FeedBucket: Simplified Haptic Feedback for VR and MR

Valentino Artizzu, Davide Fara, Riccardo Macis, Lucio Davide Spano
Department of Mathematics and Computer Science
Cagliari, Italy
tizzu93@gmail.com,davidefara@live.it,r.macis@live.it,davide.spano@unica.it

ABSTRACT

Standard development libraries for Virtual and Mixed Reality support haptic feedback through low-level parameters, which do not guide developers in creating effective interactions. In this paper, we report some preliminary results on a simplified structure for the creation, assignment and execution of haptic feedback for standard controllers with the optional feature of synchronizing an haptic pattern to an auditory feedback. In addition, we present the results of a preliminary test investigating the users' ability in recognizing variations in intensity and/or duration of the stimulus, especially when the two dimensions are combined for encoding information.

CCS CONCEPTS

• **Human-centered computing → Haptic devices**; **Mixed / augmented reality**; **Virtual reality**.

KEYWORDS

haptic feedback, mixed reality, virtual reality, haptic library

ACM Reference Format:
Valentino Artizzu, Davide Fara, Riccardo Macis, Lucio Davide Spano. 2020. FeedBucket: Simplified Haptic Feedback for VR and MR. In *Proceedings of the 2020 International Conference on Advanced Visual Interfaces, September 28 - October 2, 2020, Island of Ischia, Italy*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/1234

1 INTRODUCTION

The sense of touch offers a convenient way to provide feedback and additional information to users as a consequence of their actions or for notifying what is happening in the surrounding environment. This takes a particular interest in Virtual (VR) and Mixed Reality (MR) when experienced through head-mounted displays (HMDs). In such settings, the haptic feedback is typically provided to the user's hands through the controllers. Other dedicated devices exist, targeting specific body parts (e.g., head or chest) and offering fine-grained feedback through Galvanic Skin Response sensors, electro-stimulation and temperature control [3].

However, the standard support in controllers and development libraries for building the feedback provides only low-level parameters. This does not ease the developer work in creating effective and distinguishable feedbacks in different contexts i.e., it does not support the path of least resistance principle in developing User Interfaces [2]. For instance, Apple's Core Haptic [1] is a dedicated framework allowing developers to customize the haptic feedback and combine it with sounds, giving access to the intensity and frequency of the vibration. exploits a new degree of freedom in creating haptic patterns for Apple devices. Steam VR 2.0 and HTC Vive controllers support different vibration attributes for feedback [4], including intensity, frequency and strength thus allowing the execution of very detailed haptics. In this paper, we discuss the design of programmer-friendly API for creating haptic feedback using standard VR and MR controllers. We discuss the progress in the identification of the conceptual model and the results of a user study for identifying signals that are distinguishable by end-users. The goal is providing an out-of-the-box vocabulary reusable in the feedback implementation.

AVI 2020, September 28–October 2, 2020, Island of Ischia, Italy

ACM ISBN 123-4-5678-9010-4/20/1…$15.00
https://doi.org/10.1145/1234

2 THE FEEDBUCKET LIBRARY

The library we are currently developing for managing the haptic feedback is called FeedBucket. It extends the standard Steam VR 2.0 interface and it is designed for being simple and compact while fostering the adoption of the best practices in creating feedback patterns. In the current version, the library manages four types of feedback, namely i) haptic, ii) sound, iii) combination of haptic and sound and iv) sound with automatically generated haptic.

We structured the haptic feedback manager into a collection of customizable patterns associated with a boolean flag that, if true, requests the execution of the correspondent pattern starting a dedicated co-routine. After its execution the manager. When the execution has finished, the feedback manager sets it back to false. This design allows the user to access and edit the boolean value in any script in the virtual world definition, allowing triggering the haptic feedback from the different components in the code. While a feedback pattern is currently executed, the application might trigger another pattern. In this case, the former stops terminating the co-routine, and the manager starts a new one for the other pattern. A feedback pattern is a timed collection of feeds (the *FeedBucket*). A feed is single vibration with distinct attributes and/or a sound and it represents the basic unit of a pattern. More in detail, a feed instance contains the following information:

1. *Type*: the type of feedback among *haptic*, *sound*, *combined* and *generated* as discussed previously.
2. *Source*: the controller executing the feedback.
3. *Vibration Frequency* from 0 to 320Hz.
4. *Vibration Intensity* from 0 to 1. we use three qualitative shorthands that are low (0.4), medium (0.7) and high (1.0).
5. *Vibration Duration* in seconds. We introduced three shorthands which are short (0.1s) medium (0.15s) and long (0.2s)

(6) *Sound*: path to a sound asset
(7) *Threshold*: a parameter for the haptic from sound generation.

