VIREO: Web-based Graphical Authoring of Vibrotactile Feedback for Interactions with Mobile and Wearable Devices

Mihail Terenti, Radu-Daniel Vatavu

VIREO: Web-based Graphical Authoring of Vibrotactile Feedback for Interactions with Mobile and Wearable Devices

Mihail Terenti and Radu-Daniel Vatavu

MintViz Lab, MANSiD Research Center, Ştefan cel Mare University of Suceava
13 Universitatii, 720229 Suceava, Romania

ARTICLE HISTORY
Compiled September 17, 2022

ABSTRACT
We introduce VIREO, a web-based software tool for graphical authoring of vibrotactile feedback for mobile and wearable applications. VIREO enables flexible specification of vibrotactile patterns with model-based and free-draw sketch input, and is compatible with mobile and wearable devices that support JavaScript, either natively at the platform level or in a web browser. We demonstrate the practical utility of VIREO by presenting several applications implemented for a smartphone, smartwatch, smart armband, and a pair of smartglasses that integrate vibrotactile feedback patterns authored with VIREO. Also, we present the results of an usability evaluation study involving sixteen participants represented by coders with various programming experience and levels of programming language proficiency, including JavaScript. We discuss our contribution in the context of the results of a Systematic Literature Review that we conducted on the topic of software tools, editors, and platforms developed in the scientific community for authoring vibrotactile feedback. Since one major finding of our review is the little availability of such contributions, we release VIREO as a free web resource for researchers and practitioners that wish to author and integrate vibrotactile feedback in mobile and wearable applications.

KEYWORDS
Vibrotactile feedback; mobile devices; wearable devices; web tool; JavaScript

1. Introduction

Smart mobile devices, such as smartphones, tablets, and phablets, have become dominant in the landscape of personal computing devices, enabling interaction on the go and rich possibilities for communication, access to information, and interactions in mixed reality environments. Also in this landscape, smart wearables, such as smartwatches, smartglasses, and digital jewellery, represent the fastest-growing technological innovation for mobile users (Menear, 2020), providing new functionality and services complementary to the prevalent smartphone (Chen, Chen, Liu, Chen, and Li, 2021; Sani, Boos, Yun, and Zhong, 2015), but also likely to replace smartphones in the foreseeable future. Mobile and wearable devices enable a diversity of output modalities for users, from primarily visual feedback on smartphones and smartwatches (Matulic, Ganeshan, Fujiwara, and Vogel, 2021; Wenig, Schöning, Olwal, Oben, and Malaka, 2017) to aural feedback for devices worn on the head (Brezolin, Santos, de Lima, Zanella, Rieder,

CONTACT Radu-Daniel Vatavu. Email: radu.vatavu@usm.ro

and De Marchi, 2017; Casamassima, Ferrari, Milosevic, Rocchi, and Farella, 2013) to haptic feedback for fitness trackers and electronic rings (Han, Han, Annett, Anderson, Huang, and Yang, 2017; Turmo Vidal, Zhu, Waern, and Márquez Segura, 2021; Vatavu, Mossel, and Schönauer, 2016). However, the choice of the output modality also depends on the context in which such devices are used and worn. For instance, in some contexts of use, interaction with mobile and wearable devices needs to be subtle (Pohl, Muresan, and Hornbæk, 2019), non-disturbing to the user and the others (Anderson, Grossman, Wigdor, and Fitzmaurice, 2015), while nevertheless retaining highly informative and effective qualities despite temporary situations and interaction-related constraints determined by the context of use (Heller, Vanacken, Geurts, and Luyten, 2020; Wobbrock, 2019).

Among the possible feedback modalities available on mobile and wearable devices, *haptic feedback* enables a wide range of possible implementations that target various receptors on the human body—in the form of tactile, thermal, or force feedback (Basdogan, Giraud, Levesque, and Choi, 2020; Jones, 2016; Kang and Lee, 2018; Song, Lim, and Yun, 2016; Wang, Guo, Liu, Zhang, Xu, and Xiao, 2019; Youn, Lee, Kim, Shim, Chan, and Lee, 2021)—and has been found effective for contexts of use characterized by constraints such as those enumerate above. *Vibrotactile feedback,* which falls under the general category of haptics, is implemented via controlled vibrations targeting specific cutaneous mechanoreceptors located in the human skin. Compared to other haptic modalities, vibrotactile feedback is widely available on a diversity of mobile and wearable devices that integrate vibration motors, e.g., from the buzz of an incoming call on a smartphone or smartwatch (Hong, Lee, and Choi, 2013; Israr, Zhao, and Schneider, 2015; Park and Choi, 2017; Schönauer, Mossel, Zaiti, and Vatavu, 2015) to the rumble of a game controller (Swindells, Pietarinen, and Viitanen, 2014) to a hobbyist's custom-made wearable prototypes (Terenti and Vatavu, 2022), which makes it an accessible modality for practitioners to use in their applications.

Several options are available to design, implement, and integrate vibrotactile feedback into applications, among which two main trends can be identified: (i) specifying vibrotactile feedback programmatically at code level and (ii) using authoring tools and editors to design vibration patterns, which are exported to applications. For instance, Apple's Core Haptics[1] enables implementation of haptic patterns that are highly customizable, but need to be specified in code via API calls, and Haptrix[2] is an example of a graphical editor that leverages Core Haptics to make the developers' task easier in that regard. Unfortunately, some of these solutions target specific platforms and/or operating systems, as these two examples show, which limits the range of supported devices. Other tools address a wider range of platforms, operating systems, and programming languages, but specialize on particular haptic synthesis or rendering technology. For instance, Lofelt Studio[3] enables iOS and Android developers to design, test, and integrate tactile effects for mobile devices by deriving haptic clips from audio files. However, the audio file must be available to start off haptics design. Unlike Lofelt, Syntacts[4] enable design of rich vibrotactile patterns from scratch with a powerful editor and APIs for C++, C#, Python, and Unity developers, but the results are rendered via audio interfaces, which limits the range of addressable target devices

[1]Core Haptics — Apple Developer Documentation, https://developer.apple.com/documentation/corehaptics.
[2]Haptrix - Create & Share Haptic Experiences, https://www.haptrix.com.
[3]Lofelt - Unlock the power of haptics in mobile devices, https://lofelt.com.
[4]Syntacts — The Tactor Synthesizer, https://www.syntacts.org.

to desktop platforms.[5] More flexible tools that enable authoring of vibrotactile patterns in a way that is not platform-, synthesis-, and rendering technology-dependent have been demonstrated in the scientific literature (Hong et al., 2013; Lee and Choi, 2012; Lee, Ryu, and Choi, 2009a; Panëels, Anastassova, and Brunet, 2013; Ryu and Choi, 2008) but, unfortunately, they are not publicly available.

In this context, new initiatives and corresponding tools are needed in the community to democratize design of rich and expressive vibrotactile feedback patterns and their integration in applications addressing a wide range of mobile and wearable devices, including new devices with diverse characteristics, development platforms, and operating systems to be released in the foreseeable future. Since such devices are expected to feature web connectivity in order to access cloud services (Chen, Ma, Li, Wu, Zhang, and Youn, 2017), integrate with IoT (Furini, Mirri, Montangero, and Prandi, 2020), or simply to retrieve and share data (Schneegass, Poguntke, and Machulla, 2019), we argue that web technology in the form of programming languages, communication protocols, and data formats designed for the web[6,7] can be leveraged as an all-inclusive medium for both vibrotactile feedback design *on the web* and rendering of vibrations *on a target device*. Such an approach will democratize both the process of designing vibrotactile feedback patterns, including their management and sharing as readily accessible online collections, but also implementation on a wide range of mobile and wearable devices that support web applications natively or in a web browser. In this context, we make the following practical contributions in this paper:

(1) We conduct a Systematic Literature Review (SLR) on the topic of software authoring tools, editors, and platforms developed in the scientific community for vibrotactile feedback design in order to understand the extent of these contributions, their authoring techniques, target platforms, and addressed devices. Our findings, based on the analysis of twenty-five software tools and corresponding scientific papers published at peer-reviewed academic venues, show that most of the tools have targeted desktop environments and, arguably because of that, very few are still publicly available.
(2) We introduce VIREO (VIbrotactile Online EditoR, see Figure 1 for a screenshot), our software tool for authoring vibrotactile feedback patterns on the web with several convenient features for practitioners: model-based authoring of vibrotactile patterns, pattern composition via direction manipulation, pattern authoring via free-draw sketching and user demonstration, management of pattern libraries on the web, and easy integration with mobile and wearable devices that are connected to the web. We discuss technical implementation details of VIREO, present user workflows, and describe the techniques available in VIREO for authoring vibrotactile patterns.
(3) We demonstrate the practical utility of VIREO by presenting applications de-

[5]Although adaptation to mobile platforms might be possible; see FAQ, Can Syntacts be used for mobile haptic applications?, `https://www.syntacts.org/faq`. However, Syntacts was primarily intended for desktop-based research and applications (see link above).

[6]For example, wearables running Tizen can be programmed with just HTML, CSS, and JavaScript. Tizen Studio enables development of web applications for mobile, wearable, and TV devices, which consist of HTML, JavaScript, and CSS combined in a package deployed to a Tizen device, such as Samsung Galaxy smartwatches; see `https://docs.tizen.org/application/web/index`.

[7]Vibration API (`https://developer.mozilla.org/en-US/docs/Web/API/Vibration_API`) enables web applications to access the vibration hardware from mobile and wearable devices as long as the patterns can be described as series of on and off pulses. This functionality is limited, but via pulse-width modulation (PWM), it enables many haptic effects for web applications running JavaScript, e.g., see `http://www.hapticsjs.org` or `https://npm.io/package/vtp.js`.

Figure 1. Our tool, VIREO, freely available at `https://vireoapp.com` enables graphical authoring of vibrotactile feedback patterns for augmenting interactions on mobile and wearable devices that run JavaScript, either as standalone applications or in a web browser. VIREO allows researchers and practitioners to specify model-based vibrotactile patterns, e.g., exponential or harmonic waves, or simply to sketch a graphical pattern to start off their designs, and then to deploy those designs to a target device. *Note:* we illustrate in this picture a vibration pattern with comfortable, smooth, and soft emotional connotations (Seifi et al., 2015).

veloped for smartphone, smartwatch, smart armband, and smartglasses devices integrating vibrotactile feedback patterns authored with VIREO.

(4) We conduct an usability evaluation study with VIREO and sixteen participants representing coders with various levels of programming experience in various programming languages and platforms. Our findings show a good overall usability for authoring vibrotactile feedback patterns with VIREO, confirmed by the SUS (70.47) and CSUQ (2.68) usability tests, respectively.

2. Related Work

We relate to prior work describing applications, tools, and graphical editors designed to assist researchers and practitioners in authoring and implementing vibrotactile feedback. To this end, we conduct a Systematic Literature Review (SLR) study to identify such prior work and to perform an overview of the software tools that have been developed in the scientific community, and in particular tools that are publicly available. For the design of our SLR, we followed Siddaway, Wood, and Hedges (2019) for best practice recommendations and adopted the PRISMA method (Liberati, Altman, Tetzlaff, Mulrow, Gøtzsche, Ioannidis, Clarke, Devereaux, Kleijnen, and Moher, 2009) for the *identification*, *screening*, *eligibility*, *inclusion*, and *snowballing* steps. In the following, we describe these steps in detail.

2.1. Study Design and Implementation

2.1.1. Identification

This step consists in identifying work relevant to our topic of interest by specifying keywords, search operators, and electronic databases where to search for those keywords (Siddaway et al., 2019). Our focus is on software tools that enable authoring vibrotactile feedback patterns for mobile and wearable interactions. However, even though we are interested in mobile and wearable devices only, we considered the scope of our SLR to be much broader, as we figured that mobile and especially wearable devices have been little addressed by such tools. Regarding the relevant venues where such research has been traditionally disseminated, we considered SIGCHI conferences, available through the ACM Digital Library, and IEEE specialized venues, such as the Haptics Symposium, World Haptics, and Transactions on Haptics, available from the IEEE Xplore scientific database. We searched keywords in the abstracts of the papers available from these two databases since searching through the entire text resulted in many nonrelevant records, while searching just in the paper titles resulted in missing relevant work. Based on these considerations, we ran the following query:

```
Abstract: (vibrotactile AND (feedback OR pattern*) AND (tool* OR
platform* OR author*)))
```

that returned 18 results from ACM DL and 52 results from IEEE Explore.[8]

2.1.2. Screening

We screened the abstracts of these papers to evaluate their relevancy for our topic of interest regarding software tools for vibrotactile feedback. We removed 43 papers, most of which described applications that employed vibrotactile feedback, for which the keywords from our query were used in another context, e.g., "surgical tools" in the visuohaptic bone saw simulator of Olsson, Nysjö, Singh, Thor, and Carlbom (2015) or "operating power tools" in the exploration of Yin, Otis, Fortin, and Cooperstock (2019) of assembly tasks in VR with a simulated impact wrench. After the screening step, we arrived at a number of 27 papers.

