Enhancing the use of Haptic Devices in Education and Entertainment

Development and evaluation of an Haptic Plugin for Game Engines

Nicolò Balzarotti

Enhancing the use of Haptic Devices in Education and Entertainment

Enhancing the use of Haptic Devices in Education and Entertainment

Development and evaluation of an Haptic Plugin for Game Engines

Doctoral Thesis

by

Nicolò Balzarotti

born on Thursday 25th February, 1993
in Magenta, Milano, Italy

Supervisor: Prof. Gabriel Baud-Bovy

Doctoral Thesis Committee: Prof. Antonio Frisoli
Scuola Superiore Sant'Anna

Prof. Domenico Prattichizzo
Università di Siena

Prof. Fabio Solari
Università degli Studi di Genova

Contents

1 Introduction

1.1 The weDRAW Project

The starting point for the research presented in this thesis was the two-years Horizon 2020 European Project "weDRAW", which was coordinated by Monica Gori (Istituto Italiano di Tecnologia). Within this project, our group was responsible for all tasks involving haptics. The central idea of the weDRAW Project was that "specific sensory systems have specific roles to learn specific concepts" (Volta et al., 2018). Starting from this idea, which came from a renewed understanding of the role of sensory modalities in development, the general objective of this project was to "create and evaluate a new methodology and a novel technology for deeper learning of numbers (time) and geometry (space)".

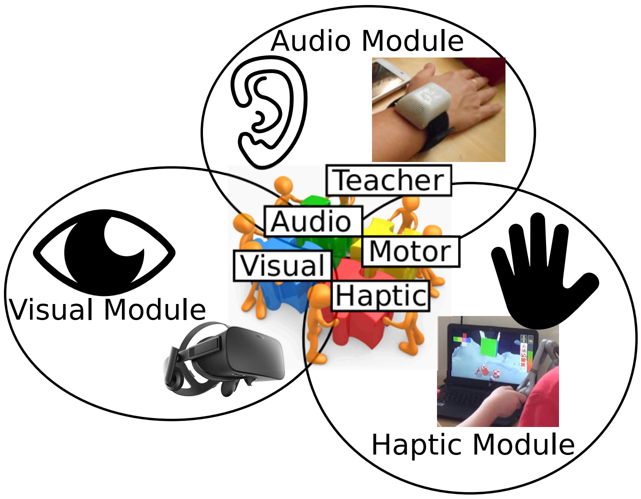

Figure 1.1: Which is the best sensory modality for learning? The weDRAW Project tries to answer this question creating a multimodal environment with an **audio**, a **visual** and an **haptic** module.

The weDRAW Project involved partners with complementary expertise; it was coordinated by the U-VIP group, headed by Monica Gori from the Istituto Italiano di Tecnologia, that led the development of the theoretical framework and the evaluation methodology in collaboration with Trinity College (Fiona Newell) and University College of London (UCL, Sara Price). On the engineering side, the consortium included Casa Paganini (Gualtiero Volpe and Antonio Camurri) from the University of Genova, which has a long experience with projects involving music, dance, and full-body movements sonification; University College of London (UCL) (Nadia Berthouze) with expertise in affective computing and automated analysis of emotions and social interactions, and two Small and Medium-sized Enterprises (SMEs) with experience in developing serious games (SGs) (LearnTPM from UK and Ignition Factory from France). Finally, the weDRAW Project also included the Chiossone Institute for testing the methodology with visually impaired children and two groups who contributed to the management and dissemination (De Agostini Editore from Italy and Dimitrios Karadimas from Greece). Our contribution is detailed in Section 1.1.1.

The project had multiple interrelated objectives such as “determining which is the best modality (visual, audio or haptic) to teach each specific concepts for the students”, “providing technology to exploit the best sensory signal” and, in particular, “developing Information and Communication Technologies (ICTs) platforms integrating multisensory interactive technologies” as well as developing “serious games (SGs) that will exploit the best learning modality for learning *arithmetic* and *geometrical* concepts” (weDRAW Proposal, 2017).

The main outcome of the weDRAW Project consisted in the development and evaluation of the following SGs and teaching activities (weDRAW Deliverable D6.1):

SpaceShape and BalloonGame SGs The primary objective of the SpaceShape game was to teach to young children fractions, 3D shapes and their transformations, including object rotations, cube elements (faces and vertices) and cube flattening. To familiarize children with the haptic device, the BalloonGame was developed: this game helps in understanding how the haptic device controls movements in a 3D space. Both games (described in Section 4.2.4) use a haptic device to render 3D objects virtually.

Cartesian Garden SG The primary objective of this game was to teach the Cartesian coordinate system (Volta et al., 2018). This game is played with visual and audio feedback (Immersive Virtual Reality with an Oculus head-mounted display).

Robot Angle and Fraction Activities The primary objective of these activities was to teach angle reflection, rotation, fractions and number estimation. These activities used a motion capture system (Microsoft Kinect) and the sonification modules of the EyesWeb platform (described in Section 3.4).

1.1.1 Thesis Contributions

The contributions of my thesis can be divided into three related parts. The first part consists in activities that were conducted *collaboratively* with the other partners of the weDRAW Project and that led to the development and evaluation of the SpaceShape game. The second part consists in the *development* of two software libraries that can be used to control haptic devices from other software platforms. The third part consists in *experimental studies*, addressing specific questions that emerged during the weDRAW Project.

A summary of this contributions is summarized in the following section.

1. Role of force-feedback in the weDRAW Project

   The general objective of my research was to integrate and to support the use of force-feedback technologies in the weDRAW Project, which required addressing a big question: *How to use force-feedback technology to teach Geometrical and Mathematical concepts?*

   This question was still far from being answered at the beginning of the project! In fact, the research activities during the first months of the project aimed at identifying the pedagogical concepts in *Mathematics* and *Geometry* that caused most problems to children and the manner in which the weDRAW multi-modal technology might help the learning. To that end, the members of the project organized workshops with educators and elementary school teachers, both in the United Kingdom and in Italy. The aim was to understand the current use of multiple sensory modalities in primary school education, and to get useful suggestions regarding the areas of difficulties in primary school students.

   To try to answer the question above, I first reviewed previous applications of force-feedback devices in education. This work was presented at a special weDRAW session at the International Conference on Multimodal Interaction (ICMI) in Scotland (2017). A summary of this work can be found in Section 3.1 and Section 3.2, and in a review paper (Baud-Bovy and Balzarotti, 2017).

   Second, I developed demos to illustrate the haptic technology to school teachers who participated to the workshops and collected their ideas and feedback. Those workshops helped the group in finding specific application fields and possible ways in which force-feedback could be used to develop SGs (Duffy et al., 2017). Altogether, this activity led to the creation over the next two years of the SpaceShape game, which uses the Phantom Omni haptic device (described in Section 2.1) to render 3D virtual objects.

Third, I did a technical review of available haptic devices and software; the aim was to find a haptic library that we could use as a base to control haptic devices, and an haptic device that was suitable for weDRAW Project's objectives. As SGs were targeted at children, the device should have an affordable price and the maximum force exerted shouldn't be too high to prevent injuries to children and damages to the device. A summary of this research is presented in Chapter 2.

Last, I developed HPGE, an Haptic Plugin for Game Engines, which was used by LearnTPM who developed the SpaceShape game itself in Unity3D. It is important to note that other partners, in particular Sam Duffy and Sara Price from UCL, played an important role in choosing the educational concept and defining the story-telling. In the last months of the project, the SpaceShape game was evaluated by training and testing children in Italian schools (U-VIP, IIT) and by collecting teacher feedback (UCL) as described in Section 4.2.4.

In a sense, the SpaceShape game is the *answer* of the weDRAW Project to the above question.

2. Software Development

At the beginning of the weDRAW Project, the requirements for the software libraries were not well defined. A general objective on our part of the work was to develop libraries that would be compatible with as many devices as possible and that could be easily integrated in various software platforms to promote the use of haptic and force-feedback technology in Virtual Reality (VR) and beyond. Two libraries were developed.

H4F Library. During the first year, I developed Haptic For Fun (H4F, see Section 4.1). The idea for this library, which stemmed from the project's proposal, was to integrate force-feedback devices in EyesWeb, a software platform (described in Section 3.4) developed by the University of Genova, which was at the center of the weDRAW Full Body platform (Camurri et al., 2000). In the weDRAW Project, the library was used to implement demos for the workshops.

HPGE Library. During the second year, I developed a Haptic Plugin for Game Engines (HPGE, see Section 4.2). The need for this library emerged because the weDRAW partners who would develop the SGs (LearnTPM and Ignition Factory) preferred working with Unity3D (see Section 3.3). This library took advantage of CHAI3D's sophisticated haptic algorithms to render 3D structures in a flexible and powerful manner (Section 4.2.1), which was useful for the SpaceShape game. HPGE is available on Github[1] and was first described at the International Conference on Games and Learning Alliance (GALA) (Balzarotti and Baud-Bovy, 2018a).

3. Experimental Studies

I conducted several experimental studies to address some of the issues that emerged in the weDRAW Project.

Multimodal shape perception in children. The goal of the study described in Chapter 5 was to investigate how different sensory modalities might be used to convey geometrical information to children in elementary school. To that end, I measured the capacity of children to discriminate the orientation ellipses of various eccentricities in the visual, audio and haptic modalities. The study also included a multimodal condition to investigate whether children integrate optimally the three sensory cues. The main results indicated that the performance of the children improved between 7-8 year-olds and 10-11 year-olds without reaching an adult level yet. The discrimination performance was worse in the audio modality and generally best in the visual modality. We did not find evidence that children or adult integrated different sensory modalities cues. In fact the performance in the haptic and multimodal condition were on average comparable and only slightly worse than in the visual condition.

Haptic perception of virtual textures. In the context of the development of the SpaceShape game, we thought that it might be important to be able to use virtual textures to enhance the difference between the surfaces of objects in the haptic modality like one might use different colors in the visual modality. This raised the question about how to simulate the textures so that the contrast from a perceptual point of view would be maximized. Haptic rendering algorithms are complex and many parameters can have an incidence on the interaction with the user. To answer this question, we report the results of two studies in which we used Multidimensional Scaling (MDS)

[1] https://github.com/HapticPlugin/HPGE-wrappers

techniques to identify the perceptual dimensions that correspond to the virtual textures that can be created with CHAI3D haptic rendering algorithms, and identify the parameters that have the largest impact (see Chapter 6). Results of the first study were published at the EuroHaptics conference (Balzarotti and Baud-Bovy, 2018b).

