Surround Haptics: Tactile Feedback for Immersive Gaming Experiences

Ali Israr[1]
Seung-Chan Kim[1]
Ivan Poupyrev[1]
{israr, kimsc, ivan.poupyrev}
@disneyresearch.com

Jan Stec[1,2]
janstec@gmail.com

[1] Disney Research
4720 Forbes Ave. Suite 110
Pittsburgh, PA 15213 USA

[2] Entertainment Technology Center
Carnegie Mellon University
700 Technology Drive
Pittsburgh, PA 15219 USA

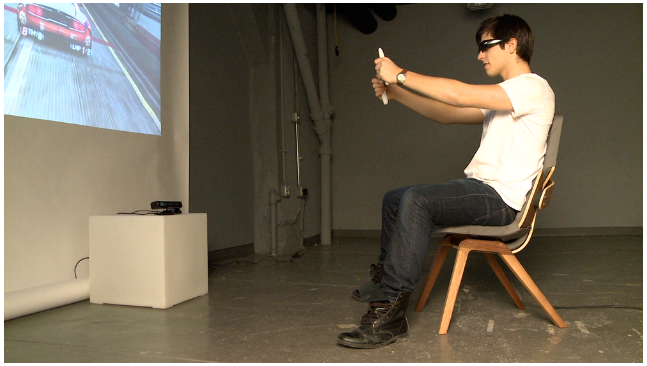

figure 1. A user playing a game while sitting on a chair equipped with the Surround Haptics system.

CHI'12, May 5–10, 2012, Austin, Texas, USA.
ACM 978-1-4503-1016-1/12/05.

Abstract
In this paper we propose an architecture for rendering rich and high-resolution haptic feedback on the user's body while playing interactive games. The haptic architecture consists of three main elements, namely, haptic engine, haptic API/codec, and haptic display. The haptic engine extracts events from the game, assigns haptic feedback to these events, and sends coded packets to the haptic API/codec. The haptic API/codec translates the coded packets and computes driving signals based on carefully evaluated algorithms derived from psychophysical modeling of tactile perception. The driving signals are then routed to the haptic display embedded with an array of vibratory transducers. A user feels high resolution and refined tactile sensations on the body through the display. We have integrated the Surround Haptics system with a driving simulation game to provide an enjoyable gaming experience.

Keywords
tactile feedback, interactive games, haptic API/codec

ACM Classification Keywords
H.5.2 [Information Interfaces And Presentation]: User Interfaces - Haptic I/O

General Terms
Design

Introduction

Designing tactile technologies that enhance the vocabulary of tactile "expressions" is one of the most important research directions in haptics and HCI today. Most of the available tactile technologies are not able to create detailed tactile sensations on large areas of human skin. For example, most tactile apparatus (e.g. chairs, controllers, mobile devices) provide either global vibrations or discrete pulses in several locations. There have been a few attempts to integrate them into complex, continuous tactile stimuli, but there are no formal guidelines of how to use this knowledge in practical embedded environments.

Surround Haptics is a new tactile technology that uses a low-resolution grid of inexpensive vibrating actuators to generate high-resolution, continuous, moving tactile strokes on human skin [1]. The user would feel smooth tactile motion, akin to what we feel when someone drags a finger across our skin rather than discrete tactile pulses and buzzes that are so common today.

Surround Haptics is based on carefully designed and thoroughly evaluated algorithms that use tactile illusions, e.g. Tactile Apparent Motion and Tactile Phantom Sensations, to create and move virtual actuators on a grid of actuators. A virtual actuator can be created using any pair of physical actuators (figure 2.1): the user would feel only a single vibrating point, not two separate physical ones. The algorithm is recursive and allows for the generation of virtual actuators using physical-virtual and virtual-virtual actuator pairs (figures 2.2, 2.3). This permits the creation of complex trajectories on skin using low-resolution actuator grids (figure 2.4) (e.g., see [2]).

