Oliver Schneider^{1,2}, Siyan Zhao¹, and Ali Israr¹

 ¹ Disney Research Pittsburgh, 4720 Forbes Avenue, Pittsburgh, PA, USA
² University of British Columbia, 201-2366 Main Mall, Vancouver, BC, Canada oschneid@cs.ubc.ca,{siyan.zhao,israr}@disneyresearch.com

Abstract. Despite ongoing research into delivering haptic content, users still have no accessible way to add haptics to their experiences. Lack of haptic media infrastructure, few libraries of haptic content, and individual differences all provide barriers to creating mainstream haptics. In this paper, we present an architecture that supports generation of haptic content, haptic content repositories, and customization of haptic experiences. We introduce *FeelCraft*, a software plugin that monitors activities in media and associates them with expressive tactile patterns known as *feel effects*. The FeelCraft plugin allows end-users to quickly generate haptic effects, associate them to events in the media, play them back for testing, save them, share them, and/or broadcast them to other users to feel the same haptic experience. The FeelCraft architecture supports both existing and future media content, and can be applied to a wide range of social, educational, and assistive applications.

Keywords: Feel effect; haptic authoring; haptic broadcasting; haptic media.

1 Introduction

In recent years, haptic feedback has shown promise to enhance user experience in movies, games, rides, virtual simulations, and social and educational media [3, 5, 7]. However, current mainstream media has yet to use the richness of haptic modality within its content. The lack of haptic authoring tools, production infrastructure, standardized playback protocols, and skilled and trained workers has contributed to the difficulty of integrating haptic content with accompanying media. To reduce the gap between haptics and mainstream communication, haptic feedback must be expressive, coherent, and synchronized with the content of the media, and also meet user expectations. We believe that allowing end-users to access, customize, and share haptic media will create an intimate, engaging, and personalized experience, and proliferate the use of haptics.

In this paper, we propose and implement an architecture that channels media content to dynamic and expressive tactile sensations. We introduce FeelCraft, a media plugin that monitors events and activities in the media, and associates them to user-defined haptic content in a seamless, structured way. The FeelCraft plugin allows novice users to generate, recall, save, and share haptic content, and

play and broadcast them to other users to feel the same haptic experience, without having any skill in haptic content generation. In the current implementation, we concentrate on the vibrotactile array as the source of sensation and the back as the surface for stimulation, however, the FeelCraft architecture can be easily adapted for other haptic feedback modalities.

We begin by presenting relevant background work related to haptic media infrastructure. We then present the FeelCraft architecture and describe in detail each component of the system. Finally, we conclude the paper with our envisioned application ecosystem.

2 Related Work

Infrastructures to integrate haptic feedback in media have been primarily derived from media type and user interactions associated with the content. These infrastructures fall into two categories: event triggers and direct mappings.

In the **event triggers** scheme, haptic information is embedded in the media and played back using predefined protocols [11]. A common example is a video game controller that rumbles on predefined triggers embedded in the games. This technique is complex and requires a proper production infrastructure that is expensive. Another disadvantage of this technique is that media designers would require access to tools and libraries for creating expressive haptic effects, similar to the libraries for visual and sound effects [1].

Direct mappings use cues from existing media and directly maps them to haptic effects. For example, a typical way to enhance movies and other visual content with haptics is to monitor activity in a video feed [12] and map these activities to haptic transducers arranged along the seat [3,13]. This way, movements in a visual scene are mapped to gross motion collocated with events seen in 4D movies and rides (www.d-box.com). Similarly, sound has been used to derive haptic cues for video games and music [2,10]. For example, Buttkicker technology (Guitammer, USA) shakes the entire seat using low-pass filtered sound. The Vybe Haptic Gaming Pad (Comfort Research) divides sound into three bands and maps them to transducers located in the seat and back. The advantage of the direct mapping scheme is that no change is required in the current media production process. However, each technique is limited to its media type.

Tactile icon libraries have been developed independent of media types. Immersion Corp. (www.immersion.com) has devised a library that could be used by designers, but it is limited to single actuators. Other researchers have proposed libraries and authoring tools [4,16] along with MPEG-V standardization protocols, but are limited to in-lab use and specific hardware [6,9,17]. Furthermore, customization and personalization have been found to be important to end-users [14,15], but most libraries do not support them. Recently, a library of *feel effects* (FEs, defined in [8]) show a promising approach to creating an engaging library. Of course, libraries require an infrastructure to connect to media before they can be widely applied.