2.1.3. Eligibility and Inclusion

We scrutinized the remaining papers by reading the entire text in order to identify tools, editors, platforms, or applications developed for authoring vibrotactile patterns. To this end, we formulated the following eligibility criteria: (EC_1) the content of the paper must be available, in English, and the content was peer-reviewed, (EC_2) the paper is about vibrotactile feedback, and (EC_3) the paper describes a tool, in any form, that enables authoring of vibrotactile feedback patterns for integration in applications. By applying these criteria, we removed 14 references,[9] leaving a subset of 13 papers relevant to our topic of investigation.

[8]An adapted form of the query was used in IEEE Xplore due to the specifics of the IEEE Xplore search engine, but with the same keywords and operators: "Abstract": vibrotactile AND ("Abstract": feedback OR "Abstract": pattern*) AND ("Abstract": tool* OR "Abstract": platform* OR "Abstract": author*).

[9]We also removed (Lee and Choi, 2012) that described VibScoreEditor, a tool relevant to our scope, but already presented in a previous paper (Lee et al., 2009a) by the same authors.

Table 1.: Authoring tools for vibrotactile feedback identified in our Systematic Literature Review (SLR) study. *Notes:* different colors depict different goals, target devices, and authoring techniques for these tools. The last row shows our tool, VIREO, with distinct characteristics in this landscape.

	Reference	Tool name	Objective of the tool/research†	Authoring platform‡	Target device§	Authoring technique¶	Avail-able?
1	Enriquez and MacLean (2003)	Hapticon Editor	Haptic icon design	Desktop	Terrain display	Composition of predefined patterns User demonstration	–
2	Swindells, Maksakov, MacLean, and Chung (2006)	Haptic Icon Prototyper	Haptic icon design	Desktop	Generic, not specified	Composition of predefined patterns Waveform editing	–
3	Eid, Andrews, Alamri, and El Saddik (2008)	HAMLAT	Enhancing immersive experiences *(extension of a VR graphics engine with haptics)*	Desktop	VR devices	Direct manipulation *(edit haptic properties-stifness, friction)*	–
4	Ryu and Choi (2008); Lee et al. (2009a)	posVib Editor	Generic-purpose vibrotactile pattern authoring	Desktop	Multi-actuator devices	Composition of predefined patterns	–
5	Park, Hwang, and Hwang (2009)	Vibration Composer	Software tool recommendations for designing vibrotactile patterns	Desktop	Handheld	Composition of predefined patterns	–
6	Lee and Choi (2012); Lee et al. (2009a); Lee, Ryu, and Choi (2009b)	VibScore Editor	Generic-purpose vibrotactile pattern authoring	Desktop	Generic, not specified	Concept of "vibrotactile score" Composition of predefined patterns	–
7	Kim, Cha, Ryu, and Oakley (2010)	–	Enhancing immersive experiences *(tactile video)*	Desktop	Custom glove	Direct manipulation *(map vibrations to video with a software brush)*	–
8	Park, Choi, Hwang, Kim, Sa, and Joung (2011)	–	Enabling virtual buttons for mobile touchscreen devices	Mobile	Mobile devices	Select from library *(72 records)*	–
9	Hong et al. (2013)	New Vibration Editor	Generic-purpose vibrotactile pattern authoring	Mobile	Mobile devices	User demonstration *(pressure input on touchscreen)*	–
10	Panëels et al. (2013)	TactiPEd	Generic-purpose vibrotactile pattern authoring	Desktop	Multi-actuator devices	Waveform editing	–
11	Swindells et al. (2014)	ViviTouch Studio	Prototyping mixed audio, video, and vibrotactile feedback	Desktop	Game controllers	Composition of predefined patterns	–
12	Martínez, García, Oliver, Molina, and González (2014)	VITAKI	Enhancing immersive experiences (VR)	Desktop	VR devices	Composition of predefined patterns Waveform editing	–
13	Schneider and MacLean (2014)	mHIVE	Collaborative manipulation and exploration of haptic sensations	Mobile	Generic, not specified	Composition of predefined patterns Waveform editing	–
14	Dong, Gao, Al Osman, and El Saddik (2015)	Web MPEGV Haptic Authoring	Enhancing immersive experiences *(MPEG-V)*	Web	VR devices	Direct manipulation *(edit haptic properties-stifness, friction)*	–
15	Israr et al. (2015)	Feel Messenger	Enhancing social and interpersonal communication	Mobile	Mobile devices	Select from library	–
16	Seifi et al. (2015)	VibViz	Taxonomy and navigation for vibrotactile library	Web	Generic, wristband	Select from library *(120 records)*	Yes [link]
17	Schneider, Israr, and MacLean (2015)	Mango	Enabling professional animators to design for vibrotactile grids	Desktop	Vibrotactile arrays	Animation tools *(timeline, keyframes, object paths)*	–
18	Le, Zhu, Kosinski, Fjeld, Azh, and Zhao (2016)	Ubitile	Vibrotactile feedback for interactive tabletops	Ring & tabletop	Ring & tabletop	User demonstration *(movements of the finger wearing a ring)*	–
19	Schneider and MacLean (2016)	Macaron	Generic-purpose vibrotactile pattern authoring	Web	Generic, not specified	Waveform editing	Yes [link]
20	Huang, Chan, Jian, Chang, Chen, Yang, Hung, and Chen (2016)	VibroPlay	Enhancing immersive experiences (VR)	VR (HMD)	Chair with actuators	Direct manipulation *(switch virtual actuators on/off in VR)*	–
21	Boer, Vallgårda, and Cahill (2017)	Hedonic Haptic player	Exploration of "aesthetic form-giving" in interaction design	Desktop	Custom wearable	Composition of predefined patterns	–
22	Park and Choi (2017)	PhysVib	Enhancing immersive experiences *(extension of a video game physics engine with haptics)*	Mobile	Mobile devices	Automatic generation *(exponentially-decaying sinusoidal model)*	Yes [link]
23	Clark, Schneider, MacLean, and Tan (2017)	MacaronMix	Generic-purpose vibrotactile pattern authoring	Web	Generic, not specified	Automatic generation *(vibrotactile morphing with DTW)*	Yes [link]
24	Danieau, Guillotel, Dumas, Lopez, Leroy, and Mollet (2018)	HFX Studio	Enhancing immersive experiences (VR)	Desktop	Generic, not specified	Haptic perceptual models	–
25	Pezent, Cambio, and O'Malley (2021)	Syntacts	Generic-purpose vibrotactile pattern authoring Audio-controlled haptics	Desktop	Vibrotactile arrays	Composition of predefined patterns Waveform editing Numerical models	Yes [link]
	This paper	VIREO	Generic-purpose vibrotactile pattern authoring for Specific applications in mobile and wearable computing	Web	Mobile and wearable devices supporting web standards and JavaScript	Pattern models Waveform editing Composition of patterns Direct manipulation *(pattern composition)* User demonstration *(free draw input)*	Yes [link]

† *Legend:* generic tools (7) immersive environments/media (6) specific applications, devices, or user groups (8) research-oriented explorations (3).

‡ *Legend:* web (4) desktop (14) mobile (5) specific device (2).

§ *Legend:* mobile devices (5) VR and game controllers (4), custom device (9), generic, not always specified (7).

¶ *Legend:* composition of patterns (10) library-based (3) waveform editing (6) user demonstration (3) model-based (4) specific techniques (5).

2.1.4. Snowballing

We browsed the list of references of the eligible papers and identified 8 new papers relevant to our scope (a *backward snowballing* procedure) from a total of 591 references. We also looked at the Google Scholar citations of the papers from our set and identified 4 more papers from a total of 676 citations (a *forward snowballing* procedure). Our final set is composed of 25 papers and corresponding tools for authoring vibrotactile feedback; see Table 1 for an overview.

2.2. Results

The tools for authoring vibrotactile feedback identified in our SLR were introduced for a diversity of reasons and a variety of objectives involving vibrotactile feedback as well as for a variety of target platforms and devices. In our analysis, we grouped this prior work according to several criteria; see the next subsections and the columns of Table 1.

2.2.1. Objective of the tool

We identified four main reasons to introduce tools for authoring vibrotactile feedback in the scientific community. In decreasing order, these are: (1) to support new user experiences in immersive media and environments (30.8%) (Danieau et al., 2018; Dong et al., 2015; Eid et al., 2008; Huang et al., 2016; Israr et al., 2015; Kim et al., 2010; Martínez et al., 2014; Park and Choi, 2017), (2) to address specific devices or specific professional user groups (30.8%) (Enriquez and MacLean, 2003; Le et al., 2016; Park et al., 2011; Pezent et al., 2021; Schneider et al., 2015; Schneider and MacLean, 2014; Swindells et al., 2006, 2014), (3) generic tools to assist with authoring vibrotactile feedback (26.9%) (Clark et al., 2017; Hong et al., 2013; Lee and Choi, 2012; Lee et al., 2009a; Panëels et al., 2013; Pezent et al., 2021; Ryu and Choi, 2008; Schneider and MacLean, 2016), and (4) research-oriented exploration of vibrotactile feedback that, as part of the underlying scientific method, required new tools (11.5%) (Boer et al., 2017; Park et al., 2009; Seifi et al., 2015). For instance, Huang et al. (2016) introduced VibroPlay, a tool for authoring 3D tactile effects for immersive media experiences, that implemented authoring via direct manipulation in VR: the haptic designer touches virtual replicas of actual actuators, reproduced in the VR environment, by switching them on and off to produce the desired effect. Lee et al. (2009a) and Lee and Choi (2012) introduced and evaluated the VibScore Editor, an application developed for generic-purpose vibrotactile pattern authoring that employs the concept of a "vibrotactile score," implemented with graphical symbols of musical notes. Le et al. (2016) were interested in enhancing applications for interactive tabletops with vibrotactile feedback on the user's hand. To this end, they described an authoring technique based on user demonstration: movements of the finger wearing a ring with built-in IMU sensing were mapped to vibration duration and intensity, respectively. Boer et al. (2017) is an example of a scientific examination where a new tool for vibrotactile feedback was needed to enable "different forms of pleasurable experiences targeting different moods and situations of the wearer" (p. 911). The authors explored and discussed aspects of aesthetic form-giving in interaction design, which they evaluated with a custom wearable device and application for creating vibrotactile feedback for that device. VibViz (Seifi et al., 2015) was introduced as both a taxonomy and tool to assist designers in the exploration of haptic collections with "easy and highly navigable access to large, diverse sets of vibrotactile stimuli, on the premise that multiple access pathways facilitate discovery and engagement" (p. 254).

2.2.2. Authoring platforms

We found that the majority of the tools introduced in the scientific community addressed desktop platforms (56.0%) followed by mobile (20.0%), web (16.0%) (Clark et al., 2017; Dong et al., 2015; Schneider and MacLean, 2016; Seifi et al., 2015), and tools addressing specific devices (8.0%) (Huang et al., 2016; Le et al., 2016). Of these, tools for the web are the easiest to access by practitioners from various platforms and operating systems. For example, Schneider and MacLean (2016) introduced Macaron, a vibrotactile effect editor on the web implementing gallery examples, composability of patterns via copy and paste, and visibility of the pattern parameters. Clark et al. (2017) delivered an update, MacaronMix, featuring automatic generation of vibrotactile effects with Dynamic Time Warping as part of a morphing process where parent signals are blended to create new patterns. Seifi et al. (2015) presented VibViz, a web application for navigating a library of 120 vibrotactile patterns with associated sensory, emotional, and metaphorical connotations. Dong et al. (2015) developed the Web MPEGV Haptic Authoring tool for multimedia applications to enable developers to edit haptic properties for 3D objects on the web.