2 Haptic Devices and Software

Humans interaction with the physical world is guided by somatosensory information, that comes from two subsystems; *cutaneous* and *kinestetic* (Lederman and Klatzky, 2009). Those two subsystems are stimulated by contact forces and movements occurring during the exploration of objects, giving information about object's shape and temperature, the position of the object, the position of our body and its movements (Rincon-Gonzalez et al., 2011). The cutaneous system is sensible to vibrations and temperature, thanks to mechanoreceptors and thermoreceptors; mechanoreceptors embedded in muscles, tendons, and joints are sensible to joint position and applied forces and are part of the kinestetic subsystem. Damages to those systems, although rare, cause extremely severe impairments in everyday actions; constant visual attention and vigilance is required for every movement, including walking (where if the patient distracts might fall) and keeping objects in hands (Cole and Paillard, 1995). The cutaneous and kinestetic subsystems together form the *haptic* sense.

The term haptic derives from the Greek word *haptikos* (ἁπτικός), and refers to something that can be touched. The hand is arguably the part of our body that we use preferentially to explore the world with this modality. Haptic is not only used to refer to the human sense, but designates also technologies used to interact with the haptic system. Haptic devices are, generally speaking, devices that can stimulate the haptic system of a person interacting with them. Examples are shown in Figure 2.1. It is possible to distinguish between haptic devices on the basis of the haptic subsystem they stimulate, that depends on their working principle.

Haptic devices that stimulate the cutaneous subsystem include Peltier cells for thermal stimulation and vibrotactile devices. Those devices are generally *ungrounded*: that means that they do not need to be attached to the ground to transmit a force to the body. Vibrotactile devices are the most common kind of haptic devices available on the market (Figure 2.1, Left); they are used in mobile phones, in game consoles' joysticks and gaming chairs, and in full-body suites (Perret and Vander Poorten, 2018).

Figure 2.1: Example of haptic devices in commercial products. **Left**: Vibrotactile devices are used in everyday objects like mobile phones and game consoles. None of this devices is grounded. **Right**: Two force-feedback devices with different dimensions, force ranges and applications. Both devices are grounded. Images by Purism team CC BY-SA and Lega Nerd CC BY-NC-ND 4.0.

Haptic devices that stimulate the kinestetic subsystem are called force-feedback devices (Figure 2.1, Right); the interaction with the human body happens through forces produced both by the human and by the haptic device. Force-feedback devices, as opposed to cutaneous haptic devices, must be *grounded*. As determined by Newton's third law, in order to transmit an *external* force to the human operator they need to be attached to something external to the body, like a desktop or the floor. This way it is possible to create the perception of forces with an external reference frame, like an object's weight. This need limits devices' workspace (Hannaford and Okamura, 2008). Force-feedback devices are of particular interest in research fields like tele-operations, where forces sensed remotely are displayed locally through Virtual Environments (VEs).

2.1 Force-Feedback Devices

Force-feedback devices are actuated systems that provide a force-feedback to the user. They are also input devices, meaning that they provide information about the position and, possibly, orientation of the end-effector in space. Force-feedback devices complement 3D stereoscopic displays by allowing the user to touch, for example, virtual objects. Those objects can be endowed with physical properties such as weight, viscosity, hardness and texture. With those properties it is possible to

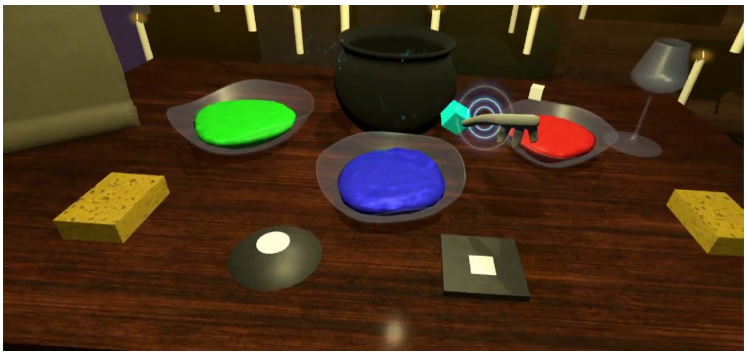

Figure 2.2: Hocus Pokus, an Immersive VR game developed at University of Bristol. The haptic device simulates physical interactions with the virtual world. The game has been built using Haptic Plugin for Game Engines (HPGE).

create a rich environment that simulates physical interactions in the real world. One example is the game Hocus Pokus[1] (developed by the University of Bristol, shown Figure 2.2) in which the user can grab objects feeling their weight and feel fluids' viscosity.

Typically, force-feedback is provided by a handle or a stylus. While some devices are equipped with handles that are able to transmit torques, the vast majority of commercially available force-feedback devices provides the force-feedback only along three directions, for complexity and cost reasons. The workspace is limited to movements of the wrist and/or of the upper arm in most devices because force-feedback devices must be grounded; bigger workspaces would require bigger mechanical systems, leading to more complex and expensive systems. Another obvious limit is that user interaction is typically limited to a single interaction point, which considerably differs from the way we naturally touch and/or manipulate real objects. For example, an action as simple as grasping might require a second haptic device (or a special handle with an actuated gripper) to be realistically implemented.

Although first force-feedback devices appeared in late nineteen-sixties, devices became to be affordable and the feedback began to be accurate only in late nineteen-nineties (Stone, 2000). Since then, a growing number of companies have started to commercialize force-feedback devices. In the near past, the video game industry has offered force-feedback devices such as actuated steering wheels for driving simulator and force-feedback joysticks at low-cost (e.g., Microsoft SideWinder Force Feedback 2). Moreover, until recently, it was possible to buy the Falcon, a 3-Degree of Freedom (DoF) device costing less than $250 which targeted gamers and game

[1] https://www.youtube.com/watch?v=AgS_N7G4M9w

developers. However the fact that the production of these devices has been discontinued (Microsoft stopped producing the SideWinder Force Feedback in 2003) suggests that existing applications are not enough to make those devices commercially viable products. Current solutions include either low-cost open-hardware or expensive commercial products. For example, Stanford University has developed the "hapkit" handler, a 1D haptic device that can be built with a 3D printer and cheap on-the-shelf materials (Morimoto et al., 2014). Commercial solutions have different prices depending on the number of DoFs, force ranges and accuracy.

For this reason, most commercial companies focus on special markets, like research, medical, army and industrial use, where the price is less of an issue. Application fields include tele-operations, physical rehabilitation and VR. A crane operator or a surgeon might sense, for example, the weight of the object lifted or the resistance of the tissue to the penetration of a needle; astronauts in the space or researchers in remote places on the Earth could, in the future, "touch" their family when away for months (Guanyang et al., 2019; Hamza-Lup et al., 2019). Uses of force-feedback devices in education are still limited to research, as their price is still incompatible with a school budget. However, previous history suggests that if the demand is sufficient to justify the needed investments, force-feedback technology can be made available at a low price. If it is possible to demonstrate the potential benefit of the introduction of force-feedback devices in schools and education, the demand for this technology might hopefully be there to support the commercialization of low-cost mass-produced haptic devices.

Figure 2.3: Geomagic Touch (formerly SensAble Phantom Omni), an affordable 3-DoF Haptic Device, used by the weDRAW Project.

In our research we have used the Phantom Omni (Figure 2.3), a grounded force-feedback device. Its workspace is 160 mm (W) × 120 mm (H) × 70 mm (d). The sensed position has a resolution of ~0.55 mm. Peak force is 3.3 N with a continuous exertable force > 0.88 N. It has 6 inputs and 3 outputs DoFs (input: 3 position + 3 rotation, output: 3 position). With this device, it is possible to convey interesting physical concepts involving external forces, like Archimedes's force or gravity, and complex object properties like textures (see Section 4.1.4).

2.2 Haptic Software

When interacting with the physical world, interaction forces depend on the movement of our body and on the objects' physical and surface properties. Those same forces needs to be simulated in a VE.

We will refer to algorithms and all the software stack used both to simulate this environment and to control haptic devices with the expression *haptic software*. The problem that haptic software has to solve can be defined as follows:

> Given a configuration of the [physical] haptic device $\mathcal{H}$, find a configuration of the [virtual] tool $\mathcal{T}$ that minimizes an objective function $f(\mathcal{H} - \mathcal{T})$, subject to [virtual] environment constraints. Display to the [physical] user a force $F(\mathcal{H}, \mathcal{T})$ dependent on the configuration of the device and the tool.
>
> — Otaduy and Lin (2005) [...] = our addition

That is, given a *physical* system composed of a user and an *haptic device* ($\mathcal{H}$) and a *virtual* system composed of an environment and a representation of the haptic device in this environment (the tool, $\mathcal{T}$), an haptic algorithm must find the "best" way to minimize the distance between the device and its representation ($\mathcal{H} - \mathcal{T}$), and apply a force on the device to reach this objective.

Differences in possible haptic rendering algorithms lie in the definition of $f(\mathcal{H} - \mathcal{T})$ given. There are different rendering algorithms, all consisting of at least two modules: (1) one that determines the configuration of the tool, and (2) one that *detect collisions* in order to define *environment constraints*.

Those problems are computationally expensive to solve. Moreover, those problems are compounded by the fact that time is an important issue in haptic rendering: a stable control loop must run at least at 1 kHz frequency. In order to get realistic rendering of contact forces, a frequency up to 5 to 10 kHz might be needed (Choi and Tan, 2004). By comparison, the refresh rate of computer monitors ranges from 50 to 60 Hz up to 100 to 120 Hz in recent models. That means that, in comparison to typical 3D collision and visual rendering algorithms, haptic algorithms needs to be 8 to 200 times faster!