The entire algorithm is based on rigorous psycho-physical experimentations and models of tactile perception [2,3]. It allows us to control speed, location, direction, length, thickness and intensity of the moving tactile pattern anywhere on the human body. We have integrated the Surround Haptics technology with interactive games, where users not only see and hear but also feel content of the game, simultaneously. Synchronized with stereoscopic visuals, spatial sound and gesture/ voice tracking schemes, the Surround Haptics system provides novel and immersive gaming experience to the user.

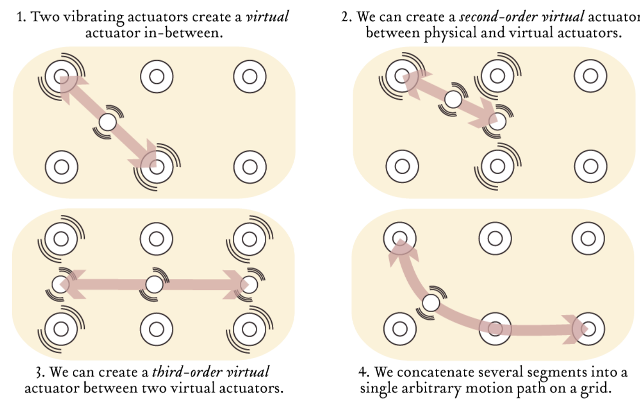

figure 2. Basic approach behind the Surround Haptics technology

In this paper, we propose and present an architecture for creating and controlling haptic feedback in interactive gaming environments, and present key components to render rich and high-resolution tactile feedback on the user's body.

Haptic Architecture

Figure 3 shows the block diagram of a typical gaming environment.

Figure 5. Snapshots of game events as the user maneuvers the car through different levels of the game, and feels internal states, such as acceleration, and external events (scraping, road bumps).

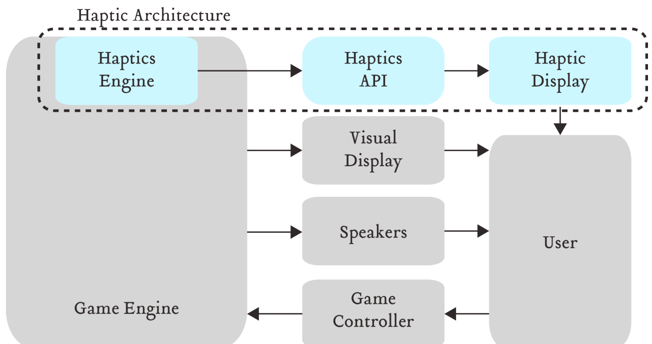

figure 3. Block diagram of a typical gaming environment and the haptic architecture.

The environment consists of a game engine that utilizes computing resources of a PC, gaming console, mobile platform, or web server. It computes features, events, and physics of the game, renders graphics and audio, and controls the input-output of the game. The game engine outputs visuals on a graphic display, such as on a projector, and audio through the speakers. A **haptic engine** residing in the game engine or operating separately computes haptics features and outputs messages to the **haptic API/codec** via a dedicated communication channel, such as the serial port. The API computes vibratory patterns using well-defined algorithms and guidelines, and creates driving signals for vibratory transducers placed in a **haptic display**. The display is mounted on a chair, vest, or placed in a hand controller that contacts the user's skin.

Haptics engine utilizes resources of the game engine to: a) extract events from the game and associate haptic feedback to them, b) manage, generate and prioritize haptic contents, and c) generate and send coded packets to haptic API. The game contents and haptic feedback are classified into four categories.

- *Egocentric feedback*, which informs the user with the internal state of the game character, such as if the car is accelerating, decelerating, slipping, etc.
- *Environmental feedback*, which informs the user about environmental conditions, such as road bumps, raining, scraping, etc.
- *Interactive feedback*, which informs the user of interactions with other characters of the game, such as bumping into other cars, cars passing by closely, shooting, etc.
- *Background and novel experiences*. These are feedback not actually associated with realistic feelings but add excitement to the game. For example: engine throttle, falling and flying, low-fuel state, etc.

Based on the type of events and haptic feedback, the haptic engine sends out low-level coded packets to the haptic API/codec via serial port. Four types of features are defined in each packet.