2.2.3. Target devices

The tools analyzed in our SLR were developed for vibrotactile feedback addressing custom devices (36.0%) (Boer et al., 2017; Enriquez and MacLean, 2003; Huang et al., 2016; Kim et al., 2010; Le et al., 2016; Panëels et al., 2013; Pezent et al., 2021; Ryu and Choi, 2008; Schneider et al., 2015), mobile devices (20.0%), and VR and game controllers (16.0%) (Dong et al., 2015; Eid et al., 2008; Martínez et al., 2014; Swindells et al., 2014), while a percent of 28.0% of the papers did not specify the target device (Clark et al., 2017; Danieau et al., 2018; Lee and Choi, 2012; Lee et al., 2009a,b; Schneider and MacLean, 2014, 2016; Seifi et al., 2015). For example, Le et al. (2016) were interested in vibrotactile feedback delivered on the user's finger with a ring during interactions with tabletops, Swindells et al. (2014) targeted game controllers, and Eid et al. (2008), Martínez et al. (2014), and Dong et al. (2015) targeted VR devices. A few tools (Hong et al., 2013; Israr et al., 2015; Park and Choi, 2017; Park et al., 2011) explicitly addressed mobile devices, while wearables were targeted by three tools identified by our SLR: VibViz with a wristband (Seifi et al., 2015), tactile video with a glove (Kim et al., 2010), and the Hedonic Haptic Player (Boer et al., 2017) delivering vibrotactile feedback at various locations on the user's body in the context of exploring aesthetic interaction design.

2.2.4. Authoring techniques

We found several types of authoring techniques implemented by the tools that we analyzed, as follows: composition of patterns (32.3%), waveform editing (19.3%) (Martínez et al., 2014; Panëels et al., 2013; Pezent et al., 2021; Schneider and MacLean, 2014, 2016; Swindells et al., 2006), use of numerical models to specify patterns (12.9%) (Clark et al., 2017; Danieau et al., 2018; Park and Choi, 2017; Pezent et al., 2021), library of patterns (9.7%) (Israr et al., 2015; Park et al., 2011; Seifi et al., 2015), authoring by means of user demonstration (9.7%) (Enriquez and MacLean, 2003; Hong et al., 2013; Le et al., 2016), and techniques specific to various target devices and/or addressing professional user groups (16.1%) (Dong et al., 2015; Eid et al., 2008; Kim et al., 2010; Lee and Choi, 2012; Lee et al., 2009a; Schneider et al., 2015). For example, Hong et al. (2013) implemented user demonstration of vibrotactile pattern authoring via pressure input on touchscreens. Park and Choi (2017) presented a technique for the automatic generation of vibrotactile effects for a video game physics engine based on an exponentially-decaying sinusoidal model. Martínez et al. (2014) and Pezent et al. (2021) presented tools featuring waveform editing and composition of predefined patterns. Schneider et al. (2015) introduced a new authoring technique, specifically addressing professional animators, which employed animation tools, such as the timeline, key frames, and object paths, to specify vibration patterns for vibrotactile arrays. Other tools, such as posVibEditor (Ryu and Choi, 2008), were specifically designed for a wide range of practitioners with a diversity of skills and experience with haptics, i.e., "posVibEditor [is a tool] for quick and easy design of vibrotactile patterns for vibration motors. The tool supports the drag-and-drop design paradigm so that novice users can easily learn and interact with the tool" (p. 120), while ViviTouch Studio (Swindells et al., 2014) was specifically developed for rapid prototyping. Some papers also included comparative evaluations of several authoring techniques. For instance, Huang et al. (2016) argued that haptic design based on sound waves did not transfer well into tactile sensations for practitioners without sufficient expertise, and proposed an authoring technique based on direct manipulation of virtual actuators in VR for a physical chair with actuators. Hong et al. (2013) compared user demonstration with conventional waveform-based authoring and reported comparable or superior performance of the former technique.

2.2.5. Availability of tools

Of all the twenty-five tools included in our analysis, only five are publicly available (Clark et al., 2017; Park and Choi, 2017; Pezent et al., 2021; Schneider and MacLean, 2016; Seifi et al., 2015), of which two, Macaron (Schneider and MacLean, 2016) and MacaronMix (Clark et al., 2017), share the same web application; see Table 1 for the web links. PhysVib (Park and Choi, 2017) is an extension for a video game physics engine that enables integration of immersive experiences in mobile video games via automatic generation of haptic effects, available on github. Macaron and MacaronMix (Clark et al., 2017; Schneider and MacLean, 2016) implement waveform editing and automatic generation of vibrotactile effects by blending existing patterns. VibViz (Seifi et al., 2015) enables access to a library of 120 vibrotactile patterns with corresponding sensory, emotional, and metaphorical mappings. And Syntacts (Pezent et al., 2021) is a haptic rendering framework for desktop platforms with audio interfaces.

2.2.6. Summary

The results of our SLR revealed a variety of useful and creative techniques for authoring vibrotactile feedback on a variety of platforms and for a diversity of target devices, including mobile and wearable devices according to our scope. Unfortunately, the tools featuring those techniques are not publicly available or, when they are, are highly specialized, such as the audio haptics from by Syntacts (Pezent et al., 2021) that are available for desktop platforms only. These findings suggest that further efforts are needed in the scientific community to make software tools readily available to researchers and practitioners in order to foster new scientific discoveries and practical developments involving vibrotactile feedback for mobile and wearable interactions. Regarding the latter, our SLR revealed a few tools targeting mobile platforms (Hong et al., 2013; Israr et al., 2015; Park and Choi, 2017; Park et al., 2011; Schneider and MacLean, 2014), but limited work for wearable platforms, for which only specific devices were targeted, such as a custom pair of gloves (Kim et al., 2010), electronic ring (Le et al., 2016), wristband (Seifi et al., 2015), and the Hedonic Haptic player device (Boer et al., 2017), respectively. However, off-the-shelf wearable devices integrate vibration motors and readily support web standards, protocols, and data formats just like the more complex mobile platforms. Therefore, we introduce VIREO, our tool that is freely available on the web, which capitalizes on the results of our SLR and features many of the useful authoring techniques that we identified in the scientific literature; see the last row of Table 1. Specifically, VIREO implements waveform editing (Martínez et al., 2014; Panëels et al., 2013; Pezent et al., 2021; Schneider and MacLean, 2014, 2016; Swindells et al., 2006), composition of patterns (Boer et al., 2017; Enriquez and MacLean, 2003; Lee et al., 2009a; Martínez et al., 2014; Park et al., 2009; Pezent et al., 2021; Ryu and Choi, 2008; Schneider and MacLean, 2014; Swindells et al., 2006, 2014), direct manipulation (Dong et al., 2015; Eid et al., 2008; Huang et al., 2016; Kim et al., 2010), and vibrotactile feedback authoring via user demonstration (Hong et al., 2013; Le et al., 2016). Just like posVibEditor (Lee et al., 2009a; Ryu and Choi, 2008), VibScoreEditor (Lee and Choi, 2012), New Vibration Editor (Hong et al., 2013), TactiPEd (Panëels et al., 2013), Macaron and MacaronMix (Clark et al., 2017; Schneider and MacLean, 2016), and Syntacts (Pezent et al., 2021), VIREO implements generic-purpose vibrotactile pattern authoring, but addresses mobile and wearable devices specifically. Also, just like VibViz (Seifi et al., 2015), Macaron and MacaronMix (Clark et al., 2017; Schneider and MacLean, 2016), and Web MPEGV (Dong et al., 2015), VIREO runs on the web. In the next section, we present VIREO in detail.

3. VIREO

We introduce VIREO, our software tool for authoring vibrotactile feedback patterns on the web and integration in mobile and wearable applications. We start with the design requirements for VIREO, informed by the results of our SLR study and a brainstorming session about requirements for authoring tools on the web, and continue with technical implementation details and a description of the main features.

3.1. Design Requirements for VIREO

Based on our analysis of the scientific literature and our goals and scope regarding vibrotactile feedback for mobile and wearable interactions, we identify the following design requirements (DRs) for VIREO:

- DR_1: *Web-based availability.* VIREO must be easily accessible to researchers and practitioners from various platforms and operating systems, for which a web-based software design solution is the most straightforward approach to implementing such a desideratum. The results from our SLR showed that the majority of the software tools proposed in the scientific literature were delivered for specific operating systems, platforms, or devices (see Table 1), which constitutes a limitation in terms of their large-scale accessibility.

DR_2: *Model-based design for accurate numerical modeling of vibrotactile feedback patterns.* VIREO must provide access to a library of mathematical models in order to specify vibrotactile patterns with numerical precision, including models commonly used for implementing vibrotactile feedback in the scientific literature, e.g., linear or exponential variation of vibration intensity over time (Park and Choi, 2017; Pezent et al., 2021). Expert practitioners can easily identify these models and readily employ them to specify custom vibrotactile feedback patterns with mathematical precision. Also, patterns obtained with various models can be combined by means of *pattern composition* techniques (Boer et al., 2017; Enriquez and MacLean, 2003; Ryu and Choi, 2008; Swindells et al., 2006, 2014) to author increasingly complex designs of vibrotactile feedback. For example, by specifying a sequence of a linearly increasing intensity of vibration followed by a linearly decreasing one, a "triangle" pattern is obtained.

DR_3: *Specification of vibrotactile patterns via user demonstration to foster rapid prototyping.* VIREO must provide simple mechanisms that enable practitioners to quickly generate vibrotactile feedback patterns based on first-order approximations provided by means of demonstrations, e.g., sketches and free drawing for a graphical authoring software tool. This design requirement enables novice users to readily employ VIREO before becoming familiarized with numerical models (DR_2), but also expert users that wish to quickly evaluate preliminary ideas for vibrotactile feedback patterns.

DR_4: *Easy integration with mobile and wearable devices.* Authoring work usually takes place on desktop PCs with large displays and computing resources, from which the results are deployed to the target devices, which in our case are represented by mobile and wearable devices supporting vibrotactile feedback via the integrated vibration actuators. Consequently, such devices must easily integrate with VIREO so that practitioners can readily test their vibrotactile pattern designs and integrate those patterns into applications. The focus of VIREO on mobile and wearable devices is motivated by the prevalence of the former and increasing adoption of the latter in the context where smart wearables represent the fastest-growing technological innovation for mobile users (Menear, 2020).

A simple way to implement DR_4 is for the target device to support web standards (e.g., HTML), web languages (e.g.. JavaScript), and data-interchange formats designed for the web (e.g., JSON), just like VIREO. This requirement is met by all smartphones and by many wearables with integrated Wi-Fi functionality. Some wearables, such as smartwatches running Tizen OS, can even be programmed with web technology alone, such as HTML and JavaScript.[10] Also, web languages are becoming increasingly popular among developers, e.g., "as of early 2020, JavaScript and HTML/CSS were the most commonly used programming languages among software developers around the world, with nearly 68 percent of respondents stating that they used JavaScript and 63.5 percent using HTML/CSS," according to a Statista April 2021 report.[11] The web-based orientation of VIREO (requirement DR_1) makes this desideratum easily implementable on the web, i.e., target devices can access a dedicated web page generated automatically by VIREO that enables controlled access to vibrotactile patterns. Thus, we add another requirement for VIREO that contours its scope more precisely:

[10] Tizen Studio enables development of web applications for mobile, wearable, and TV devices, which consist of HTML, JavaScript, and CSS combined in a package deployed to a Tizen device; see https://docs.tizen.org/application/web/index.

[11] Statista. Most used programming languages among developers worldwide, as of early 2020, https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages.

DR_5: *JavaScript-orientation.* Target devices that integrate with VIREO or for which VIREO exports vibrotactile feedback patterns support JavaScript applications, either natively or in a web browser.

Design requirements DR_1 to DR_3 ensure that VIREO is easily accessible to a large category of practitioners that use diverse platforms and operating systems, while DR_4 and DR_5 ensure that VIREO addresses uniformly a large category of mobile and wearable devices by leveraging their built-in web connectivity. Moreover, support for DR_5 can be found in previous initiatives in the HCI engineering community for JavaScript-based development frameworks for wearables, such as Weave (Chi and Li, 2015) for cross-wearable interaction, FlowIO (Shtarbanov, 2021) for wearable soft robotics, and SAPIENS (Schipor, Vatavu, and Wu, 2019b) for peripheral interactions with mobile and wearable devices, to name just a few examples.