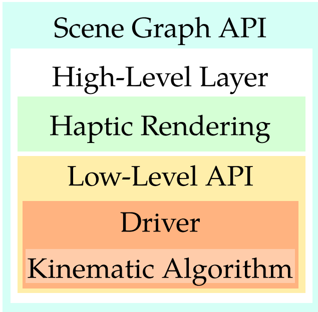

Figure 2.4: Haptic Libraries can be implemented with different abstraction layers (adapted from Kadleček and Kmoch, 2011). The lowest level is the driver that interface with the haptic device, while the highest level is the scene graph API, which manages objects hierarchy in the rendered scene.

Haptic libraries usually provide several features that make it easier to interact with haptic devices and to design haptic-enabled programs (see Figure 2.4).

This software includes low-level *drivers* that take care of reading the position of the device and sending force commands. To that end, these drivers include *kinematic algorithms* that solve the inverse dynamic problem to control device's actuators based on desired force and device's mechanical properties. Most haptic software also include *haptic rendering algorithms* that compute collision forces and other forces required to simulate the physical environment (e.g., gravity, spring forces, electromagnetic forces, friction etc.). Finally, haptic software might also provide a graphical rendering of the VE.

At the implementation level, a description of the VE is stored in the "scene graph". The scene graph is a tree-like structure that contains a hierarchy of objects where transformations like rotations, translations and scale changes are applied to each object's child. This structure is used both for the visual and haptic rendering. The scene graph simplifies the development (as it is possible to create complex objects created by different parts which are moved together), but might limit performances or, depending on the implementation of the Application Programming Interface (API), it can also limit developers' freedom. Due to different needs in refresh rates and computational complexity, separate threads are used for low-level control of the device, for physical simulations and for visual rendering.

Kadleček and Kmoch (2011) compared features of haptic libraries that were available at the time; an updated summary is presented in Table 2.1.

Table 2.1: Haptic APIs comparison
C = commercial, A = academic, * = partial

API	CHAI3D	libNiFalcon	HAPI	H3D API	OpenHaptics
Open Source	●	●	●	●	○
Cross platform	●	●	●	●	●
License	BSD3	BSD3*	GPL/C	GPL/C	C/A
Development state	●●○	●●○	●●●	●●●	●●●
API manual	○	○	●	●	●
API reference	●	●	●	●	●
Device range	●●●	●○○	●●●	●●●	●●○
Abstraction layer	High/Low	Low/Driver	High/Low	High	High/Low

One of the haptic libraries shown in Table 2.1 is CHAI3D. This library has many advantages, including the fact that it implements many haptic effects like magnetic field, texture rendering and viscosity, provides support for multiple haptic devices, and has cross-platform support (GNU/Linux, Windows and Mac OS X). Also, it provides a scene graph API, while leaving it easy to access to low-level object properties; moreover it is available under a free software license.

For those reasons, we've adopted it as a base for our work, that will be discussed in Chapter 4. The following part of this chapter briefly describes CHAI3D's structure and API, as described by its documentation[2].

2.3 CHAI3D

CHAI3D first launched in 2003 at the Robotics and Artificial Intelligence Laboratory at Stanford University (Conti et al., 2005) and is now maintained by Force Dimension[3]. It is written in `C++`, platform agnostic and supports a variety of commercially-available three-, six- and seven DoFs haptic devices. CHAI3D already provides support for SensAble devices including the *Phantom Omni*, and for Force Dimension devices, including *Delta*, *Omega*, and the *Novint Falcon* device.

CHAI3D's classes can be grouped in two large sets (shown in Figure 2.5). The first set (Figure 2.5 Left) provides device abstraction. The main class is `chai3d::cGenericHapticDevice`, which implements functions generally required to interface with haptic devices. Each specific haptic device must derive from this base class and must implement functions required to get device position

[2] `https://www.chai3d.org/download/doc/html/wrapper-overview.html`
[3] `chai3d.org`

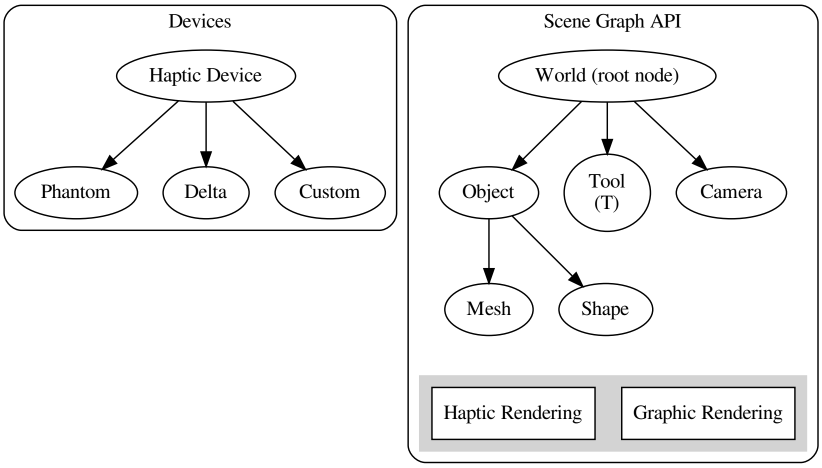

Figure 2.5: CHAI3D main components. **Left**: multiple device support is provided with a generic Haptic Device class. It is possible to add support for custom device by deriving from this class. **Right**: scene graph API implementation. A World can contain different kind of objects, including the haptic Tool (T) for haptic rendering and the Camera for visual rendering.

(`getPosition()`), to set the force and torque (`setForceAndTorque()`), and to access other device's features like the buttons (`getUserSwitch()`).

The second (Figure 2.5 Right) set provides classes used by the scene graph and needed to implement the VE. The scene graph is based on `chai3d::cGenericObject` class and includes the following subclasses:

Tools. The Haptic Tool ($\mathcal{T}$) represents the haptic device ($\mathcal{H}$), which can allow a single point interaction (`cToolCursor`) or a grip interaction (`cToolGripper`), depending on the hardware. Each tool is linked to a `cGenericHapticDevice`.

Meshes. This category includes `cMesh` and `cMultiMesh`. Those objects can be imported from mesh files of different formats.

Shapes. Simple geometrical primitives like `cShapeBox`, `cShapeSphere` and `cShapeCylinder`. Collision detection and haptic algorithms are specialized for those kind of shapes, so those objects are faster than the same shapes implemented as meshes.

Camera. `cCamera` includes information for graphical rendering, such as the projection matrix.

Each haptic object in the environment can either be a mesh or a shape. Shapes are geometrical shapes that can be described by simple equations. Contrarily, meshes are a collection of vertices and triangles that define the shape of a polyhedral object. While complex meshes can be imported from files, thus allowing the use of 3D object creation tools, the increased complexity of the object definition is reflected by increased complexity and availability of haptic rendering algorithms. For example, the "magnetic field" effect is not implemented on meshes, and collision detection is more expensive. Two collision detection algorithms are supported: Axis-Aligned Bounding Box (AABB) or a Brute Force method (Lin et al., 1996). Meshes haptic surfaces are rendered with the Finger-proxy algorithm as described by Ruspini et al. (1997).

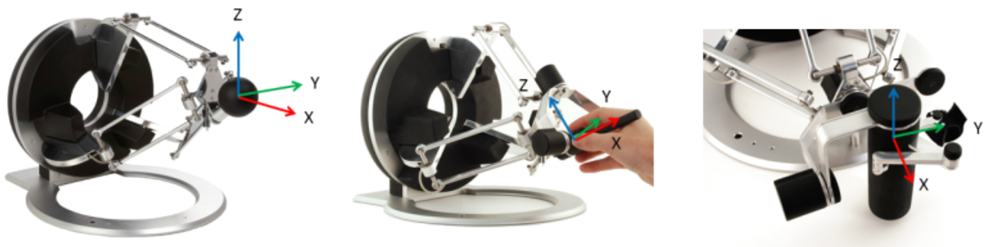

Figure 2.6: Device Coordinates in CHAI3D.

Objects in CHAI3D's VE are defined in a right handed reference frame (shown in Figure 2.6), a convention shared with programs like Blender, OpenGL and Godot, while Unity3D or first versions of DirectX use a left handed reference.

2.4 A Lightweight CHAI3D Version

As further explained in Section 4.1 and Section 4.2, we needed to integrate haptic rendering with other software platforms, including EyesWeb and Unity3D. Both those software platforms already provide means to visualize scenes and produce sounds. For this reason, we did not need CHAI3D's visualization and sound generation features. We used a lightweight version of CHAI3D, that was stripped from unnecessary components. We removed all the unneeded dependencies from `CHAI3D`, including `OpenAL` and `theoraplayer`, used for sound reproduction, and `glew`, used for graphical visualization. Without those dependencies, it's possible to compile the library also on an android device!

3 Virtual Environment and Virtual Reality

The term Virtual Reality (VR) was coined in 1989 by Jaron Lanier, who worked at VPL Research (Virtual Programming Language, USA, 1984), one of the first companies to develop and build VR products (Steuer, 1992). The term *virtual* means "being on or simulated by a computer"[1]. There is no general agreement on the definition of Virtual Reality (VR) and Virtual Environment (VE). For example, Blascovich does not distinguish between the two terms and defines VR as "an organization of sensory information that leads to perception of a synthetic environment as non-synthetic", thus including even audio recordings of musical performances in the definition. In contrast, Luciani makes a distinction between VR and VE based on the role of the human in this simulation; in VE the user is placed in front of the simulation (**vis-a-vis**) and can **manipulate** objects, while in VR the human is immersed **inside** the simulation and can **move** in the environment. In this thesis I'll use the terms VE and VR according to Luciani's definition.

In VE and VR, the communication between the real word (the *user*) and the virtual world (the *computer* system) happens by the means of *transducers* (shown in Figure 3.1). Transducers are physical systems that transform a digital representation of a world, simulated by the computer, into something that is perceptible by the human (*actuators* or *displays*), and that transform physical properties (e.g., sound pressure or visible light) into digital signals that are used to modify the virtual world (*sensors*) (Luciani, 2007). Transducers must be adapted to human sensory and motor abilities; specific devices are built to interact with each sense.

