulated live-view of the current actuation strength for the current point in time or while playing back a pattern using a heatmap shader.

Curve interface

The implemented curve interface is shown in Fig. 2b. It mimics common animation software and features a much higher degree of control over single actuators than the paint interface, as arbitrary waveforms can be defined for each actuator, including the possibility of linear, static, or smooth transitions between actuation points. It is also possible to define a pattern for multiple actuators at the same time by selecting them, overlapping their curves and pulling them at the same time. The curve interface features the same live-view of actuation strength as the paint interface.

Study 1

For this first study, we invited five participants with *technical* backgrounds (all male, mean age 27 years, SD = 3 years). We defined a set of tasks the users should perform without any further instructions and give verbal feedback on their thinking process (think-aloud method):

- Design two clearly defined TPs (e.g. “TP that moves from the forehead to the right ear, to the back of the head, to the left ear, and back to the start”).
- Design three purpose-driven TPs (e.g. “TP to make the user stop walking”).
- Design three TPs of their choice for a purpose of their choice.

Procedure. The participants filled out an informed consent form and introductory questionnaire and were subsequently made familiar with the HapticHead prototype before putting it on and adjusting it properly. Without further instructions the participants then performed the tasks outlined above and commented verbally on their progress. We captured a video of the participant performing the tasks and the 3DTactileDraw interface in order to evaluate their comments afterwards. In the end every participant filled out a final questionnaire.

Study 1 most important feature requests

- An option to invert the rotation direction of the 3D head.
- An option to undo and redo the last step.
- Right-click and hold to turn the 3D model of the head.
- an option to define intensity in the paint mode instead of always maximum intensity. This was not implemented as it is not possible to set the stroke width (intensity) while drawing the stroke intuitively with just mouse and keyboard.
- A warning when changing the mode from curve mode to the paint mode that all data gets lost in the process (no conversion possible, see Limitations).

Results and discussion of study 1. We gathered a variety of feature requests, things that should be fixed and general feedback to the prototype. The most important feature requests are shown in the sidebar. Participants generally slightly preferred the paint mode but could see value in the curve mode as well, depending on which specific pattern they wanted to design. They agreed that (a) designing patterns is easy in both interface types, (b) the interfaces are easy to navigate, (c) the live view of the TP resembles the feeling on the HapticHead and (d) that the design interface could also be used to design TPs for other body parts or the entire body.

Several other feature requests and bug fixes were also analyzed and implemented in a second version of the prototype before starting the second study.

Study 2

In the second study, we invited 12 participants with *non-technical* backgrounds (7 women, 5 men, mean age 45 years, SD = 17 years). Just like in study 1, we defined a set of tasks (see sidebar). This time, we left out the clearly defined TPs and included more purpose-driven TPs instead. The procedure of study 2 was the same as in study 1, including the think-aloud method.

- Design eight purpose-driven TPs (e.g. "TP simulate a spider moving on the head" or "TP that signalizes a staircase"). For two of these TPs, participants were asked to use the curve interface in order to get our non-technical participants to work with the more complicated interface type.
- Design two TPs of their choice for a purpose of their choice.

Results and discussion of study 2. As we already gathered the most important feature requests in the first study, the participants of the second study did not miss any important feature except for the missing conversion process between the two interface types. We expected the non-technical participants of the second study to like the paint interface much more than the curve interface but surprisingly they also liked to use the curve interface, which was used 45 % of the time despite being only incentivised verbally in two of the ten TP design trials. The results to the general questions were very similar to study 1 with general agreement to the four statements mentioned in the results of study 1. In the second study, we also asked specific questions for each interface type. Participants generally agreed that in the **paint interface** strokes were easy to draw, it was clear how to remove a stroke, the interface was intuitive and all TPs could be created in a quickly. The participants had mixed opinions on whether all TPs could be designed with high precision. The **curve interface** on the other hand was rated as a bit less intuitive but more precise. General agreement was also given to the statements that it is clear how to hide and show curves, and how to manipulate them.