3 FeelCraft Plugin and Architecture

A FeelCraft plugin maps media to haptic sensations in a modular fashion, supporting arbitrary media types and output devices. By using a FeelCraft plugin, users can:

- link existing and new media to the haptic feedback technology,
- use an FE library to find appropriate semantically-defined effects,
- author, customize, and share a common, evolving repository of FEs, and
- play and broadcast haptic experiences to one or more user(s).

A pictorial description of the FeelCraft architecture is shown in Figure 1. The conceptual framework of FeelCraft revolves around the FE library introduced in [8]. The FE library provides a structured and semantically correct association of media events with haptic feedback. By using the authoring interface to tailor FE parameters, a repository of FEs can remain general while being used for unique, engaging, and suitable sensations for different media. The playback system, authoring and control interface, Event2Haptic mappings and media plugin support seamless flow of the media content to the haptic feedback hardware.



Fig. 1. FeelCraft architecture. The FeelCraft plugin is highlighted in green. The FE library can connect to shared Feel Effect Repositories to download or upload new FEs. A screenshot of our combined authoring and control interface is on the right.

Media (1) can be entertaining, such as video games, movies, music, etc., or social and educational. The media can also be general user activity or embedded events in applications. In our implementation, we use the popular sandbox indie game Minecraft (https://minecraft.net).

Media Plugin (2) is a software plugin that communicates with the media and outputs events and activities. This plugin can be as simple as receiving

messages from the media or as complicated as extracting events and activities from a sound stream. With existing media, common plugin systems are automatic capture of semantic content from video frames [12], camera angles [3], or sounds [2,10], or the interception of input devices (such as game controllers or keyboard events). We use a CraftBukkit Minecraft server modification to capture in-game events.

Event2Haptic (3) mappings associate events to FEs, which are designed, tuned and approved by users using the FE library. This critical component links the media plugin's output to the haptic playback system. Currently, six FEs are triggered by six recurring in-game events: presence of rain, low player health, movement on horse, strike from a projectile, in-game explosions and player falls. Our implemention provides the option to store this mapping directly in the source code, or in a text-based JavaScript Object Notation (JSON) file.

Feel Effect (FE) Library (4) is a collection of FEs. A key feature of an FE is that it correlates the semantic interpretation of an event with the parametric composition of the sensation in terms of physical variables, such as intensity, duration, temporal onsets, etc. [8]. Each FE is associated with a family and semantically similar FEs are associated with the same family. For example, the Rain family contains FEs of *light rain* and *heavy rain*; as well that that of *sprinkle*, *drizzle*, *downpour* and *rain*. In our implementation, each FE family is represented as a Python source file that defines parametric composition of the FE and playback sequences for the FeelCraft Playback system, and each FE is coded as preset parameters in a JSON file. FE family files are necessary to play corresponding FEs in the family, and new FE families can be developed or downloaded through the shared FE repository. The FE can also be created, stored and shared. FE family and FE files are stored in a local directory of the plugin and loaded into FeelCraft on startup.

Authoring and Control Interfaces (5,6) allow users to create and save new FEs and tune, edit and playback existing FEs. Users modify an FE by varying sliders labeled as common language phrases instead of parameters such as duration, intensity, etc. (Figure 1). Therefore, users can design and alter FEs by only using the semantic logic defining the event. The interface also allows users to map game events to new FEs and broadcast to other users, supporting a What-You-Feel-Is-What-I-Feel (WYFIWIF) interface [14].

Playback and Communication Protocols (7) renders FEs using the structure defined in FE family files, and outputs them through a communication method (8) to one or more devices (9). Our implementation includes an API controlling the commercially-available Vybe Haptic Gaming Pad via USB.

4 Application Ecosystem

FeelCraft plugins are designed to make haptics accessible to end-users using existing media and technology. For example, a user may want to assign a custom vibration to a friend's phone number, or add haptics to a game. In this case, a user would download a FeelCraft plugin for their device, browse FEs on an online feel repository, and download FE families they prefer. Once downloaded, the FeelCraft authoring interface allows for customization, as a rain FE for one video game may not quite suit another game. The user could create a new FE for their specific application, and once they were happy with it, upload their custom FE for others to use. If the user wanted to show a friend their FE, they could use the playback system to drive output to multiple devices, or export the FE to a file and send it to them later. Figure 2 illustrates this ecosystem with application areas. Just like the Noun Project for visual icons (http://thenounproject.com) and downloadable sound effect libraries, we envision online repositories of FEs that can be continually expanded with new FEs by users. Our current FE repository includes six original families described in [8] and an additional four new families: *Ride, Explosion, Fall* and *Arrow*.



Fig. 2. Application ecosystem for FeelCraft and an FE Repository.