Various technological solutions exists for the different modalities. The visual system can be stimulated by using stereoscopic displays (monitor that can "transmit" different images to different eyes, creating the illusion of 3D), Cave Automatic Virtual Environment (CAVE, constituted by multiple projectors that project a 3D scene on up to 6 walls of a room), and head-mounted displays (one or two small displays

[1] https://www.merriam-webster.com/dictionary/virtual

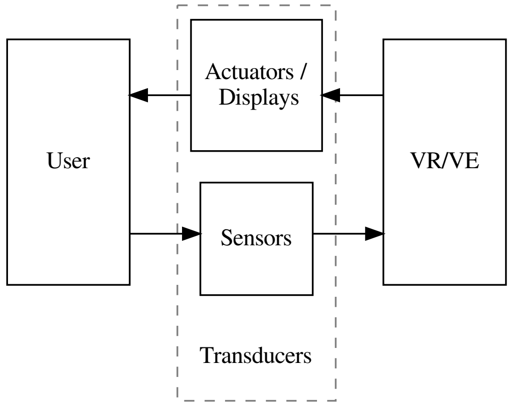

Figure 3.1: Interaction between *virtual* systems and the *user* happens by the means of *transducers*. Each sensory modality requires a transducer that brings information to the user, and one that brings information to the computer (e.g., headphones and a michrophone or a screen monitor and a video camera).

placed in front of one or each eye). The auditory modality can be stimulated by using headphones, speakers and spatialized speakers, or even cochlear implants. The haptic modality can be stimulated by using haptic devices (described in Chapter 2).

Common input systems include microphones (e.g., condenser, dynamic, piezoelectric), keyboards and mouses, and systems that can capture movements of the body (e.g., cameras and Inertial Measurement Units). Tracking movements is important to adjust the virtual scene rendered by the visual displays as a function of head movements in the VE.

Since the early and bulky headsets commercialized by VPL Research in 1985 (shown in Figure 3.2, Left), technological progresses have allowed the development of smaller and better headsets, which became popular in 2016 when *Oculus Touch* and *HTC VIVE* were released to the market. This led to an explosion of products for VR, to the point that many other companies (including Amazon, Apple, Facebook, Google Microsoft, Samsung and Mozilla) started working on their own VR headsets.

The implementation of VR headsets and VR software still poses some technological difficulties, including the need of synchronization between head movements and VR. Delays in the update of the virtual scene can cause "cybersickness", with symptoms including headache, nausea and vomiting (Davis et al., 2014).

Figure 3.2: **Left**: First VR headset from VPL Research, which has been described as a "motorcycle helmet [with a] visor [that] is opaque and protrudes abruptly at eye level" (Wright, 1987) **Right**: modern VR headset (Oculus Rift).

Given the objectives of the weDRAW Project and of my thesis, in the following sections I focus on applications of VR in education (Section 3.1) and on the role of haptics in VEs (Section 3.2).

3.1 Educational Virtual Environments

The possibility of using VR in an educational setting has been explored from the onset of VR (Helsel, 1992; Steuer, 1992). The use of VE in education is at confluence between three different but complementary ideas about how children learn.

First, the importance of the active exploration of the child for learning is a basic tenet of the constructivist theory of learning. This theory is based on Dewey's idea that education is driven by experience. Piaget expanded this theory, adding that children's knowledge of the world is based on exploratory interactions and that children's play and exploration are important parts of the cognitive development. Many have emphasized the educational benefits of being able to explore actively a problem space and to experiment (Hmelo-Silver, Cindy E., 2004).

In this respect, educational VEs might provide many additional benefits over real life situations. With VR it is possible to create various virtual worlds in a flexible way. (i) Those worlds can also be safer than reality, allowing children to explore controlled situations without risks. (ii) It is possible to experience things that are not possible in the real world. (iii) In addition to that, there is the possibility of providing additional feedback and/or information and of reducing the time needed to see the effect of their actions. (iv) Children can explore properties of the world and develop their intuition based on examples, facilitating the learning process from the specific to the general, and from the concrete to the abstract (Potkonjak et al., 2016).

As an example, De Jong et al. (2013) reviewed the use of physical and virtual laboratories, concluding that virtual laboratories could be cost-effective, offer the opportunity to use experimental systems that are beyond the reach for schools and enable students to investigate conjectures that are not possible in physical experiments.

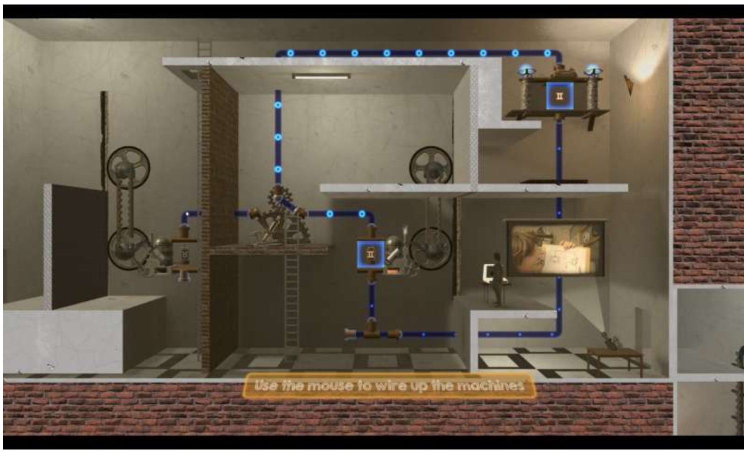

Figure 3.3: PlayStation Wired is a VE game that teaches electric circuits theory.

Second, VEs provide the possibility of including game-elements in non-game contexts (Huotari and Hamari, 2012), a concept often referred to as *gamification*. "Gameful" experiences support engagement, increase user activity and social interactions; as a consequence, gamification might play an important role in learning (Hamari, J. et al., 2014). Serious games, games integrating educational objectives with specific evidence-based game mechanics are known to support learning and its generalization (Breuer, Johannes and Bente, Gary, 2010; Djaouti, Damien et al., 2011). As an example of the role that gamification might have on learning, Merchant et al. (2014) did a meta-analysis of virtual-based games, simulations and virtual worlds in K-12 or higher educational settings. In general, games were found to be more effective than simulation or virtual worlds. However, only eight out of the sixty-nine analyzed studies targeted elementary grades (the age group I'm interested to in this thesis) and none of these eight included games. In commercial and non-commercial educational software, the opposite holds true. Many educational games exists, including Microsoft® Encarta®, GCompris, and many Nintendo games (e.g., Dr Kawashima's Brain Training), but only few of them make use of VEs, and even less make use of a 3D VR (Merchant et al., 2014; Mikropoulos and Natsis, 2011). This is in part explained by the fact that before the release of head-mounted displays in 2016 it was difficult to find a VR headset, and studies on education could only use in-house proprietary headsets (Johnson-Glenberg, 2018). An example of SG using a VE to teach electric circuits is PlayStation's Wired (shown in Figure 3.3).

Third, while games do not necessarily need to be technological, the technology used in VEs can also provide a *multi-modal experience*, which can convey information in a richer way and might also facilitate learning (Shams, Ladan and Seitz, Aaron R., 2008; Taljaard, 2016).

However, Mikropoulos and Natsis (2011) found that only a limited number of educational VE used multi-sensory interaction channels. More precisely, twelve (23 %) studies combined visual and auditory channels and only four (8 %) of the studies used haptic channel.

3.2 Haptics in Virtual Environments

Several reasons can be advanced to include force-feedback devices in educational VEs.

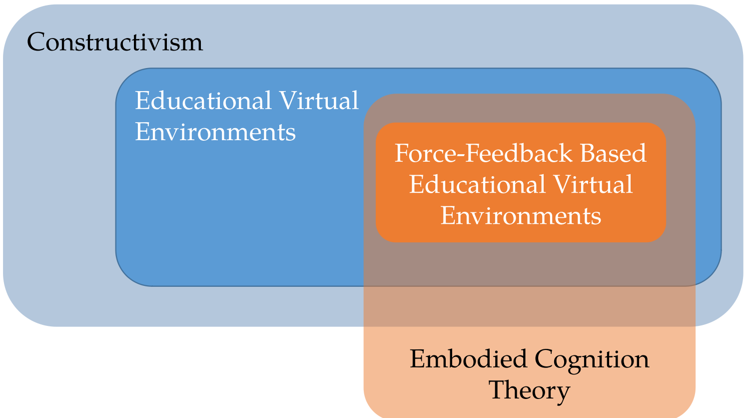

Figure 3.4: The use of force feedback in VEs is the conjunction between Constructivism and Embodied Cognition.

First, both the monitor screen and the mouse seriously constrain the user and the interactivity with the virtual scene (Nyarko et al., 2002). As force-feedback devices are input/output devices, they can provide an alternative to the mouse for interacting with VEs, thus increasing *immersiveness* and *affordance*, lowering the cognitive effort required to manipulate objects in the virtual world.

Second, the role of force-feedback might be particularly relevant for educational applications. As a matter of fact, as shown in Figure 3.4, force-feedback devices might be what's needed to constructivism to take advantage of *Embodied Cognition* in educational VEs. According to the *Embodied Cognition Theory* of learning, human cognition is rooted in the interaction between our perception and the external world (Barsalou, 1999). Our learning processes might be shaped by actions taken by our bodies (Hostetter and Alibali, 2008).

Yet, most of the VR simulations involve only visual and auditory feedback; for example, VR headsets integrate monitors and headphones but not force-feedback devices. As a matter of fact, the integration between haptic devices and VR has always been difficult. Wright (1987) describes Jaron Lanier itself as being a "believer" of the VR technology; however, he understood the limits that the technology had back then. Even if he already knew that it was possible to create the illusion of a contact with an object through vibrators, he said that it was still unclear how to simulate the "substance" of an object (i.e. the contact forces; Lanier proposed a malleable skeleton that could rigidify with electronic commands).

Indeed, VEs and haptics are two separate fields which developed independently. On one side, VEs have been extensively used in video games, but also for cognitive assessment, training and rehabilitation and for educational and for professional training. On the other side, haptic devices have initially been used in fields like physical rehabilitation and exergames (portmanteau of "exercise" and "games", games whose primary objective is to enhance or incentive physical movement). Attempts to join the two fields comes from both sides. The interest in using VEs in physical rehabilitation has increased; studies have shown how *gamification* might have a positive effect on motivation and interest (Proffitt, 2016).