3.2. Technical Implementation Details

We implemented VIREO using Node.js,[12] a popular cross-platform runtime environment based on JavaScript. For the back-end, we employed NestJS,[13] one of the most popular Model-View-Controller frameworks for Node.js, enabling design and development of efficient and scalable applications for the web. We preferred NestJS since it offers a collection of handy, well-structured tools out of the box, such as WebSockets and an API for CLI tools for best-practice architectural patterns and well-structured web applications, among other useful features. We developed the client-side application with VueJS,[14] a fast and powerful JavaScript framework for reactive UIs so that VIREO can be accessed from many devices with a uniform user experience. This choice of technology enabled us to reuse JavaScript code between the server and client sides of VIREO, reducing thus efforts and resources for code writing and testing. To represent vibrotactile patterns graphically in the UI, we employed Chart.js,[15] one of the most popular JavaScript libraries for data visualization on the web. Moreover, Chart.js can be easily extended via plugins, which allowed us to rapidly add the extra functionality needed in our UI, such as direct manipulation of the individual models composing a vibrotactile pattern, custom visualization effects, and free-draw input directly on the chart with the mouse, stylus, and finger. Next, we present user workflows in VIREO and describe authoring techniques for vibrotactile feedback patterns.

3.3. User Workflows

To implement our design requirements DR_2 to DR_4 (Subsection 3.1), we devised the following main user workflows for VIREO (see Figure 2 for visual illustrations):

(a) A new user registers with VIREO. After logging in, the user gets access to a workspace where they can create, edit, test, and export vibrotactile patterns. The initial workspace contains an example of a vibrotactile pattern visualized as a 1D signal on the time dimension. The pattern is simple—a linear model followed by an exponentially decreasing one, describing the variation in vibration intensity—with only few parameters to control, such as the slope for the linear

[12] Node.js, `https://nodejs.org`.
[13] NestJS - A progressive Node.js framework, `https://nestjs.com`.
[14] Vue.js, the progressive JavaScript framework, `https://vuejs.org`.
[15] Chart.js — Open source HTML5 Charts for your web site, `https://www.chartjs.org`.

model and the decay rate for the exponential model, respectively; see Figure 2a. The user can play with these parameters to see their effect, explore and add new models, and save the pattern to their library.

(b) A registered user logs into their workspace in VIREO, where their previously authored vibrotactile patterns are available in the form of a *library of patterns*. The user can add, edit, and remove patterns; see Figure 2b. There are three ways to author a new vibrotactile pattern in VIREO: (i) from scratch, by using mathematical models, e.g., linear, exponential, harmonic, etc., (ii) from scratch, by sketching the shape of the pattern with the mouse, stylus, or finger on the screen, and (iii) by cloning an existing vibrotactile pattern from the library and editing it anew.

(c) The user wishes to test how a selection of the vibrotactile patterns they created with VIREO feels on a smartphone and a smartwatch, respectively. The user adds those patterns from the library to a *collection of patterns*, for which a quick access code is automatically generated by VIREO; see Figure 2c. From the target device, VIREO is accessed with that code in the web browser, which gives instant access to the patterns of the collection. The user can play the patterns on the smartphone and smartwatch to evaluate their effect.

(d) The user has decided on a collection of vibrotactile patterns to use in an application targeting a specific device. They export the collection as a JavaScript file that contains the patterns encoded in JSON, but also JavaScript code that plays the patterns with the Vibration API,[16] a Web API compatible with many platforms. The file is imported into the application that is being developed for the target device; see the user workflow in Figure 2d.

(e) The user wants to share a subset of vibrotactile patterns from their VIREO library with a colleague developer. They create a new collection, to which they add the vibrotactile patterns they wish to share, and send the access code to their colleague. With the access code, the second user can either play the vibrotactile patterns on a target device or import them into their own library from their VIREO workspace; see Figure 2e for the workflow.

We kept the user workflows simple and with few steps in order to ensure the functionality planned for VIREO, according to our design requirements, with minimum effort expected from users. The most important feature of VIREO is authoring vibrotactile feedback patterns, for which we implemented several options: model-based design and editing of model parameters, composition of multiple models, and free-draw input for rapid specification of vibrotactile feedback based on user demonstration. In the following subsections, we present these features in detail.

3.4. Model-based Design of Vibrotactile Feedback

We define a vibrotactile pattern as a function of time $A(t)$ specifying the intensity of the vibration, normalized between between 0 (no vibration) and 100 (vibration provided at the maximum intensity of the actuator). For example, the function $A(t) = 100t\big|_0^2$, defined over the time interval $[0, 2s]$, specifies a pattern that increases linearly over a period of one second to the maximum intensity of the actuator, where it is capped for another second. Individual patterns can be composed in order to generate more complex patterns, i.e., $A(t){=}A_1(t){+}A_2(t-\tau)$, where τ specifies the time offset for the

[16]Vibration API - Web APIs, https://developer.mozilla.org/en-US/docs/Web/API/Vibration_API.

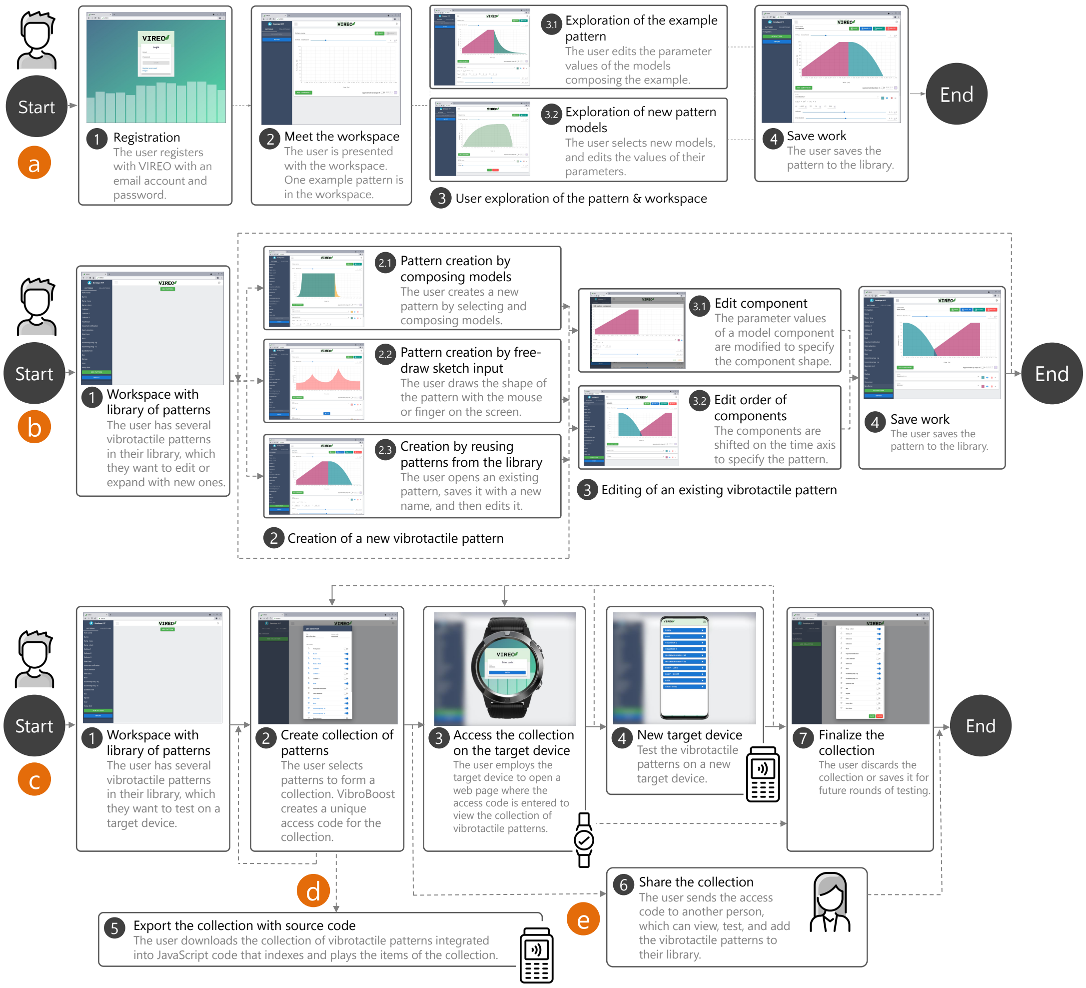

Figure 2. Five user workflows, (a) to (e), implemented in VIREO; see Subsection 3.3 for details.

second pattern. For example, $A_1(t)$ is the linearly increasing pattern defined before and $A_2(t)$ is a fast exponentially decreasing signal, $A_2(t) = 100e^{-3t}$, which starts immediately after $A_1(t)$, i.e., $\tau = 2s$, and decreases to zero in the following second: $A(t) = 100t|_0^2 + 100e^{-3(t-2)}|_2^3$ defined over the time interval $[0, 3s]$. This composition rule applies for series of more than two patterns, as follows: $A(t) = \sum_{i=1}^{n} A_i(t - \tau_i)|_{t_{1i}}^{t_{2i}}$, where τ_i is the offset of the i-th pattern of the composition defined over the time interval $[t_{1i}, t_{2i}]$. Note that in the above mathematical formalism the patterns $A_i(t)$ are defined only for positive t values, otherwise they compute to 0 (no vibration).

We implemented the following numerical models $A(t)$ for vibrotactile patterns in VIREO, for which we drew inspiration from prior software authoring tools, systems, and studies implementing vibrotactile feedback (Boer et al., 2017; Hong et al., 2013; Park and Choi, 2017; Pezent et al., 2021):

(1) *Linear model.* $A(t) = at + b$, where a and b are the slope and intercept. This model can be used to create patterns of increasing (a>0), decreasing (a<0), or constant (a=0) intensity of the vibration that start at the intensity denoted by

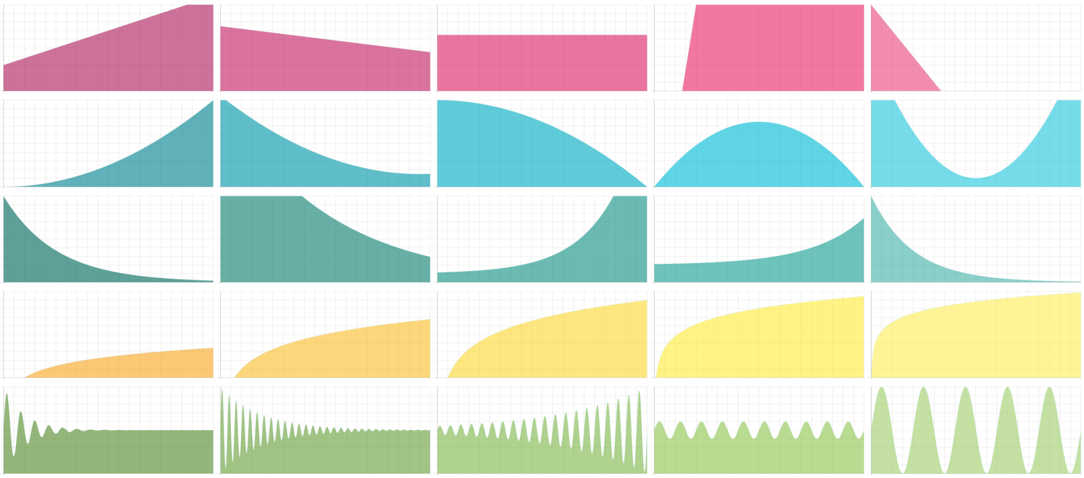

Figure 3. Examples of vibrotactile patterns generated using the numerical models available in VIREO. From top to bottom: linear, quadratic, exponential, logarithmic, and harmonic models with various parameters; see the text for the corresponding formulae.

b; see the first row in Figure 3 for a few examples.

(2) *Quadratic model.* $A(t) = at^2 + bt + c$, a standard form expression that defines a parabola with the axis of symmetry parallel to the vertical axis. The coefficient a controls the degree of curvature and its sign specifies how the parabola opens, either upwards (a>0) or downwards (a<0). Coefficients a and b determine the location of the axis of symmetry ($-b/2a$), and c specifies the height of the parabola when intercepting the vertical axis. This model generates vibration patterns that increase or decrease with more degrees of freedom compared to the linear model (e.g., $A(t) = 100(t-1)^2$ starts with a fast decrease in vibration intensity that attenuates progressively), but also symmetric patterns that increase then decrease and vice versa (e.g., $A(t) = -300t^2 + 300t$ increases to 75% of the maximum vibration intensity of the actuator during the first half of a second and then decreases to zero in the next half); see the second row from Figure 3 for more examples.