Due to the nature of a think-aloud study, some of the non-technical participants were fiddling around a bit with the interface in the beginning, so some kind of tutorial or introduction might be useful in a future release.

Limitations of the current prototype

There is no easy conversion possible between the two interface types as TPs are defined very differently. The curve interface defines specific waveforms for every single actuator while the paint mode defines strokes to be interpolated between actuators using an algorithm such as the one in [8]. A conversion would theoretically be possible from paint to curve interface with some loss of information but the other way around would require the extraction of possible strokes from curve data which can have multiple solutions and a lot of strokes, so this is not feasible.

CONCLUSION

In this work we present a first implementation of two possible design interfaces for TPs, which can work with a large number of actuators in complex arrangements while still remaining intuitive to use. Especially the paint interface is very different from previous works in this domain and was received quite well by the participants in our studies.

We are already working on a virtual reality version of 3DTactileDraw, which will allow setting the stroke width (intensity) while drawing the stroke by using the HTC Vive trigger button. It will presumably also allow a more intuitive movement around the model head while drawing strokes.

REFERENCES

[1] David Cuartielles, A. Göransson, T. Olsson, and S. Stenslie. 2012. Developing Visual Editors for High-Resolution Haptic Patterns. *The Seventh International Workshop on Haptic and Audio Interaction Design* (2012), 42–44.

[2] M. J. Enriquez and K. E. MacLean. 2003. 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*. IEEE Comput. Soc, 356–362. https://doi.org/10.1109/HAPTIC.2003.1191310

[3] Frank A. Geldard. 1957. Adventures in tactile literacy. *American Psychologist* 12, 3 (1957), 115–124.

[4] Kyungpyo Hong, Jaebong Lee, and Seungmoon Choi. 2013. Demonstration-based vibrotactile pattern authoring. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction - TEI '13*. ACM Press, New York, New York, USA, 219. https://doi.org/10.1145/2460625.2460660

[5] Lynette A. Jones and Nadine B. Sarter. 2008. Tactile Displays: Guidance for Their Design and Application. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 50, 1 (feb 2008), 90–111. https://doi.org/10.1518/001872008X250638

[6] Oliver Beren Kaul, Kevin Meier, and Michael Rohs. 2017. *Increasing Presence in Virtual Reality with a Vibrotactile Grid Around the Head*. Springer International Publishing, Cham, 289–298. https://doi.org/10.1007/978-3-319-68059-0_19

[7] Oliver Beren Kaul and Michael Rohs. 2016. 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*. ACM Press, New York, New York, USA, 2533–2539. https://doi.org/10.1145/2851581.2892355

[8] Oliver Beren Kaul and Michael Rohs. 2017. 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*. ACM Press, New York, New York, USA, 3729–3740. https://doi.org/10.1145/3025453.3025684

[9] Jaebong Lee, Jonghyun Ryu, and Seungmoon Choi. 2009. Vibrotactile score: A score metaphor for designing vibrotactile patterns. In *Proceedings - 3rd Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, World Haptics 2009*. IEEE, 302–307. https://doi.org/10.1109/WHC.2009.4810816 arXiv:arXiv:1011.1669v3

[10] Sabrina Panëels, Margarita Anastassova, and Lucie Brunet. 2013. TactiPEd: Easy Prototyping of Tactile Patterns. Springer, Berlin, Heidelberg, 228–245. https://doi.org/10.1007/978-3-642-40480-1_15

[11] Jonghyun Ryu and Seungmoon Choi. 2008. posVibEditor: Graphical authoring tool of vibrotactile patterns. In *2008 IEEE International Workshop on Haptic Audio visual Environments and Games*. IEEE, 120–125. https://doi.org/10.1109/HAVE.2008.4685310

[12] Hasti Seifi, Chamila Anthonypillai, and Karon E. MacLean. 2014. End-user customization of affective tactile messages: A qualitative examination of tool parameters. In *2014 IEEE Haptics Symposium (HAPTICS)*. IEEE, 251–256. https://doi.org/10.1109/HAPTICS.2014.6775463