The use of haptic devices in video games is not completely neglected; common game controllers like those of the Nintendo Switch, Sony PlayStation and Microsoft Xbox, to name a few, are able to produce vibro-tactile feedback. The Nintendo Switch™, for example, bundles two controllers with an haptic feedback that Nintendo® calls "Rumble HD". This is an "high definition" rumble that has been used by Nintendo in a game (part of their 1-2 Switch game) where the user has to move the controller and feel vibrations that simulate a box containing balls. The task is to guess how many balls are inside this box. This has even inspired an experiment with real objects (Plaisier and Smeets, 2017). Recently, Nintendo added "physicality" to some VR games. Nintendo Labo®, shown in Figure 3.5, let children build the physical part of the game with cardboard. Those objects are then used in the

Figure 3.5: Nintendo Labo®. Game controllers are build with cardboard, rubber bands and tape.

VR (creating what is called Mixed Reality) to interact with the game. This allows for cheap force-feedback devices, with the downside that forces are regulated only by physics (i.e. it is not possible to control them with the software). Some games available in arcades provide basic force-feedback in games like skiing, driving simulators and first person shooting games.

While the integration between haptic and VR has slightly improved today, more research is needed to have realistic force-feedback in VR simulations. The weDRAW Project, with its focus on multisensory integration and education, was a project that could reduce the gap between haptic research and VEs. In order to understand how we tried to simplify the use of haptic devices in VEs, described in Chapter 4, it is necessary to describe how VEs are usually created.

3.3 Game Engines

Game engines are Software Development Environments (SDEs) designed to build computer games. In modern video-game development, despite not being strictly required, they are a fundamental part of the process as they reduce the time and the expertise required to build a game. More than a hundred 3D game engines exist (such as Unity3D[2], UnrealEngine[3] and Godot[4]), but it is possible to identify common

[2]unity3d.com
[3]unrealengine.com
[4]godotengine.org

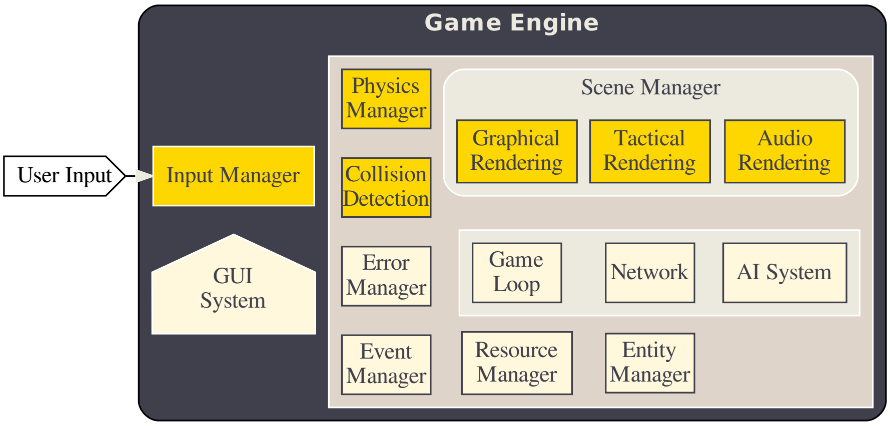

Figure 3.6: Common features in game engines. Features implemented both in game engines and some haptic libraries are colored in orange. Unity3D implements all those features (adapted from Zarrad, 2018).

architecture designs and principles and their most important features (shown in Figure 3.6, Zarrad, 2018).

In addition to the features provided by haptic software and described in Chapter 2.2, like physics simulations and collision detection, game engines have advantages over haptic software that speed up application development (Paul et al., 2012). First, much of the development can be done via a Graphical User Interface (GUI), avoiding tedious implementation of low-level features to the game developer. Second, game engines provide state machines and a game loop with a predefined set of callbacks that make it easier to define objects' behavior and their interactions. Third, other features like Artificial Intelligence and Animators (skeletal animation and morphing) further simplify the application development. Fourth, features implemented both in game engines and in some haptic libraries (shown in orange in Figure 3.6) usually are more advanced in game engines. For example, while both software are built on similar graphical API such as OpenGL or DirectX, game engines provide high level interface to configure object rendering (e.g., deferred rendering, shading, anti-aliasing), a large set of materials, lights or ways to create and edit them. Game engines also have more advanced graphical pipelines, and support more kind of input and output devices: both allows reading input from keyboards and mouses but joysticks and head-mounted displays support is usually missing in haptic libraries.

Finally, many if not most VR applications are also developed with game engines, and the community of developers using game engines is one or two orders of magnitude larger than the community of haptic device users, making it easier to find resources and get help.

For all these reasons, using a game engine to develop haptic-based applications would be a considerable benefit. However, game engines don't support force-feedback devices in a standard way (but see Cuevas-Rodriguez, Maria et al., 2013). Connecting a force-feedback device to a game engine is challenging because of the complex algorithms that need to be implemented to simulate contact with virtual object and the hard real-time constraints described in Section 2.2. In fact, to our knowledge, games engines have never been used to develop haptic based VR applications for educational purposes (see review by Baud-Bovy and Balzarotti, 2017).

One logical way to use a force-feedback device with a game engine is therefore to integrate an existing haptic library with the game engine. Our work focused on Unity3D, a proprietary cross-platform game engine with a very large developer base. Its Personal Edition is gratis (free to use) for non-commercial applications. Its Integrated Development Environment (IDE) allows a user to develop the game using built-in tools and the programming languages `C#` and `JavaScript`. Physical simulations are performed using `PhysX`, a `C++` physics engine developed by NVIDIA that can take advantage of hardware capabilities of the graphical card.

3.4 EyesWeb

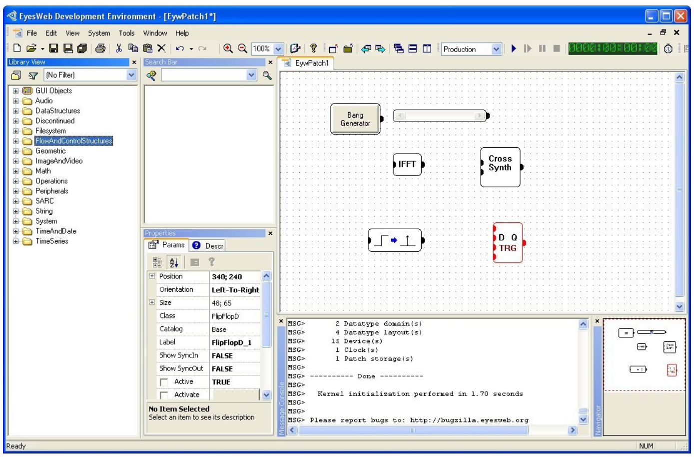

Figure 3.7: The EyesWeb GUI (version 5.3.0).

EyesWeb[3] XMI (eXtended Multimodal Interaction, show in Figure 3.7) is a gratis (free to use) Windows-only software platform focused on data generation and analysis that can be used for interactive multimodal installations (Camurri et al., 2000, 2007). It is an open platform built to support the design and development of real-time multimodal systems and interfaces. Various standards can be used for inter-module communication, including Open Sound Control (OSC), MIDI, FreeFrame and VST plugins, ASIO, Motion Capture standards and systems (Qualisys). It also provides an open set of reusable software components.

Even though EyesWeb is not a game engine it includes a number of features found in Figure 3.6. For example, it supports a wide number of *input devices* including motion capture systems, various types of professional and low cost video cameras, game interfaces (e.g., Kinect, Wii), multichannel audio input (e.g., microphones), analog inputs (e.g., for physiological signals). It also includes an IDE with a *GUI*, shown in Figure 3.7, which may be used to develop distributed or *networked* real-time applications. Supported outputs include multichannel audio, *video*, analog devices and robotic platforms.

4 Software Development

This section describes two haptic libraries that we developed for the weDRAW Project. Both libraries provide a plugin for other software platforms, that allow developers to include force-feedback devices in their applications.

The first target platform was EyesWeb, developed by the University of Genova, and the second platform was Unity3D. Both libraries use some CHAI3D components, described in Section 2.3.

4.1 First Haptic Library – Haptic For Fun

The aim of Haptic For Fun (H4F) was to provide an interface to control force-feedback devices from EyesWeb, a software developed by University of Genova which was used to develop some games in the weDRAW Project. The communication with EyesWeb has been implemented by using Open Sound Control (OSC), a standard communication protocol supported by EyesWeb. Although written in `C++`, the library also exports `C` API to make it easier to use it with other software. To communicate with haptic devices, the library uses CHAI3D's device abstraction layer, making Haptic For Fun (H4F) compatible with all devices supported by CHAI3D; however, this is the only part of CHAI3D used by the library: we implemented haptic rendering algorithms as described in Section 4.1.1. This library also permits to control an hardware extension to the Phantom Omni device; this extension includes two Light-Emitting Diodes (LEDs) and a pair of speakers for co-localized multisensory experience. Details are given in Section 4.1.3.

This chapter describes the functions of the library and illustrates some application examples.

4.1.1 Implemented Haptic Effects

H4F includes basic effects with which it is possible to create various VEs (a summary of the effects are described in Table 4.1). The effects in H4F have been implemented specifically to develop applications described in Section 4.1.4.

Constraints effects are used to force a specific exploration path; the device can move at the desired speed, but the exploration is constrained on those shapes (e.g., functions drawn on the Cartesian plane). A Bezier interpolation can be used to approximate many curves. Those that cannot be created with a Bezier function like a circle or an ellipse were implemented with separate functions. Another kind of constrained exploration is the plane (e.g., the plane of the Cartesian Plane), or the wall, which limits the exploration to a half-space (3D space divided by a plane).

Constrained movements are used to create a constrained movement (to control the position of the end-effector), a 1D constraint can be passed to the `follow_path` function, along with the required movement time.

Contact forces are used to simulate the contact with an object. The simplest is a constant force field (e.g., electric field, gravity). The spring is a force proportional to the distance from a given point. The movement can also be constrained inside or outside specific shapes, like a cube or a sphere: the force is null outside of the object and increases as a function of the distance from the border when the end-effector penetrates the object.