(3) *Exponential model.* $A(t) = A_{max}e^{-\lambda t+D} + c$ specifies vibration intensity that varies exponentially with time, where A_{max} denotes the maximum amplitude and c is the intercept as in the previous models. This model can be used to generate a variety of vibration patterns that increase or decrease at various rates according to the λ and D parameters, e.g., $A(t) = e^{2.5t+0.5}$ specifies a vibration that increases slowly from 0 to 50% of the maximum intensity of the actuator over one and a half seconds; see the third row from Figure 3.

(4) *Logarithmic model.* $A(t) = A_{max}ln(at) + c$ is the inverse of the exponential model, useful to specify vibrations that increase at various rates then cap at specific intensity values; see the fourth row from Figure 3.

(5) *Harmonic model.* $A(t) = A_{max}e^{-\tau t}sin(2\pi\omega t) + c$ specifies harmonic oscillations with frequency ω. $A_{max}e^{-\tau t}$ is an envelope function parameterized by the maximum intensity A_{max} and the decaying constant τ. This model can generate a variety of patterns that involve oscillations; see the fifth row from Figure 3 for several examples.

Figure 4 shows how the parameter values are specified for models in VIREO. Also, the patterns generated using these models can be combined to create increasingly more

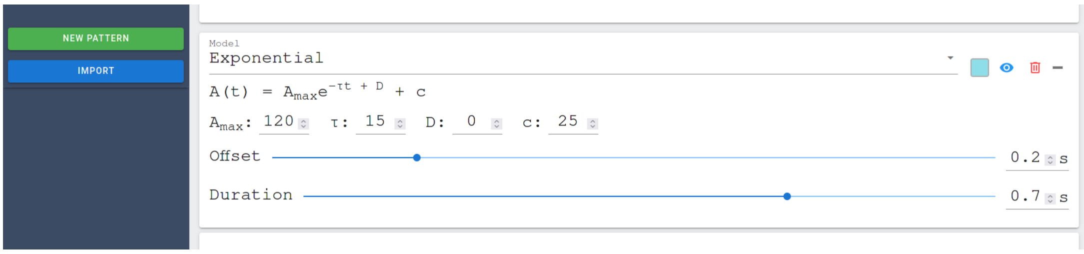

Figure 4. Specifying parameter values, time offset, and duration for numerical models in VIREO.

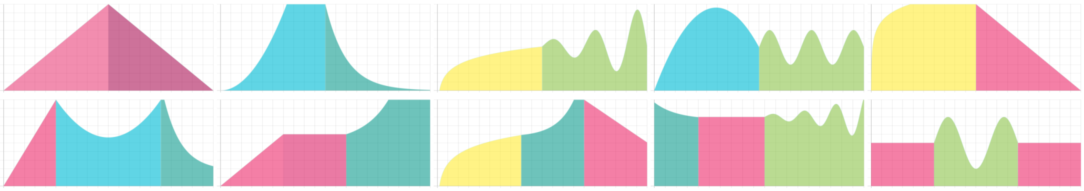

Figure 5. Examples of vibrotactile patterns authored with VIREO via pattern composition. *Note:* the colors match those from Figure 3, e.g., red hues specify the linear model. Through composition, increasingly complex vibration patterns can be authored in VIREO.

complex patterns; see Figure 5. The time offset of the individual, simpler patterns can be specified numerically or via direct manipulation with the mouse by dragging the pattern to the desired location, where a magnetic effect adjusts its position precisely.

3.5. Sketch-based Specification of Vibrotactile Feedback

Model-based authoring of vibrotactile feedback produces numerically accurate specification of patterns of various complexity, as demonstrated in the previous section. However, a use case that arises in practice is when the practitioner wishes to rapidly test how a vibrotactile pattern feels like, without being concerned for the time being by the exact mathematical modeling of that pattern. To support this requirement (DR_3), also useful for novice users, we implemented a *free draw* authoring feature, where the user employs the mouse, stylus, or finger to draw the approximate shape of the vibrotactile pattern on the screen; see Figure 6, left. VIREO approximates this shape with rectangles of variable width (which are actually linear models with zero slope), for which the minimum width can be specified by the practitioner, e.g., 50ms for the example illustrated in Figure 6, right. Patterns created with free-draw input are compatible for composition with other patterns since we implemented the free draw feature just like any another model in VIREO.

3.6. Integrating Vibrotactile Feedback Patterns from VIREO in Mobile and Wearable Applications

According to the user workflow illustrated in Figure 2d, collections of vibrotactile patterns can be exported from VIREO in the form of a JavaScript file. The file contains an array of patterns specified as a series of vibration intensities and corresponding time duration. This intermediate representation (see Figure 6, right for an example) is an approximation of the pattern waveform from VIREO that is convenient for rendering vibrotactile feedback with the Vibration Web API on mobile and wearable

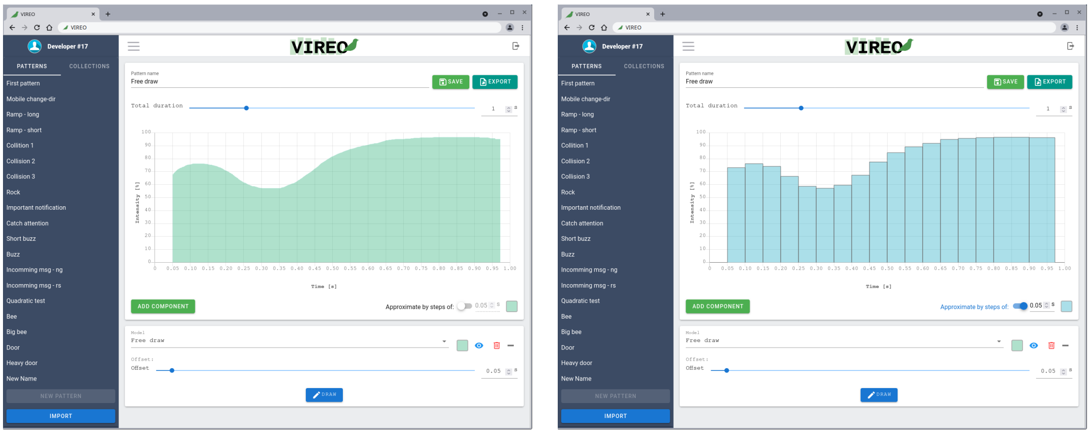

Figure 6. *Left*: free-draw pattern authoring in VIREO with the mouse, stylus, or finger directly on the screen. *Right:* an approximation of the pattern from the left image with rectangles (linear models with zero slope).

devices supporting JavaScript, either as standalone applications or in the web browser. However, since the Vibration API only accepts patterns represented as series of on-off pulses,[17] we simulate vibration intensity with pulse-width modulation (PWM) (Barr, 2001). Listing 1 shows an example of JavaScript code exported from VIREO. The object `vireo` exposes the method `toNavigatorFormat(patternName)` that takes as a parameter the name of the vibrotactile pattern and implements PWM to convert vibration intensities from the pattern representation into on-off pulses expected by the `navigator.vibrate(...)` function of the Vibration Web API; see Listing 2. The `navigatorVibrate(patternName)` method calls `navigator.vibrate(...)` directly.

```
const VIREOPatterns = {
  "Start": [
    {intensity:84, duration:100}, {intensity:83, duration:50},
    {intensity:33, duration:100}, {intensity:84, duration:100},
    {intensity:83, duration:50}, {intensity:33, duration:100},
    {intensity:84, duration:100}, {intensity:83, duration:50},
    {intensity:33, duration:100}, {intensity:84, duration:100},
    {intensity:83, duration:50}, {intensity:33, duration:100}
  ],
  "Game over": [
    {intensity:58, duration:50}, {intensity:55, duration:50},
    {intensity:53, duration:50}, {intensity:52, duration:50},
    {intensity:51, duration:50}, {intensity:51, duration:50},
    {intensity:50, duration:50}, {intensity:50, duration:50},
    {intensity:50, duration:100}, {intensity:95, duration:50},
    {intensity:85, duration:50}, {intensity:75, duration:50},
    {intensity:65, duration:50}, {intensity:55, duration:50},
    {intensity:45, duration:50}, {intensity:35, duration:50},
    {intensity:25, duration:50}, {intensity:10, duration:100}
  ]
};
function toNavigatorFormat(patternName) { ... }
function navigatorVibrate(patternName) { ... }
function vireo() { ... }
export default vireo;
```

Listing 1 Excerpt of JavaScript code with a collection of vibrotactile patterns exported from VIREO.

```
import vireo from "./exported-patterns";

const pattern = vireo.toNavigatorFormat("Start");
```

[17] https://developer.mozilla.org/en-US/docs/Web/API/Vibration_API

```
navigator.vibrate(pattern);

// or
// vireo.navigatorVibrate("Start");
```
Listing 2 Excerpt of JavaScript code used in end-user applications.

4. Demonstrative Applications

We demonstrate in this section how VIREO can be used for practical implementation of vibrotactile feedback in user interfaces for mobile and wearable devices, and present applications for a smartphone, smartwatch, smart armband, and pair of smartglasses. Note that our goal in this section is not to focus on design choices for various mappings between application events and vibration patterns (that have been examined thoroughly elsewhere in the scientific literature (Boer et al., 2017; Brewster and Brown, 2004; Cauchard, Cheng, Pietrzak, and Landay, 2016; Plaisier, Holt, and Kappers, 2020; Seifi et al., 2015; Shiraga, Kinoshita, and Go, 2016; Vyas, Taha, Blum, and Cooperstock, 2020)), but rather to demonstrate the versatility of our web-based approach adopted for VIREO to readily implement vibrotactile feedback for mobile and wearable interactions. To this end, we employ the VibViz library (Seifi et al., 2015) to identify vibration patterns possessing diverse functional and affective characteristics suited to our application types.

4.1. Vibrotactile Feedback for a Smartphone Video Game

We implemented the classical Snake[18] game, where the user maneuvers a continuously moving snake, and tries to catch apples to accumulate points, while the snake's body grows in length. We chose three game events to augment with vibrotactile feedback: the snake changes its direction of movement following user input, points are awarded when apples are eaten, and the game is over. Figure 7a illustrates the corresponding vibrotactile patterns, which we chose from the VibViz (Seifi et al., 2015) library by using the keyword "game." For example, when points are awarded to the user after the snake has eaten an apple, a vibration pattern is rendered with a happy connotation (Seifi et al., 2015), while the "game over" event has a pattern suggesting sadness, but also encouragement. Since VibViz does not provide numerical descriptions of these patterns, we specified them with the numerical models available in VIREO, e.g., we created the "game over" pattern by combining an exponential model ($10e^{-10t}+50\big|_0^{0.5}$) with a linear one ($-200t+100\big|_{0.5}^{1}$) and the "points awarded" pattern with three consecutive linear models ($650t\big|_0^{0.15}$, $650(t-0.15)\big|_{0.15}^{0.30}$, and $650(t-0.3)\big|_{0.30}^{0.45}$) and a triangle composed of a linearly increasing and decreasing model ($650(t-0.45)\big|_{0.45}^{0.60}$ and $-650(t-0.60)+97.5\big|_{0.60}^{0.75}$). The same results were obtained, with unnoticeable differences, when we recreated these patterns in VIREO with the free-draw input technique, which approximated the drawings with 50ms-long rectangles as in Figure 6, right.

[18]Snake (video game genre) - Wikipedia, https://en.wikipedia.org/wiki/Snake_(video_game_genre).

Figure 7. Applications for a smartphone, smartwatch, smart armband, and smartglasses, respectively, for which we integrated patterns from VIREO to provide vibrotactile feedback to the user corresponding to various application events, e.g., notification of “points awarded” in a mobile game or a “strong encouragement” vibration as a motivation during jogging for a smartwatch user.