- *Pattern type*: dots, line, curve, circle.
- *Pattern mode*: trigger once or continuously run in a loop.
- *Pattern parameters*: location, thickness, strength, speed, direction.
- *Pattern quality*: soft, light, fluttery, penetrating, heavy, etc.

Haptic API/codec receives packets from the haptic engine and generates vibratory signals to output to the

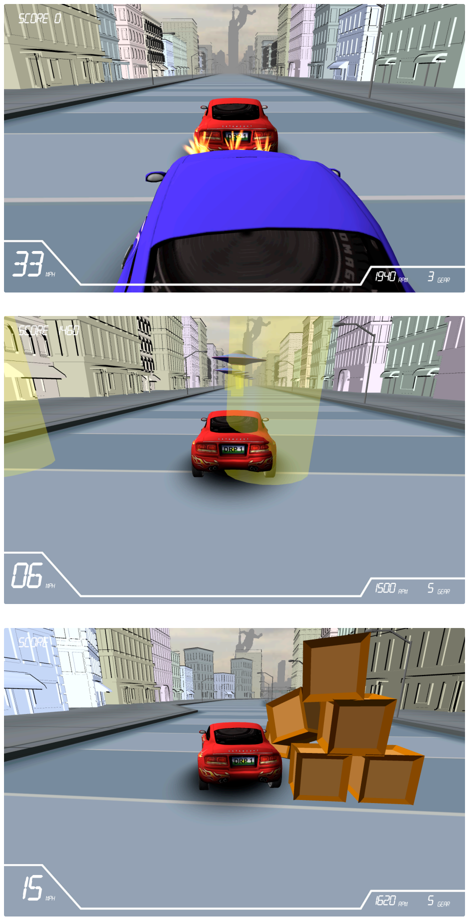

Figure 6. Snapshots of gaming events as the user maneuvers the car through different levels of the game and interacts with other characters and objects.

haptic display. The following information is either uploaded or hard coded into the API:

- Type, number and location of transducers in the haptic display
- Location of the stimulating skin
- Type of the display (chair, vest, pad, handheld)
- Psychophysical database library
- Tactile Brush algorithm [2]

The Tactile Brush algorithm is the basis for haptic API. The API computes timings, intensity, frequency, waveform for transducers embedded in the haptic display and outputs the signal through digital-to-analog converters. The API updates at 4 kHz rate.

Haptic display consists of an array of vibratory transducers arranged in pattern and a built-in power amplifier, and is mounted on a device that touches the human body. In our case, we place the array in a soft-pad, and mount it on a custom-made plywood chair (figure 4). The chair is carefully designed to conform around a user's body of different sizes. The transducers are placed on the back and seat of the chair in order to provide localized, rich and multidimensional tactile feedback to the large skin surface in contact with the chair.

Immersive Gaming Experience

We used Surround Haptics to design and explore immersive, video-tactile-audio gaming environments. We wanted to replace common vibrating tactile game controllers with a technology that delivers highly detailed, dynamic and spatial tactile experiences tightly synchronized with game events and interaction.

figure 4. A haptic display, a pattern of vibrating transducers arranged in a pattern in a pad, is mounted on the custom made gaming chair.

We developed a driving simulation game in Unity 3D: Game Development Tool (http://unity3d.com/) and integrated it with the haptic architecture to render tactile feedback associated with events of the game. The user sits on a chair embedded with a pattern of vibratory actuators on the back and seat, and maneuvers the car through different levels of the game.

A broad range of game events has been enhanced with spatial tactile strokes. These include collisions, road imperfections, tire traction, skidding, ripples of force when landing, acceleration, braking, objects falling on the car, damage, etc. (see figures 5 and 6)

References

[1] Israr, A. and Poupyrev, I. Exploring Surround Haptics Displays. In Proc. of the CHI, Work-in-progress, 2010. ACM. pp. 4171-4176.

[2] Israr, A. and Poupyrev, I. Tactile Brush: Drawing on Skin with Tactile Grid Display. In Proc. of CHI, 2011. ACM. pp. 2019-2028.

[3] Israr, A. and Poupyrev, I. Control Space of Apparent Haptic Motion. In Proc. of World Haptics Conference, 2011. IEEE. pp. 457-462.