4.1.2 OSC Device

The aim of the integration between the haptic library and EyesWeb was to exploit what was already available to provide a rich multisensory environment, as shown in Figure 4.1. To provide this integration, we added to CHAI3D support for the OSC protocol.

OSC is a network protocol commonly used in audio production, show control and musical performances. Presented in 2005 by Wright as a way to facilitate networked music projects, it is based on User Datagram Protocol (UDP/IP). The protocol is easy and can be used to transmit integers (i), floats and doubles (f, d), strings (s), bools (T, F), arrays ([,]) and binary blob (B) when needed. The implementation details are easy and can be implemented in less than 300 lines of code. Commands are sent to specific IP addresses; two sockets are needed to send and receive data. Actions can be specified by using a path; an example command can be `udp:127.0.0.1:3200/force ddd 1.0 0.0 0.0`.

OSC messages can be used to control a physical haptic device by reading its position and rotation, and manually producing forces that could be used to accept

Table 4.1: List of haptic `C` exported functions, including haptic effects along with their description; the effects were implemented in order to be able to create demos for the teachers' workshops

Function Name	Description
`init`, `terminate`	manage device connection
`stop`, `start`, `is_running`	manage a low-level high priority haptic thread
`is_switch_pressed`, `get_position` `get_rotation`, `get_force`	get device information
Constraints	
`multisegment`, `multiline`, `Bezier` `circle`, `ellipse`	1D constraint (free motion)
`plane`, `wall`	2D constraints (free motion)
`follow_path`	constrained motion
Haptic Effects and Objects	
`constant_field`, `spring`, `damper`	simple haptic effects
`sphere`, `cube`	define 3D object geometry
`add_object_property`	add objects haptic effects (surface, viscosity)

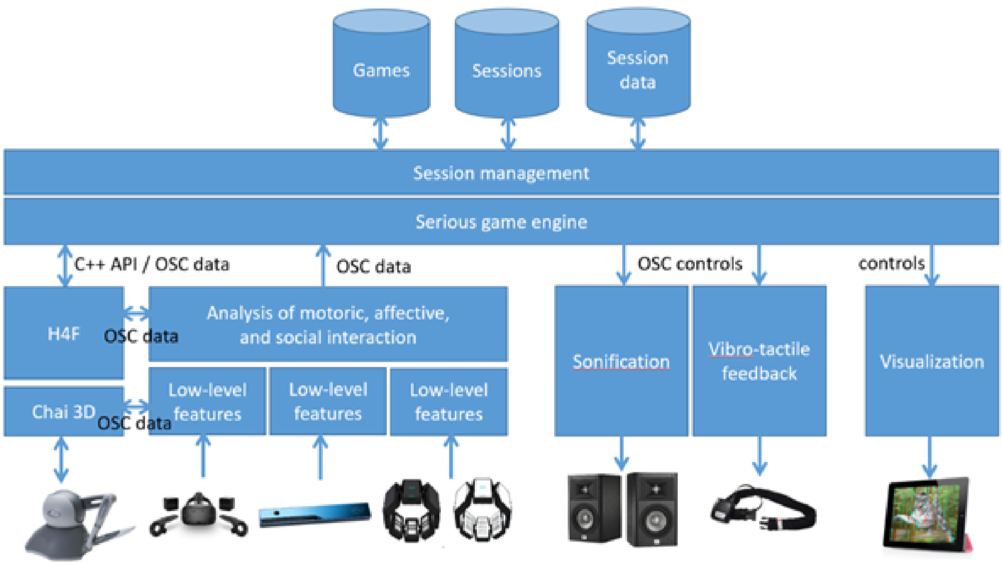

Figure 4.1: EyesWeb integration with other platforms in the weDRAW Project

Table 4.2: API exposed via OSC

Sender	/path types	description
Server	`/force ddd`	The OSC device receives 3d force commands. If the OSC device is virtual, force level could be visually rendered.
Client	`/force/advertise i`	Tell H4F to start advertising force information The changes are sent on force changes only, and only if the haptic thread is running.
Client	`/force/client_port i`	Tell H4F where (port) to send force information
Client	`/position ddd`	Send a change of 3D position to H4F. This could be simulated using a 3D joystick for example.
Client	`/rotation ddd`	Send a change in the stylus rotation to H4F.
Client	`/switch ii`	Send button state. The button will retain its state until you change it again. (button ids: 0, 1, states: pressed = 1, else 0)

commands via an OSC connection and send them to the real device. At the same time, if needed, it could read the actual device position and computed force and send it back to another OSC port. This feature could also be used without a real haptic device, to help in debugging rendering problems; the fake device position could be read by a mouse and sent via OSC device, and expected forces sent to another device for visualization. OSC could also be used to enable and disable haptic effects we implemented (described in Section 4.1.1). The APIs for OSC control are shown in Table 4.2.

To add OSC support to H4F, we extended CHAI3D's `chai3d::cGenericHapticDevice` class (described in Section 2.3) by adding a custom `chai3d::cOSCDevice` device that listens to OSC messages. The `C` library liblo[1] (released under LGPLv2.1) was used for the OSC implementation.

4.1.3 Haptic Add-on

We used the H4F library to run an experiment on multisensory perception in children, as described in Chapter 5.

Studies show that multisensory integration might be better when stimuli in different sensory modalities are co-localized (Neil et al., 2006). For that reason, we needed to have a way to provide co-localized multisensory feedback. We have designed and

[1] liblo.sourceforge.net

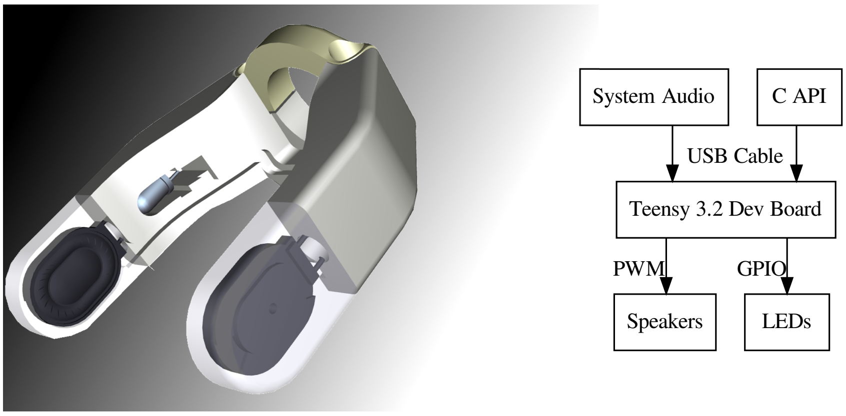

Figure 4.2: **Left**: 3D rendering of the add-on. The pair of speakers placed inside and one LED are visible. **Right**: Connection diagram for the haptic add-on. LEDs are controlled with a `C` API. The addon advertise itself as an USB audio card, so system audio output can be routed directly to it. A teensy board (arduino-like prototyping platform) drives the speakers and the LEDs. Power is supplied by the USB cable.

built an "add-on" for the haptic device, shown in Figure 4.3. This device is constituted by two parts: one is attached on the end effector of the haptic device; here are situated a pair of LEDs and a pair of speakers, used to provide a co-localized multisensory feedback to the user (Figure 4.2, Left). A cable connects this part of the add-on to a second part: a small box containing a Teensy 3.2 (an ARM 32 bit development board similar to an arduino) and an audio amplifier. This device is connected to the computer via an USB cable. This cable provides the add-on electrical power and is also used to transmit audio for the speakers and commands to turn on and off the LEDs. From a software standpoint, the Teensy advertises itself as an Audio card (Figure 4.2, Right) and as a MIDI device; MIDI commands can be used to control the LEDs status, while audio is routed throw the speakers. H4F provides a `C` API to send the required MIDI commands. Light intensity of the two LEDs can be controlled individually with 100 levels of intensity.

4.1.4 Applications

The effects we implemented in the library were those required for the following demos:

Draw the Symmetry The user is asked to draw the symmetry line of a 2D shape. The drawing is done by pressing a button on the haptic device. A feedback about the overall precision might be given (as a sound feedback or visual rating). A force field gives guidance about the axis of symmetry of the shape. The movements are constrained to a plane. An elastic force field could attract the hand toward the axis of symmetry: $f = -k(x - x_p)$ where x_p is the orthogonal projection of the hand position x on the axis of symmetry.

Adjust the Rod The user has to place a rod along the axes of symmetry. It can grab it by pressing the button; it follow his movements, and the rotation is done by moving the handle in the 3rd dimension (i.e. pulling against himself). Movements and forces are in 3D. The force in the plane parallel to the figure pushes the end-effector toward the axis of symmetry. The orientation of the rod depends on the position of the end effector in the direction that is perpendicular to the plane with the figure. An elastic force in the perpendicular can help to align the rod by attracting the end effector toward the desired orientation.

Cartesian Plane The user is given a 2D Cartesian's plane. One of different possible functions might be present (e.g., a line, a logarithm) and he is able to move only along the function's trajectory. This task can be done in two different modalities:

Exploration Task By feeling a force that varies with one of the two axes and that forces the user along the 3rd dimension.

Drawing Task The user has to draw the function (that is not drawn but just perceived). When pressing the button needed to draw the line, the force feedback is removed, so the user has to remember the function trend to be able to reproduce it correctly.

Archimedes In this game, the user has to find the couple of 3D shapes sharing the same volume. This is done by taking one object at a time (grabbing it with a button on the haptic device) and pulling it underwater. The user can, in this way, feel the Archimedes' Force (Buoyancy), and use it to infer the object's volume. The user can then place two object with the same volume one above the other. If the choice is right, they will disappear and new object will be created (and the user receive a given amount of points). The game can thus continue for a given time / until a score is reached.