4.2. Delivery of Notifications on Smartwatches

We implemented a smartwatch application for joggers. We used Tizen Studio to develop a HTML/CSS/JavaScript web application that we deployed to a Samsung Galaxy Watch 3. The application monitors running time and provides the user with vibrotactile feedback according to specific milestones: start of the running exercise, light and strong encouragements when the running time approaches predefined milestones, such as 15 and 30 minutes, and achieving a new personal record of running time. We created vibrotactile patterns for these events in VIREO using composition of the linear, logarithmic, and harmonic models, respectively; see Figure 7b for a visual illustration of these patterns and our smartwatch application. For example, we created the vibration for “light encouragement” with a logarithmic model $(23ln(750t)|_0^{0.10})$ followed by a linear model $(-670(t-0.10)+100|_{0.10}^{0.25})$, to suggest positive emotion (Seifi et al., 2015) to the user.

4.3. Augmenting Touch Input on Interactive Tabletops with a Wearable Device and Vibrotactile Feedback

For our next demonstration, we considered an information visualization application running on an Ideum interactive tabletop, for which we enhanced multitouch input with vibrotactile feedback delivered on the user's hand. We developed a web application displaying a map with touristic points of interest indicated by colored circles; see Figure 7c. The user employs zoom-in and zoom-out multitouch gestures to browse regions of the map. However, because of the large number of points of interest, not all of them can be effectively presented visually at once on the map. Rather, some points come into view and others are hidden as the user zooms in into a specific region of the map. To inform the user about how many points of interest are available in that region, even if not all of them are visible, we provided vibrations on the user's arm with the Myo[19] armband and the corresponding Myo JavaScript API. We selected four application events to augment with vibrotactile feedback when the user touches the screen (denoted by numbers 8 to 11 in Figure 7c): many points of interest are available around the touch point but are hidden from view, some points of interest are available, very few points are still hidden from view, and selection confirmation for a point of interest. We designed those patterns in VIREO using composition of the linear, exponential, and harmonic models, respectively, to suggest the metaphorical connotations (Seifi et al., 2015) of different amounts of data. For example, the vibration pattern with two spikes was specified in VIREO by composing exponential ($e^{19.6(t-0.1)} + 49|_{0.10}^{0.30}$) and linear ($-333.3(t-0.3) + 100|_{0.30}^{0.45}$) models.

4.4. Vibrotactile Feedback on Smartglasses

As a final demonstration, we accessed the interactive map web application presented in Subsection 4.3 from the Vuzix Blade[20] smartglasses, a device with see-through lenses and integrated touchpad and vibration motors in the temple; see Figure 7d. The small display of the glasses device exacerbated the visualization problem about too many points of interest to be visualized at once on the map. Since Vuzix Blade can run a web browser that supports JavaScript and the Vibration Web API, our application ran on the smartglasses without any adaptation, delivering the four types of vibrotactile feedback patterns to the user when touching the glasses temple.

4.5. Summary

All the applications presented in this section were developed using HTML, CSS, and JavaScript either as standalone applications (e.g., for the Galaxy Watch 3) or applications on the web that can be accessed with a web browser (for the smartphone, tabletop, and smartglasses demonstrators, respectively). Our goal with these applications was to highlight the practicality and feasibility of the DR_5 design requirement for VIREO: by leveraging the built-in capacity of mobile and wearable devices to connect to the web and support JavaScript, vibrotactile feedback can be straightforwardly integrated in applications targeting such devices.

[19] https://developerblog.myo.com
[20] https://www.vuzix.com/products/blade-smart-glasses-upgraded

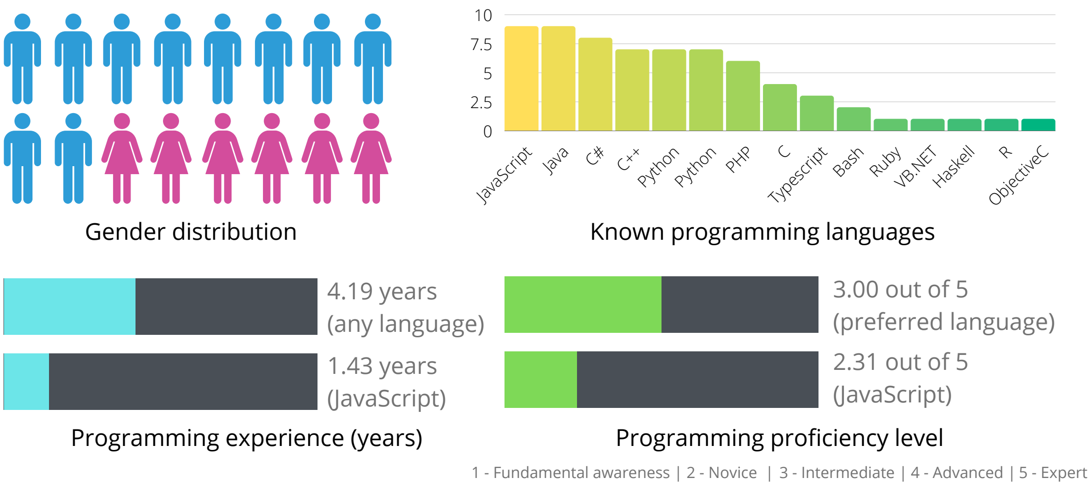

Figure 8. Demographic details of the participants from our usability study.

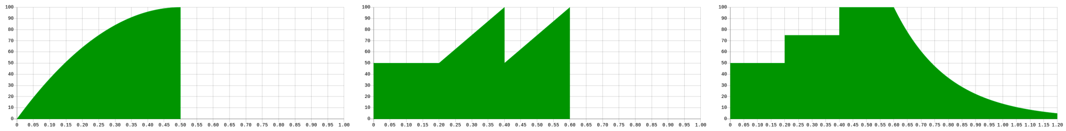

Figure 9. The task for the usability study was to employ VIREO to author the three vibrotactile patterns illustrated in this figure. From left to right, the complexity of the vibration patterns increases from one model to a combination of linear models to a combination of different models (right).

5. User Study

We conducted a user study with coders with various levels of experience in various programming languages to evaluate the usability of VIREO as a software tool on the web for authoring vibrotactile feedback for applications on mobile devices. The study was approved by the Ethics Committee of the University of Suceava, ref. 50/21.01.2022.

5.1. Participants

We recruited a total number of sixteen participants from our University's student body with an announcement on an email list. Ten participants (62.5%) declared themselves as male and six (37.5%) as female. Their age range was between 21 and 34 years (M=25.6, SD=3.4 years). All of the participants were Computer Science students having up to ten years of programming experience (M=4.2, SD=3.0 years) with a diversity of programming languages and platforms. Their average self-reported programming proficiency level was 3.00 (SD=1.03) on a scale from 1 ("fundamental awareness") to 2 ("novice"), 3 ("intermediate"), 4 ("advanced") to 5 ("expert"). Our participants self-reported experience with JavaScript, the programming language of choice for VIREO, was up to four years (M=1.4, SD=1.6 years) and their average JavaScript proficiency level was 2.31 (SD=1.25) reported on the scale ranging from 1 ("fundamental awareness") to 5 ("expert"). A number of eight participants (50%) were also working in the software industry as software developers, engineers, and quality assurance testers. None of the participants had used software tools for designing vibrotactile feedback before our study. Figure 8 shows the demographic details of our sample of participants.

5.2. Procedure and Task

The study was conducted entirely online. Registration to the study was done via a Google Forms questionnaire that collected the email addresses of the volunteers. We created individual login accounts in VIREO for each participant and sent the login details to the email addresses they used to register for the study. A second Google Forms document secured the informed consent from the registered participants that responded to our follow-up email. Upon login, the participants were provided a tutorial of VIREO and a 3-minute long video demonstrating how to use VIREO to create vibrotactile feedback patterns and collections of patterns. The instructions given to the participants were to freely explore VIREO and, once they felt they understood how it worked, they had to author three vibrotactile patterns with different levels of complexity (see Figure 9) and add them to a collection. To enable the participants to easily test the vibrotactile patterns on their smartphone, we also provided the snake video game described in Section 4 already set up to play the three vibrotactile patterns from Figure 9. We did not impose any time constraints on the exploration of VIREO or on performing the vibrotactile pattern authoring task. When we analyzed the logs automatically captured during the study, we found that the participants spent on average 88 minutes with VIREO. After the participants finished the task, they were asked to fill in a Google Forms questionnaire that collected their experience using VIREO with specific measures, presented in detail next.

5.3. Measures

We collected the following measures during our study:

(1) SUS, the System Usability Scale (Brooke, 1996), a 10-item instrument for evaluating overall usability. For each item of SUS, the degree of agreement is entered using a 5-point Likert scale with responses ranging from 1 ("strongly disagree") to 5 ("strongly agree"). The individual responses are aggregated into a score ranging from 0 (low usability) to 100 (perfect score).

(2) CSUQ, the Computer System Usability Questionnaire (Lewis, 1995),[21] a 19-item instrument for assessing overall user satisfaction with system usability that provides four scores: overall usability (CSUQ-Overall), system usefulness (CSUQ-SysUse), information quality (CSUQ-InfoQual), and interface quality (CSUQ-InterQual). For each item of CSUQ, the degree of agreement is entered using a 7-point Likert scale with possible responses ranging from 1 ("strongly agree") to 7 ("strongly disagree") and a "not applicable" point outside the scale. Lower scores denote better usability.

(3) Perceived-Difficulty-Models. We evaluated the perceived difficulty of employing mathematical models to author vibrotactile feedback patterns in VIREO (see Subsection 3.4), which we measured with a 5-point Likert scale with items from 1 ("not difficult") to 2 ("little difficult"), 3 ("moderately difficult"), 4 ("difficult") to 5 ("very difficult") in response to the question "How would you rate the use of mathematical models for creating vibrotactile patterns in VIREO?"

(4) Perceived-Difficulty-FreeSketch. We evaluated the perceived difficulty of employing the free-sketch option to author vibrotactile feedback patterns in VIREO (see Subsection 3.5), which we measured with a 5-point Likert scale with

[21] We used the CSUQ version instead of PSSUQ by following the recommendation from (Lewis, 1995, p. 77) to apply CSUQ when the usability study is in a nonlaboratory setting, which was the case for our study.

items ranging from 1 ("not difficult at all") to 5 ("very difficult") in response to the question "How would you rate the free-sketch option for creating vibrotactile patterns in VIREO?"

5.4. Results

The average SUS score was 70.47 (SD=24.63), representing a relatively high value falling in the "acceptable" range and located close to the "good" usability mark on the SUS scale, respectively, according to the acceptability ranges[22] and adjective ratings[23] for interpreting SUS scores from (Bangor et al., 2009; Bangor, Kortum, and Miller, 2008). Also, the average CSUQ-Overall score was 2.68 (SD=1.41), located in the first part of the 7-point CSUQ scale (lower values that are closer to 1 denote better performance). The scores obtained on each CSUQ subscale were of a similar magnitude: 2.74 for CSUQ-SysUse (SD=1.59), 2.75 for CSUQ-InfoQual (SD=1.37), and 2.40 for CSUQ-InterQual (SD=1.41), respectively. By converting the CSUQ-Overall score to a 0-100 scale (Lewis, 2018a, p. 6), we obtained 70.96 (SD=26.52), a value nearly identical with the one obtained with SUS. These results show a good overall usability for the task of authoring vibrotactile feedback patterns with VIREO, confirmed by two popular usability tests.

To understand the effect of our participants' programming experience and programming proficiency levels on perceived usability, we divided the participants into two groups, beginners and advanced, by applying the following rule: a participant was in the advanced group if they had at least five years of programming experience or their self-declared programming proficiency level was at least 4 ("Advanced, I can successfully apply theory concepts"). By applying this rule, we obtained two equally-sized groups with eight participants each: beginners (M=1.75, SD=1.17 years of programming experience and M=2.25, SD=0.89 on the programming proficiency scale) and advanced (M=6.63, SD=2.13 years of programming experience and M=3.75, SD=0.46 on the programming proficiency scale). Mann-Whitney U tests did not find a statistically significant effect of the programming experience on the SUS (U=25.000, Z=-0.738, p=.505, n.s.) or CSUQ (U=31.000, Z=-0.105, p=.959, n.s.) scores. These results suggest a similar level of perceived usability for both beginners and advanced programmers, but further evaluations with larger user groups are recommended.