We can set pauses in a pattern, which are mapped simply to feeds of zero amplitude. The generation of haptic feedback in real-time from sound works as follows. While the audio is playing, we analyse the current spectrum summing-up the intensities. When we get a value higher than the one specified in the threshold parameter, we generate the haptic feedback using the values specified in the other feed structure fields. As an indication of possible values for the threshold, we suggest starting from 0.04 (we use it as a default value), but a fine-tuning of the feature requires some exploration on the chosen clip for selecting the best value. Once all the feeds have been defined, they composed in a pattern. Besides the list of feeds, the pattern contains also an optional *delay* field for postponing its execution after the triggering. Users can exploit a set of pre-defined patterns expressing different haptic sensations like a bump, saw, shake and so on. Finally, the Feedback Manager is the control point for the haptic feedback aspect in the application, and it is started at the beginning of the application execution. It receives the notifications from the different interactive elements and it launches all the co-routines associated with the feedback patterns. In addition, it manages the execution of each pattern, controlling the temporal distribution of the feeds and updating the pattern internal state accordingly. When the pattern requires the generation of the feedback according to a sound clip, the manager keeps the auditory and the haptic feedback synchronized, relieving the developer from this task.

3 USER STUDY

In order to determine usable values for the parameters in haptic feedback creation, we are planning different small studies. We aim at establishing which levels are correctly separable by users through standard controllers. We started by studying the interplay between the duration and the intensity while keeping the frequency constant. The test consisted of 10 tasks to carry-out in a virtual environment. For each task, the participant had to push two virtual buttons, each one activating a single haptic stimulus. Then, after having tried both feedbacks, we asked the participant whether the two vibrations were different or not and, if yes, in which dimension (duration, intensity or both). Before giving a final answer they were allowed to try each pair of buttons as many times as they liked. The pairs of stimuli were set as follows:

(1) High difference in duration (0.6s)
(2) High difference in intensity (0.7)
(3) Medium difference in intensity (0.2)
(4) Medium difference in duration (0.2s)
(5) No difference at all (intensity 0.6 and duration 0.6s)
(6) Duration and intensity swapped (0.5 and 0.7)
(7) Small difference in intensity (0.1)
(8) Small difference in duration (0.1s)
(9) Medium diff. in intensity, higher absolute value (1 vs 0.2)
(10) Very small difference in intensity (0.05)

Nine people participated in the test, 7 males and 2 females. The age ranged between 22 and 35 years old ($\bar{x} = 26.5$, $s = 4.5$). They had an average familiarity with VR applications ($\bar{x} = 2.6$, $s = 1.3$ in a 1-5 Likert scale) and a very good level of experience with videogames ($\bar{x} = 4.0$, $s = 1.1$).

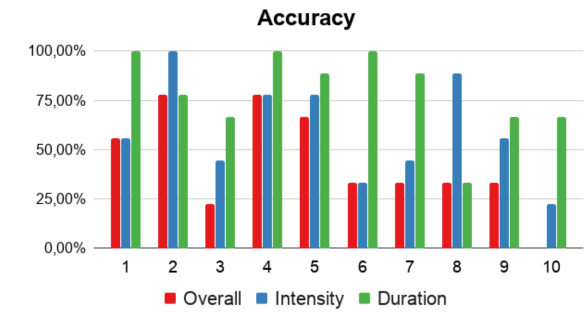

Figure 1: Participants' accuracy in recognizing the differences in the stimuli pairs. We report the overall accuracy and the values over the intensity and duration dimensions.

Figure 1 shows the test results. Even considering the small size of the sample, are interesting. The overall accuracy in distinguishing the differences was 69.4%, whit a higher precision on the duration parameter (78.9%) with respect to the intensity (60.0%). The analysis of the collected data and a qualitative evaluation of the participant's comments allow us to sketch-up some guidelines for a further development of the library. First of all, we noticed that intensity differences are harder to distinguish from duration ones. Besides, higher absolute values in the intensity magnitudes ease the recognition of the same difference level, if compared to lower ones. This suggests that we should not encode information on the intensity dimensions using linear scales. We can recommend some minimum values for supporting users in identifying a difference in two vibrations, which are 0.2 (in a 0 to 1 scale) for the intensity and 200ms for the duration. However, the results show that a difference in the stimulus duration alters the perceived level of intensity and vice-versa. This means that, if we would like to use the two dimensions as separate information channels, we must take into account the interplay between the dimensions, which usually translates in a lower number of bins for representing values.

4 CONCLUSION AND FUTURE WORK

In this paper, we reported some preliminary results in the design and the implementation of a simplified library for managing haptic feedback in VR and MR environments. The library aims at incorporating usable defaults for fostering the creation of more informative stimuli with standard hardware. Future work will expand the investigation on the perceivable levels and interplay between the dimensions in the feedback.

REFERENCES

[1] Apple. [n.d.]. Core Haptics. https://developer.apple.com/documentation/corehaptics. Accessed: 2020-04-17.
[2] Brad Myers, Scott E Hudson, and Randy Pausch. 2000. Past, present, and future of user interface software tools. *ACM Transactions on Computer-Human Interaction (TOCHI)* 7, 1 (2000), 3–28.
[3] Teslasuit. [n.d.]. Teslasuit software. https://teslasuit.io/software/. Accessed: 2020-04-17.
[4] Valve Software. [n.d.]. Class SteamVR_Action_Vibration Documentation. https://valvesoftware.github.io/steamvr_unity_plugin/api/Valve.VR.SteamVR_Action_Vibration.html. Accessed: 2020-04-17.