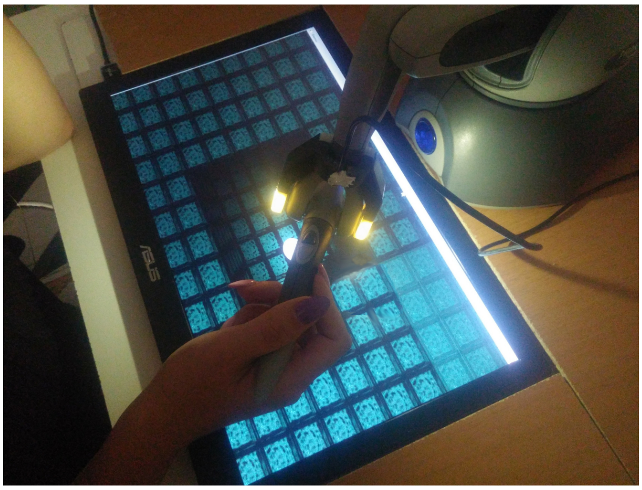

Figure 4.3: The `Multisensory Addon`: an addon for the Phantom Omni that produces sounds and emits light through two LEDs.

4.2 Second Haptic Library – Haptic Plugin for Game Engines

The content of this section is based on the paper "Hpge: an haptic plugin for game engines" that Balzarotti and Baud-Bovy published in 2018 on International Conference on Games and Learning Alliance.

The general aim of the HPGE library was to add support for force-feedback devices to the Unity3D Game Engine, for reasons described in Section 3.3. In particular, one of the objective was to be able to theoretically use any game engine with the library: although we provide script for Unity3D integration, the plugin can easily be adapted to other game engines.

Just like we did with H4F, we took advantage of CHAI3D's device support. This time, however, we also decided to take advantage also of its sophisticated haptic rendering algorithms, instead of implementing required effects by ourselves.

4.2.1 Integration Principle

Integration between Unity3D and CHAI3D happens by duplicating elements of Unity3D's scene graph for which haptic effects are enabled into objects in haptic plugin, taking into account differences in the coordinate system of the two software (see Figure 4.4). Such a duplication is necessary because the force feedback must be computed at a high frequency to insure stability. This way the dedicated thread running at 1 kHz in the plugin, which is responsible for haptic rendering, has access to objects' position, rotation and effect properties and is able to compute forces. Object's position and haptic properties are updated by Unity3D only when there are changes in `GameObjects` properties and at a lower frequency (see Section 4.2.2).

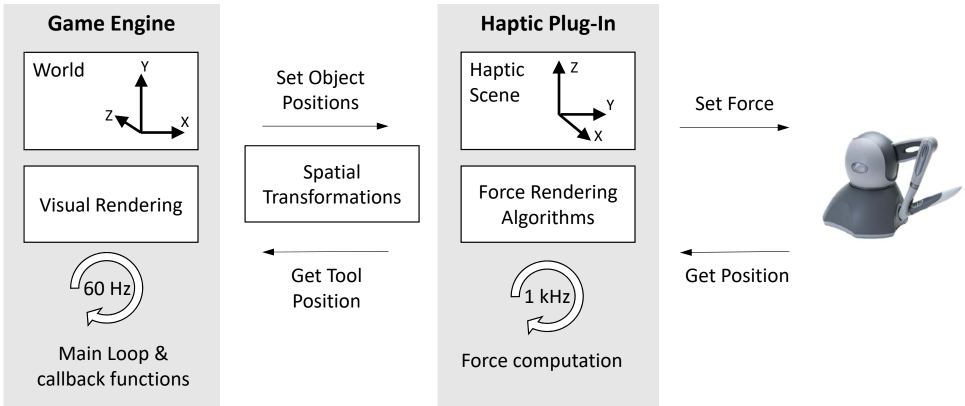

Figure 4.4: Unity3D (left) left-handed vs CHAI3D (right) right-handed world coordinate systems.

Making it easier to develop haptic-enabled applications might lead to a drop in the haptic devices price due to increased market request. This might bring more haptic educational SGs.

4.2.2 Implementation

Architecture. To make the haptic plugin compatible with a variety of game engines and software, we adopted a layered architecture for HPGE (see Figure 4.5).

- At the lowest level, `HPGE` is a `C` library that wraps a custom version of CHAI3D where unnecessary dependencies (OpenGL, OpenAL and Theora) have been removed. Those dependencies have been removed both to simplify the compilation process of the library on other platforms where those libraries might be missing (for example, we built it under Termux[2] on Android) and to reduce the binary size. The choice of a `C` API makes the integration with many programming languages particularly easy[3]. All required spatial transformations are implemented at this level.

[2] termux.com
[3] https://rosettacode.org/wiki/Call_a_function_in_a_shared_library

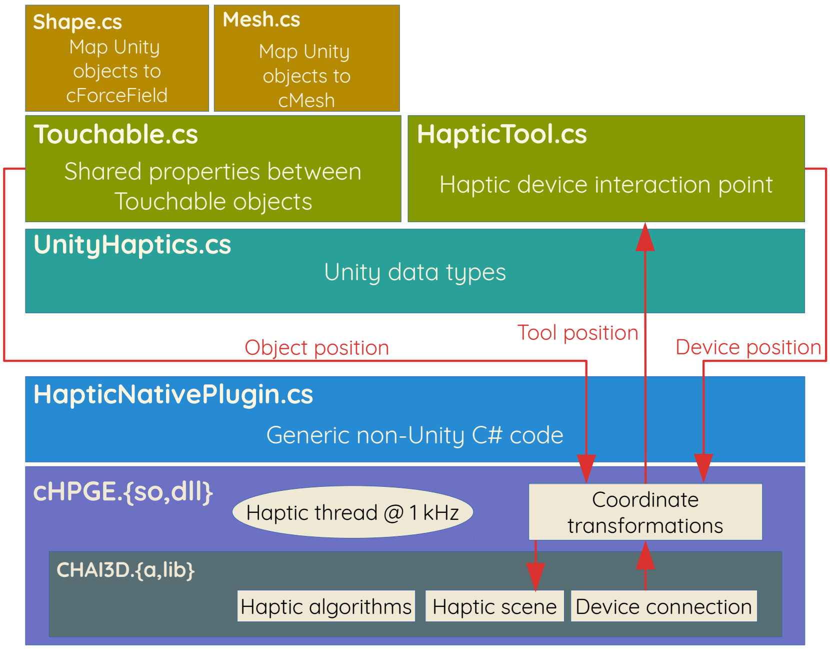

Figure 4.5: Architecture details of the integration between HPGE and Unity3D. Names on the left upper corner are the filenames.

- `HapticeNativePlugin.cs` is a `C#` wrapper that uses only `C#` data types in order to allow any `C#` program to use the library. This layer handles the conversion between native and managed code.
- The layers above (see Figure 4.5) are specific to Unit3D. `UnityHaptics.cs` takes advantage of Unity3D data types to simplify the `C#` API.
- At the highest level, the `HPGE` library contains several scripts that extend the `MonoBehaviour` Unity3D class (see Section 4.2.3).

Table 4.3: Scripts and function implementation details in Unity3D integration

`Unity3D C#` script	Method overridden	Comment
HapticTool.cs	Awake	Connect to the haptic device
	OnStart	Create a tool device in CHAI3D
	FixedUpdate	Update tool position and rotation in Unity
Touchable.cs	Start	Create equivalent object in CHAI3D
	FixedUpdate	Update object position in CHAI3D
	OnValidate	Update haptic parameter

The `HapticMesh.cs` and `HapticShape.cs` inherit from the `Touchable.cs` script. Table 4.3 indicates Unity3D callbacks that are overridden by these scripts. Details on execution order inside Unity3D are available on their manual[4]. The `OnValidate` callback is triggered when a corresponding property in Unity3D is changed in Unity3D IDE. The haptic parameters are updated on-the-fly, also when the game is running, which allows fast prototyping.

Interoperability. The plugin encapsulates CHAI3D haptic scene and all force rendering computation algorithms inside a dynamic library with `C` bindings (see Figure 4.5). Moreover, all spatial transformations between the game engine and the haptic device coordinate systems are implemented at this level (Figure 4.4). The `C` API allows one to manage the haptic scene that is used by the plug-in to simulate contact with virtual objects without having to deal with real-time constraint explicitly. Keeping all crucial computation in the lowest level of the plugin and the choice of a `C` API makes the integration with all sort software particularly easy. Exported `C` functions are described in Table 4.4. For example, we have used Python and Julia to control the haptic device and test the plugin during its development. In principle, one might call plugin functions from Unreal `C++` scripts to create new `Component` classes that can dropped on any `Actor` like the previously described C# scripts used in Unity3D to endow GameObjects with haptic properties.

Object position updating. Scenarios with moving objects (Animated Scenes) can present difficulties if the object moves to a new position that englobes the tool. In both cases, the movement might result in an abrupt change of the tool position with respect to surface of the object, which lead to an abrupt change in the interaction force. The problem is particularly acute if the object stiffness is high. This problem can however be mitigated by minimizing the size of the instantaneous displacements or, equivalently, by increasing the frequency at which the position is updated.

In Unity3D, the `Update` callback is called once per visual frame and runs typically at 60 Hz (which corresponds to 16.7 ms period), which is relatively slow for a quick object displacement. In principle, the `FixedUpdate` can be called several times for each `Update` callback (see the loop around physical simulation in left panel of Figure 4.6). To be able to take advantage of this possibility, the object (and tool) position in HPGE is updated inside the `FixedUpdate` callback.