Regarding the actual method to author vibrotactile patterns, i.e., model-based or free-sketch input, we found an average Perceived-Difficulty-Models of 1.88 (Mdn=2) and an average Perceived-Difficulty-FreeSketch of 2.31 (Mdn=2), both close to the "little difficult" anchor from the 5-point Likert scale used to evaluate these measures. A Wilcoxon signed-rank test did not detect any statistically significant difference between the two authoring methods ($Z = -1.567$, p=.117>.05, *n.s.*). Also, Mann-Whitney U tests did not detect any statistically significant effects of the programming experience, beginners and advanced, on Perceived-Difficulty-Models (U=26.000, Z=-0.732, p=.574, n.s.) or Perceived-Difficulty-FreeSketch (U=30.000, Z=-0.217, p=.878, n.s.), respectively.

[22] According to the standard letter grade scale for interpreting SUS scores (Bangor, Kortum, and Miller, 2009), products that score in the 90s are "exceptional," products that score in the 80s are "good," and products that score in the 70s are "acceptable"; also see Lewis (2018b).

[23] According to Bangor et al. (2009), the correspondence between adjective ratings and mean SUS scores are: "worst imaginable" (M=12.5, SD=13.1), "awful" (M=20.3, SD=11.3), "poor" (M=35.7, SD=12.6), "OK" (M=50.9, SD=13.8), "good" (M=71.4, SD=11.6), "excellent" (M=85.5, SD=10.4), and "best imaginable" (M=90.9, SD=13.4); also see Lewis (2018b).

6. Limitations

There are a few limitations to VIREO, which we acknowledge in this section. First, our implementation leveraging JavaScript and the Vibration API enables us to address a variety of mobile and wearable devices, but also limits the applicability of VIREO to those devices that support the API. Although a large number of devices and platforms are implicitly supported, e.g., browser compatibility specifications[24] include Chrome, Edge, Firefox, Opera, WebView Android, Chrome Android, Opera Android, and Samsung Internet, other platforms and browsers, such as Internet Explorer, Safari, and Safari on iOS, are not. For these platforms, future work will look at generating platform-specific code for integrating vibrotactile patterns authored with VIREO (Subsection 3.6). Second, VIREO does not dispose yet of a set of predefined vibrotactile patterns to start off the authoring process, although such a feature would be helpful for rapid prototyping alongside our free-sketch input method (Subsection 3.5). We will address this limitation in future versions of VIREO. In the next section, we present more future work opportunities.

7. Conclusion and Future Work

We introduced VIREO, a tool for the web that enables design of vibrotactile feedback patterns and easy deployment and integration of those patterns in applications targeting mobile and wearable devices with web connectivity and support for JavaScript. We also presented the results of a SLR study conducted to analyze and understand contributions in the scientific community regarding tools and platforms that enable authoring and implementation of vibrotactile feedback. Due to its web-based orientation and specific focus on mobile and wearable devices, VIREO stands out distinctively in this landscape. Moreover, we make VIREO freely available to researchers and practitioners. Future work will consider several new features that we plan for the next versions of VIREO, as follows: (1) integration with web-based software architecture for implementing interactions in smart environments (Schipor, Vatavu, and Vanderdonckt, 2019a; Schipor et al., 2019b) to extend the range of possible mobile and wearable applications of VIREO in the context of ambient intelligence, (2) integration with Virtual and Augmented Reality applications, such as applications running on Microsoft HoloLens and HTC Vive, for which WebSocket-based communications can be used to integrate directly with the collections of patterns from VIREO, and (3) extension of VIREO with a library of vibrotactile patterns as a starting point for practitioners to customize vibration patterns for their applications.

Acknowledgements

This work is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 860114.

[24]Details available at https://developer.mozilla.org/en-US/docs/Web/API/Vibration_API. Also see https://caniuse.com/vibration for more details and tests.

Notes on contributors

Mihail Terenti is a Marie Skłodowska-Curie Early Stage Researcher at the Ştefan cel Mare University of Suceava, Romania, where he works on multimodal feedback to augment the user experience of touch input.

Radu-Daniel Vatavu is a Professor of Computer Science at the Ştefan cel Mare University of Suceava, where he directs the Machine Intelligence and Information Visualization Research Laboratory. His research interests include human-computer interaction, augmented reality, ambient intelligence, and accessible computing.

References

Anderson, F., Grossman, T., Wigdor, D., Fitzmaurice, G., 2015. Supporting subtlety with deceptive devices and illusory interactions. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, p. 1489–1498.
URL https://doi.org/10.1145/2702123.2702336

Bangor, A., Kortum, P., Miller, J., 2009. Determining what individual SUS scores mean: Adding an adjective rating scale. J. Usability Studies 4 (3), 114–123.
URL https://uxpajournal.org/determining-what-individual-sus-scores-mean-adding-an-adjective-rating-scale

Bangor, A., Kortum, P. T., Miller, J. T., 2008. An empirical evaluation of the system usability scale. International Journal of Human–Computer Interaction 24 (6), 574–594.
URL https://doi.org/10.1080/10447310802205776

Barr, M., September 2001. Pulse width modulation. Embedded Systems Programming, 103–104.

Basdogan, C., Giraud, F., Levesque, V., Choi, S., 2020. A review of surface haptics: Enabling tactile effects on touch surfaces. IEEE Transactions on Haptics 13 (3), 450–470.
URL http://dx.doi.org/10.1109/toh.2020.2990712

Boer, L., Vallgårda, A., Cahill, B., 2017. Giving form to a hedonic haptics player. In: Proceedings of the 2017 Conference on Designing Interactive Systems. DIS '17. ACM, New York, NY, USA, p. 903–914.
URL https://doi.org/10.1145/3064663.3064792

Brewster, S., Brown, L. M., 2004. Tactons: Structured tactile messages for non-visual information display. In: Proceedings of the Fifth Conference on Australasian User Interface - Volume 28. AUIC '04. Australian Computer Society, Inc., AUS, p. 15–23.

Brezolin, F. L., Santos, G., de Lima, J. C., Zanella, M., Rieder, R., De Marchi, A. C. B., 2017. Evaluating the performance of wearable tecassist device using aural and tactile feedbacks. In: Proceedings of the XVI Brazilian Symposium on Human Factors in Computing Systems. IHC 2017. ACM, New York, NY, USA.
URL https://doi.org/10.1145/3160504.3160574

Brooke, J., 1996. SUS: A Quick and Dirty Usability Scale. In: Jordan, P. W., Thomas, B., McClelland, I. L., Weerdmeester, B. (Eds.), Usability Evaluation in Industry. CRC Press, London, UK, p. 189–194.
URL https://www.taylorfrancis.com/chapters/edit/10.1201/9781498710411-35/susquick-dirty-usability-scale-john-brooke

Casamassima, F., Ferrari, A., Milosevic, B., Rocchi, L., Farella, E., 2013. Wearable

audio-feedback system for gait rehabilitation in subjects with Parkinson's disease. In: Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication. UbiComp '13 Adjunct. ACM, New York, NY, USA, p. 275–278.
URL https://doi.org/10.1145/2494091.2494178

Cauchard, J. R., Cheng, J. L., Pietrzak, T., Landay, J. A., 2016. ActiVibe: Design and Evaluation of Vibrations for Progress Monitoring. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, p. 3261–3271.
URL https://doi.org/10.1145/2858036.2858046

Chen, M., Ma, Y., Li, Y., Wu, D., Zhang, Y., Youn, C.-H., Jan. 2017. Wearable 2.0: Enabling human-cloud integration in next generation healthcare systems. Comm. Mag. 55 (1), 54–61.
URL https://doi.org/10.1109/MCOM.2017.1600410CM

Chen, X., Chen, W., Liu, K., Chen, C., Li, L., 2021. A comparative study of smartphone and smartwatch apps. In: Proceedings of the 36th Annual ACM Symposium on Applied Computing. SAC '21. ACM, New York, NY, USA, p. 1484–1493.
URL https://doi.org/10.1145/3412841.3442023

Chi, P.-Y. P., Li, Y., 2015. Weave: Scripting cross-device wearable interaction. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, p. 3923–3932.
URL https://doi.org/10.1145/2702123.2702451

Clark, B., Schneider, O. S., MacLean, K. E., Tan, H. Z., 2017. Predictable and distinguishable morphing of vibrotactile rhythm. In: Proceedings of the IEEE World Haptics Conference. WHC '17. pp. 84–89.
URL http://dx.doi.org/10.1109/WHC.2017.7989881

Danieau, F., Guillotel, P., Dumas, O., Lopez, T., Leroy, B., Mollet, N., 2018. HFX Studio: haptic editor for full-body immersive experiences. In: Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology. VRST '18. ACM, New York, NY, USA.
URL https://doi.org/10.1145/3281505.3281518

Dong, H., Gao, Y., Al Osman, H., El Saddik, A., 2015. Development of a web-based haptic authoring tool for multimedia applications. In: Proceedings of the IEEE International Symposium on Multimedia. ISM '15. pp. 13–20.

Eid, M., Andrews, S., Alamri, A., El Saddik, A., 2008. HAMLAT: A HAML-Based Authoring Tool for Haptic Application Development. In: Ferre, M. (Ed.), Haptics: Perception, Devices and Scenarios. Springer, Berlin, Heidelberg, pp. 857–866.
URL https://doi.org/10.1007/978-3-540-69057-3_108

Enriquez, M., MacLean, K., 2003. The hapticon editor: a tool in support of haptic communication research. In: Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. HAPTICS '03. pp. 356–362.

Furini, M., Mirri, S., Montangero, M., Prandi, C., 2020. Can IoT Wearable Devices Feed Frugal Innovation? In: Proceedings of the 1st Workshop on Experiences with the Design and Implementation of Frugal Smart Objects. FRUGALTHINGS'20. ACM, New York, NY, USA, p. 1–6.
URL https://doi.org/10.1145/3410670.3410861

Han, T., Han, Q., Annett, M., Anderson, F., Huang, D.-Y., Yang, X.-D., 2017. Frictio: Passive kinesthetic force feedback for smart ring output. In: Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology. UIST '17. ACM, New York, NY, USA, p. 131–142.

URL https://doi.org/10.1145/3126594.3126622
Heller, F., Vanacken, D., Geurts, E., Luyten, K., 2020. Impact of situational impairment on interaction with wearable displays. In: Proceedings of the 22nd International Conference on Human-Computer Interaction with Mobile Devices and Services. MobileHCI '20. ACM, New York, NY, USA.
URL https://doi.org/10.1145/3406324.3410540
Hong, K., Lee, J., Choi, S., 2013. Demonstration-based vibrotactile pattern authoring. In: Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction. TEI '13. ACM, New York, NY, USA, p. 219–222.
URL https://doi.org/10.1145/2460625.2460660
Huang, D.-Y., Chan, L., Jian, X.-F., Chang, C.-Y., Chen, M.-H., Yang, D.-N., Hung, Y.-P., Chen, B.-Y., 2016. VibroPlay: Authoring Three-Dimensional Spatial-Temporal Tactile Effects with Direct Manipulation. In: SIGGRAPH ASIA 2016 Emerging Technologies. SA '16. ACM, New York, NY, USA.
URL https://doi.org/10.1145/2988240.2988250
Israr, A., Zhao, S., Schneider, O., 2015. Exploring embedded haptics for social networking and interactions. In: Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems. CHI EA '15. ACM, New York, NY, USA, p. 1899–1904.
URL https://doi.org/10.1145/2702613.2732814
Jones, L. A., Dec. 2016. Perspectives on the evolution of tactile, haptic, and thermal displays. Presence: Teleoper. Virtual Environ. 25 (3), 247–252.
URL https://doi.org/10.1162/PRES{_}a{_}00266
Kang, N., Lee, S., 2018. A meta-analysis of recent studies on haptic feedback enhancement in immersive-augmented reality. In: Proceedings of the 4th International Conference on Virtual Reality. ICVR 2018. ACM, New York, NY, USA, p. 3–9.
URL https://doi.org/10.1145/3198910.3198911
Kim, Y., Cha, J., Ryu, J., Oakley, I., 2010. A tactile glove design and authoring system for immersive multimedia. IEEE MultiMedia 17 (3), 34–45.
URL http://dx.doi.org/10.1109/MMUL.2010.5692181
Le, K.-D., Zhu, K., Kosinski, T., Fjeld, M., Azh, M., Zhao, S., 2016. Ubitile: A finger-worn I/O device for tabletop vibrotactile pattern authoring. In: Proceedings of the 9th Nordic Conference on Human-Computer Interaction. NordiCHI '16. ACM, New York, NY, USA.
URL https://doi.org/10.1145/2971485.2996721
Lee, J., Choi, S., 2012. Evaluation of vibrotactile pattern design using vibrotactile score. In: Proceedings of the IEEE Haptics Symposium. HAPTICS '12. pp. 231–238.
URL http://dx.doi.org/10.1109/HAPTIC.2012.6183796
Lee, J., Ryu, J., Choi, S., 2009a. Graphical authoring tools for vibrotactile patterns. In: Proceedings of the 3rd Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics '09. pp. 388–389.
URL http://dx.doi.org/10.1109/WHC.2009.4810912
Lee, J., Ryu, J., Choi, S., 2009b. Vibrotactile score: A score metaphor for designing vibrotactile patterns. In: World Haptics 2009 - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. pp. 302–307.
URL http://dx.doi.org/10.1109/WHC.2009.4810816
Lewis, J. R., 1995. IBM computer usability satisfaction questionnaires: Psychometric

evaluation and instructions for use. International Journal of Human–Computer Interaction 7 (1), 57–78.
URL http://dx.doi.org/10.1080/10447319509526110