5 Conclusion

In this paper we have presented FeelCraft, a software plugin that allows users to author, customize, share and broadcast dynamic tactile experiences with media and user activities. The plugin uses *feel effects* (FEs), semantically structured haptic patterns. We integrated the FeelCraft plugin with a popular sandbox game, MineCraft. Users can associate six events in the game to corresponding FE, modify and broadcast to other users to share the haptics experience. The newly authored FEs are saved and shared with other users for communal use via an online FE repository. The proposed plugin can also be used with a wide range of entertaining, social, and educational media. Future work includes expanding the FE repository, networked communication and sharing, and supporting output to different haptic device types while maintaining semantic meaning, connecting end-users to haptics in an even more accessible manner.

References

- Cha, J., Ho, Y.S., Kim, Y., Ryu, J., Oakley, I.: A Framework for Haptic Broadcasting. IEEE Multimedia 16(3), 16–27 (Jul 2009)
- Chang, A., O'Sullivan, C.: Audio-haptic feedback in mobile phones. In: CHI '05 extended abstracts. pp. 1264–1267 (Apr 2005)
- Danieau, F., Fleureau, J., Guillotel, P., Mollet, N., Christie, M., Lecuyer, A.: Toward Haptic Cinematography: Enhancing Movie Experiences with Camera-Based Haptic Effects. IEEE MultiMedia 21(2), 11–21 (2014)
- Enriquez, M., MacLean, K.: The hapticon editor: a tool in support of haptic communication research. In: Haptics Symposium (HAPTICS '03). pp. 356–362. IEEE Comput. Soc (2003)
- Farley, H., Steel, C.: A quest for the holy grail: tactile precision, natural movement and haptic feedback in 3D virtual spaces. In: ASCILITE 2009. pp. 285–295. Australasian Society for Computers in Learning in Tertiary Education (ASCILITE) (Dec 2009)
- Gao, Y., Osman, H.A., El Saddik, A.: MPEG-V based web haptic authoring tool. In: IEEE International Symposium on Haptic Audio Visual Environments and Games (HAVE). pp. 87–91. IEEE (Oct 2013)
- Israr, A., Poupyrev, I., Ioffreda, C., Cox, J., Gouveia, N., Bowles, H., Brakis, A., Knight, B., Mitchell, K., Williams, T.: Surround Haptics: Sending Shivers Down Your Spine. In: ACM SIGGRAPH Emerging Technologies. Vancouver, Canada (Aug 2011)
- Israr, A., Zhao, S., Schwalje, K., Klatzky, R., Lehman, J.: Feel Effects: Enriching Storytelling with Haptic Feedback. ACM Transactions on Applied Perception (in Press) p. 17 (2014)
- Kim, J., Lee, C.G., Kim, Y., Ryu, J.: Construction of a haptic-enabled broadcasting system based on the MPEG-V standard. Signal Processing: Image Communication 28(2), 151–161 (Feb 2013)
- Lee, J., Choi, S.: Real-time perception-level translation from audio signals to vibrotactile effects. In: CHI '13. pp. 2567–2576. Paris, France (Apr 2013)
- 11. Modhrain, S.O., Oakley, I.: Touch TV : Adding Feeling to Broadcast Media (2001)
- Myongchan Kim, Lee, S., Seungmoon Choi: Saliency-Driven Tactile Effect Authoring for Real-Time Visuotactile Feedback. LNCS: Haptics: Perception, Devices, Mobility, and Communication 7282, 258–269 (2012)
- Bach-y Rita, P., Collins, C.C., Saunders, F.A., White, B., Scadden, L.: Vision Substitution by Tactile Image Projection. Nature 221(5184), 963–964 (Mar 1969)
- 14. Schneider, O.S., MacLean, K.E.: Improvising Design with a Haptic Instrument. In: Haptics Symposium (HAPTICS '14). Houston, USA (2014)
- Seifi, H., Anthonypillai, C., MacLean, K.E.: End-user customization of affective tactile messages: A qualitative examination of tool parameters. In: Haptics Symposium (HAPTICS '14). pp. 251–256. IEEE (Feb 2014)
- Swindells, C., Pietarinen, S., Viitanen, A.: Medium fidelity rapid prototyping of vibrotactile haptic, audio and video effects. In: Haptics Symposium (HAPTICS '14). pp. 515–521. IEEE (Feb 2014)
- Waltl, M., Rainer, B., Timmerer, C., Hellwagner, H.: A toolset for the authoring, simulation, and rendering of sensory experiences. In: Multimedia (MM '12). pp. 1469–1472 (Oct 2012)