Lewis, J. R., 2018a. Measuring Perceived Usability: The CSUQ, SUS, and UMUX. International Journal of Human–Computer Interaction 34 (12), 1148–1156.
URL http://dx.doi.org/10.1080/10447318.2017.1418805

Lewis, J. R., 2018b. The system usability scale: Past, present, and future. International Journal of Human–Computer Interaction 34 (7), 577–590.
URL https://doi.org/10.1080/10447318.2018.1455307

Liberati, A., Altman, D., Tetzlaff, J., Mulrow, C., Gøtzsche, P., Ioannidis, J., Clarke, M., Devereaux, P., Kleijnen, J., Moher, D., 08 2009. The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: Explanation and elaboration. Journal of Clinical Epidemiology 62, e1–34.
URL http://dx.doi.org/10.1016/j.jclinepi.2009.06.006

Martínez, J., García, A. S., Oliver, M., Molina, J. P., González, P., 2014. VITAKI: A vibrotactile prototyping toolkit for virtual reality and video games. International Journal of Human–Computer Interaction 30 (11), 855–871.
URL http://dx.doi.org/10.1080/10447318.2014.941272

Matulic, F., Ganeshan, A., Fujiwara, H., Vogel, D., 2021. Phonetroller: Visual Representations of Fingers for Precise Touch Input with Mobile Phones in VR. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. CHI '21. ACM, New York, NY, USA.
URL https://doi.org/10.1145/3411764.3445583

Menear, H., November 2020. Wearables could replace the smartphone sooner than you think. Mobile Magazine (Nov. 2020).
URL https://mobile-magazine.com/connectivity/wearables-could-replace-smartphone-sooner-you-think

Olsson, P., Nysjö, F., Singh, N., Thor, A., Carlbom, I., 2015. Visuohaptic bone saw simulator: Combining vibrotactile and kinesthetic feedback. In: SIGGRAPH Asia 2015 Technical Briefs. SA '15. ACM, New York, NY, USA.
URL https://doi.org/10.1145/2820903.2820925

Panëels, S., Anastassova, M., Brunet, L., 2013. TactiPEd: Easy Prototyping of Tactile Patterns. In: Kotzé, P., Marsden, G., Lindgaard, G., Wesson, J., Winckler, M. (Eds.), Human-Computer Interaction. INTERACT '13. Springer, Berlin, Heidelberg, pp. 228–245.
URL https://doi.org/10.1007/978-3-642-40480-1_15

Park, G., Choi, S., 2017. A physics-based vibrotactile feedback library for collision events. IEEE Transactions on Haptics 10 (3), 325–337.

Park, G., Choi, S., Hwang, K., Kim, S., Sa, J., Joung, M., 2011. Tactile effect design and evaluation for virtual buttons on a mobile device touchscreen. In: Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services. MobileHCI '11. ACM, New York, NY, USA, p. 11–20.
URL https://doi.org/10.1145/2037373.2037376

Park, T., Hwang, J., Hwang, W., 2009. Facilitating the design of vibration for handheld devices. In: Jacko, J. A. (Ed.), Human-Computer Interaction. Novel Interaction Methods and Techniques. Springer, Berlin, Heidelberg, pp. 496–502.
URL https://doi.org/10.1007/978-3-642-02577-8_54

Pezent, E., Cambio, B., O'Malley, M. K., 2021. Syntacts: Open-source software and hardware for audio-controlled haptics. IEEE Transactions on Haptics 14 (1), 225–

233.
URL http://dx.doi.org/10.1109/TOH.2020.3002696

Plaisier, M. A., Holt, R. J., Kappers, A. M., 2020. Representing numerosity through vibration patterns. IEEE Transactions on Haptics 13 (4), 691–698.

Pohl, H., Muresan, A., Hornbæk, K., 2019. Charting subtle interaction in the HCI literature. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, p. 1–15.
URL https://doi.org/10.1145/3290605.3300648

Ryu, J., Choi, S., 2008. posVibEditor: Graphical authoring tool of vibrotactile patterns. In: Proceedings of the IEEE International Workshop on Haptic Audio visual Environments and Games. pp. 120–125.
URL http://dx.doi.org/10.1109/HAVE.2008.4685310

Sani, A. A., Boos, K., Yun, M. H., Zhong, L., Jun. 2015. Rio: I/O sharing between mobile systems. GetMobile: Mobile Comp. and Comm. 19 (1), 6–9.
URL https://doi.org/10.1145/2786984.2786987

Schipor, O.-A., Vatavu, R.-D., Vanderdonckt, J., 2019a. Euphoria: A scalable, event-driven architecture for designing interactions across heterogeneous devices in smart environments. Information and Software Technology 109, 43–59.
URL https://doi.org/10.1016/j.infsof.2019.01.006

Schipor, O.-A., Vatavu, R.-D., Wu, W., Jun. 2019b. SAPIENS: towards software architecture to support peripheral interaction in smart environments. Proceedings of the ACM on Human-Computer Interaction 3 (EICS).
URL https://doi.org/10.1145/3331153

Schneegass, S., Poguntke, R., Machulla, T., 2019. Understanding the impact of information representation on willingness to share information. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, p. 1–6.
URL https://doi.org/10.1145/3290605.3300753

Schneider, O. S., Israr, A., MacLean, K. E., 2015. Tactile animation by direct manipulation of grid displays. In: Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology. UIST '15. ACM, New York, NY, USA, p. 21–30.
URL https://doi.org/10.1145/2807442.2807470

Schneider, O. S., MacLean, K. E., 2014. Improvising design with a haptic instrument. In: Proceedings of the IEEE Haptics Symposium. HAPTICS '14. pp. 327–332.
URL http://dx.doi.org/10.1109/HAPTICS.2014.6775476

Schneider, O. S., MacLean, K. E., 2016. Studying design process and example use with Macaron, a web-based vibrotactile effect editor. In: Proceedings of the IEEE Haptics Symposium. HAPTICS '16. pp. 52–58.
URL http://dx.doi.org/10.1109/HAPTICS.2016.7463155

Schönauer, C., Mossel, A., Zaiti, I.-A., Vatavu, R.-D., 2015. Touch, movement and vibration: User perception of vibrotactile feedback for touch and mid-air gestures. In: Abascal, J., Barbosa, S., Fetter, M., Gross, T., Palanque, P., Winckler, M. (Eds.), Human-Computer Interaction – INTERACT 2015. Springer International Publishing, Cham, pp. 165–172.
URL https://doi.org/10.1007/978-3-319-22723-8_14

Seifi, H., Zhang, K., MacLean, K. E., 2015. VibViz: Organizing, visualizing and navigating vibration libraries. In: Proceedings of the IEEE World Haptics Conference. WHC '15. pp. 254–259.
URL http://dx.doi.org/10.1109/WHC.2015.7177722

Shiraga, S., Kinoshita, Y., Go, K., 2016. Designing smartphone feedback based on vibration impression. In: Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems. CHI EA '16. ACM, New York, NY, USA, p. 3190–3196.
URL https://doi.org/10.1145/2851581.2892430

Shtarbanov, A., 2021. FlowIO Development Platform – the Pneumatic "Raspberry Pi" for Soft Robotics. In: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems. ACM, New York, NY, USA.
URL https://doi.org/10.1145/3411763.3451513

Siddaway, A. P., Wood, A. M., Hedges, L. V., 2019. How to do a systematic review: A best practice guide for conducting and reporting narrative reviews, meta-analyses, and meta-syntheses. Annual Review of Psychology 70 (1), 747–770.
URL https://doi.org/10.1146/annurev-psych-010418-102803

Song, J., Lim, J. H., Yun, M. H., 2016. Finding the latent semantics of haptic interaction research: A systematic literature review of haptic interaction using content analysis and network analysis. Human Factors and Ergonomics in Manufacturing & Service Industries 26 (5), 577–594.
URL https://doi.org/10.1002/hfm.20678

Swindells, C., Maksakov, E., MacLean, K., Chung, V., 2006. The role of prototyping tools for haptic behavior design. In: Proceedings of the 14th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. pp. 161–168.
URL http://dx.doi.org/10.1109/HAPTIC.2006.1627084

Swindells, C., Pietarinen, S., Viitanen, A., 2014. Medium fidelity rapid prototyping of vibrotactile haptic, audio and video effects. In: Proceedings of the IEEE Haptics Symposium. HAPTICS '14. pp. 515–521.
URL http://dx.doi.org/10.1109/HAPTICS.2014.6775509

Terenti, M., Vatavu, R.-D., 2022. Measuring the user experience of vibrotactile feedback on the finger, wrist, and forearm for touch input on large displays. In: Proceedings of the CHI Conference on Human Factors in Computing Systems Extended Abstracts. CHI EA '22. ACM, New York, NY, USA.
URL https://doi.org/10.1145/3491101.3519704

Turmo Vidal, L., Zhu, H., Waern, A., Márquez Segura, E., 2021. The design space of wearables for sports and fitness practices. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. CHI '21. ACM, New York, NY, USA.
URL https://doi.org/10.1145/3411764.3445700

Vatavu, R.-D., Mossel, A., Schönauer, C., 2016. Digital vibrons: Understanding users' perceptions of interacting with invisible, zero-weight matter. In: Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services. MobileHCI '16. ACM, New York, NY, USA, p. 217–226.
URL https://doi.org/10.1145/2935334.2935364

Vyas, P., Taha, F. A., Blum, J. R., Cooperstock, J. R., 2020. HapToes: vibrotactile numeric information delivery via tactile toe display. In: Proceedings of the 2020 IEEE Haptics Symposium. HAPTICS '20. pp. 61–67.
URL http://dx.doi.org/10.1109/HAPTICS45997.2020.ras.HAP20.34.8ad689d4

Wang, D., Guo, Y., Liu, S., Zhang, Y., Xu, W., Xiao, J., 2019. Haptic display for virtual reality: progress and challenges. Virtual Reality & Intelligent Hardware 1 (2), 136–162.
URL https://doi.org/10.3724/SP.J.2096-5796.2019.0008

Wenig, D., Schöning, J., Olwal, A., Oben, M., Malaka, R., 2017. WatchThru: expand-

ing smartwatch displays with mid-air visuals and wrist-worn augmented reality. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, p. 716–721.
URL https://doi.org/10.1145/3025453.3025852

Wobbrock, J. O., 2019. Situationally aware mobile devices for overcoming situational impairments. In: Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems. EICS '19. ACM, New York, NY, USA.
URL https://doi.org/10.1145/3319499.3330292

Yin, G., Otis, M. J.-D., Fortin, P. E., Cooperstock, J. R., Jun. 2019. Evaluating multimodal feedback for assembly tasks in a virtual environment. Proceedings of the ACM on Human-Computer Interaction 3 (EICS).
URL https://doi.org/10.1145/3331163

Youn, E., Lee, S., Kim, S., Shim, Y. A., Chan, L., Lee, G., 2021. WristDial: an eyes-free integer-value input method by quantizing the wrist rotation. International Journal of Human–Computer Interaction 37 (17), 1607–1624.
URL https://doi.org/10.1080/10447318.2021.1898848