The frequency of the `Update` and `FixedUpdate` can in part be controlled by Unity3D `Time Manager` parameters. However, experiments with a high-precision clock that measured the precise time of each callback revealed that the `Fixed`

[4] `https://docs.unity3d.com/Manual/ExecutionOrder.html`

Table 4.4: Description of C functions exported by the HPGE library

C Exported Function	Description
`get_version_info`	Get library version
`get_error_msg, last_error_msg`	Get info on errors
`init_logging`	Tells the library what to log
`start_logging`	Start logging data
`stop_logging_and_save`	Stop logging and save data to file
`tick`	Increment a counter in the log
`log_annotate`	Add note to a line in the log
`get_device_name`	Get info on connected haptic device
`(de)initialize` `is_initialized`	(De)Initialize connection with haptic device
`start`, `stop`, `is_running`	Control the haptic thread
`get_loop_frequency`, `get_loops` `get_log_frame`	Get info on the thread status
`set_hook`, `remove_hook`	Add a custom hook in the `FixedUpdate`

`get_tool_(force,position)`
`get_tool_(velocity,rotation)`
`get_tool_proxy_position`
`is_tool_button_pressed`
	Get info about haptic device

`create_mesh_object`
`create_sphere_object`
`create_box_object`
`add_object_to_world`
	Create new objects

`disable_object`
`enable_object`
`set_object_texture`
`set_object_material`
`set_object_position`
`set_object_rotation`
`set_object_rotation_euler`
`set_object_scale`
	Control existing objects

`object_exists`
`get_object_rotation`
`get_object_position`
	Get info about objects

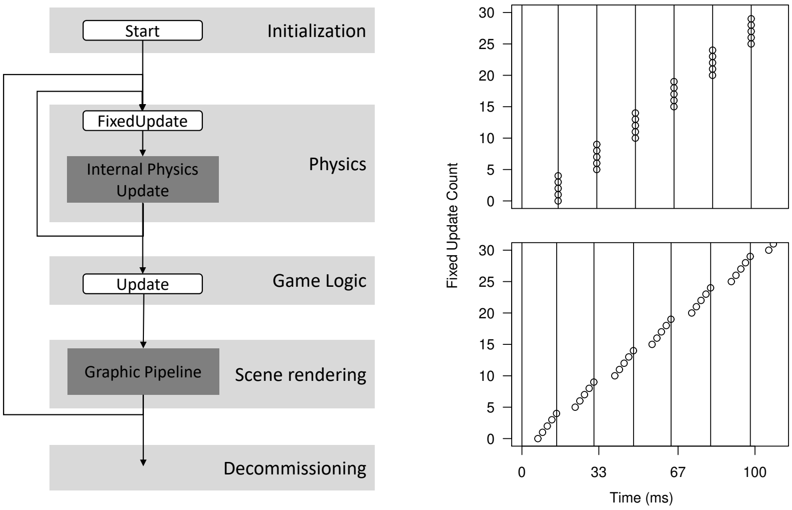

Figure 4.6: **Left**: Simplified representation of Unity3D control flow. Only callbacks (white panels) relevant for the plug-in are included. **Right**: `FixedUpdate` and `Update` frequency. The dots indicate the timings the `FixedUpdate` callbacks while the vertical lines corresponds to the `Update` callbacks. In this example, the `FixedUpdate` Period and Maximum Allowed parameters were set to 2ms and 10ms respectively in the `Time Manager`. Note that there are five `FixedUpdate` callbacks per `Update` callbacks and the actual period of the `Update` callback is about 16ms.

`Update Period` parameter in `Time Manager` did not always have the expected effect. For example, if Unity3D scene does not include any physical simulation, the `FixedUpdate` callback is called repeatedly without any delay within each `Update` callback period. The number of times the `FixedUpdate` callback, but not the timing, depends on the period set in the `Time Manager`. These measurements also revealed that the `Maximum Allowed` parameter did not influence the `Update` callback rate in this case. In order to space the `FixedUpdate` callback and obtain a smooth change of the object position, it is possible to include a small delay between the `FixedUpdate` callback, as shown in Figure 4.6, Right Bottom. Note that in this case the `Update` rate corresponded to the `Maximum Allowed` parameter in the Unity3D `Time Manager`. This example illustrate some of the challenges that might occur when time constraints are critical and the need to carefully measure and check that conditions that are necessary for a smooth rendering and/or control of the haptic feedback are met.

Another (complementary) approach is to smooth the changes of the position. This second approach was also implemented in the haptic plugin by interpolating the object position between the actual position and the new position over a user-settable time period. This approach has the effect of slightly delaying the change of position but can be used with relative slow update rates.

4.2.3 Usage

The range of applications and user experiences that a game might provide via the haptic plugin and device is very large. For example, the haptic plugin might allow the user to touch virtual objects, explore their shape and feel the properties of the surface such as its stiffness, friction or texture. In addition, the force-feedback device can also be used as 3D or 6D input device to move or rotate objects in a 3D VE or draw in space, for example.

The haptic plugin also provides interesting features for a game or application developer, with respect to more traditional haptic software.

Drag-and-Drop Support. For a game developer, the plug-in provides scripts that can be dragged-and-dropped on GameObject in Unity3D IDE to make them touchable (see Section 4.2.2). The position of the sphere that represents the handler is automatically updated when the user moves the device.

Graphical User Interface. All parameters of CHAI3D force rendering algorithm can be modified from the Unity3D GUI (see Figure 4.7). Those parameters include *surface effects* like *stiffness* and *texture*, *volumetric effects* like *viscosity*, *vibrations* and *stickslip*, and *magnetic* force.

Haptic Textures. CHAI3D implements a haptic texture rendering algorithm,which modifies the force experienced when one explores the surface of an haptic object. It is possible to define the haptic *texture* by associating a bitmap image in Unity3D IDE to the object as well as to dissociate haptic and visual textures.

Custom Haptic Force-Feedback. The library provides a hook function that is called in the main haptic loop to define a custom haptic effect. The hook function can be used to compute the desired force as a function of the position and the velocity of the device, which are passed as parameters. In this manner, a custom haptic effect can be defined without having to modify the haptic plugin. In the Hocus Pokus game (shown in Figure 2.2), this feature has been used to develop a custom effect for fluid rendering.

Device Logging. A logging mechanism allows the developer to save the position, velocity and force applied by the tool at the frequency of the haptic loop (typically 1 kHz). This makes it possible and easy to analyze the behavior of the user and the quality of interaction off-line. It is also possible to annotate custom event to keep track to game events in relation to user movements.

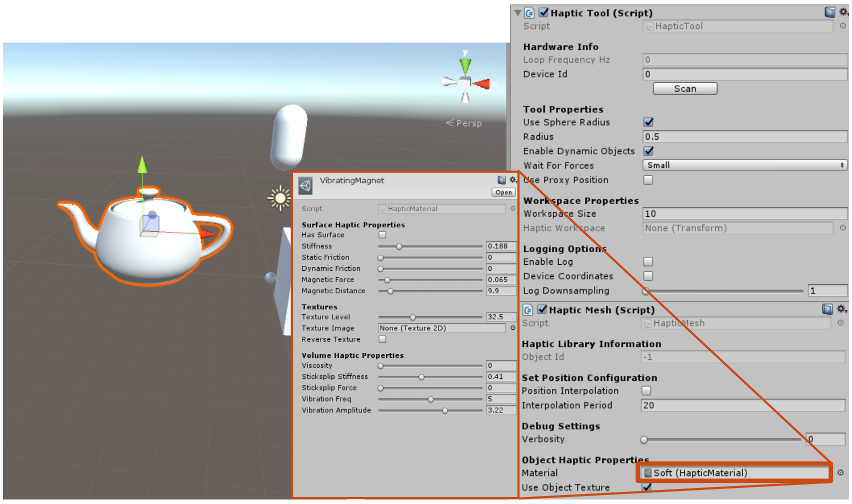

Figure 4.7: HPGE in the Unity3D GUI. Sliders can be used to change haptic properties in real-time.

The haptic scene inside the haptic plug-in is created automatically when the Touchable script is attached to a `GameObject` in Unity3D. Information about the object geometry information is recovered from the Unity `GameObjects` and automatically transmitted to CHAI3D to create a mirror object.

At a practical level, there are different kind of "scenes" depending on the interaction between the user and the game objects.

Static Scene. Static scenes are scenes where objects do not move. This is the simplest scenario for haptic rendering. In this case, force rendering depends only on the movement of the user on the touchable objects. For visualization purposes, the position of device is automatically updated in Unity3D to match the one in the haptic plugin via the `FixedUpdate` method.

Animated Scenes. Animated scenes are scenes where touchable objects are moved by Unity3D in a manner that is *independent* from the interaction force applied to the object. In other words, animated scenes do not involve physical simulation besides the contact between the user and the touchable object. In these scenes, the object position that is computed by Unity3D must be communicated to the haptic plugin (discussion with more details in Section 4.2.2).

Interactive Scenes. Interactive scenes involve scenarios with physical simulations where the action of the user has a dynamic effect on the motion of the objects. When a still object in Unity3D get touched by the haptic tool and a collision detector exists, the Unity3D objects starts moving thanks to the physic simulation done by the NVIDIA PhysX engine. This kind of interaction is more difficult than the previous two, and special attention must be used to have realistic feedback.

Unity3D Scripts. This section provides some information about `C#` scripts that have been developed specifically for the integration of the haptic plugin in Unity3D[5]. As mentioned in Section 4.2.3, the game developer can drag-and-drop these scripts on Unity3D `GameObjects` to endow them with haptic properties:

- `HapticTool.cs` is required and must be applied to a sphere representing the haptic device in the game (another `GameObject` with a different shape can be attached to the sphere in Unity3D for visualization purposes if desired).
- `HapticShadow.cs` can be used to make another object follow the position of the haptic device. It can be used to display either the position of the proxy ($\mathcal{T}$, the representation of the tool in the virtual world) or the real position of the haptic device. The difference between the two is that the proxy is constrained by the position of other touchable objects, while the real position moves freely. This script is useful for debugging purposes or force visualization (it's possible to trace a vector between the real tool and the proxy).
- `HapticMesh.cs` can be applied to any `GameObject` with a 3D mesh. The object is rendered with CHAI3D implementation of the *Finger Proxy Algorithm* (Barbagli, F. et al., 2004). The algorithm is suited for complex 3D shapes but the algorithm is computationally expensive. This is implemented inside the library using `chai3d::cMesh` (see description in Chapter 2.2).
- `HapticShape.cs` can be applied to simple shapes such as Unity3D's Cube or Sphere. The surface of those objects is rendered by CHAI3D using a simple *penalty method*. The advantages of this method is that it is computationally

[5] https://www.youtube.com/watch?v=HVYiBpmMfeM

cheap and that more haptic effects are available, including *magnet* and *volumetric* effects (like *viscosity*, *vibrations* and *stickslip*). However, the method is not indicated for thin objects because the tool might pop through them.

4.2.4 weDRAW Serious Games

One of the three final products of the weDRAW Project is the SpaceShape SG. A Non-Playing-Character crashes her Space Rocket on the Moon, and needs the help of the child playing the game to find the parts of the rocket and assemble it to fly back home. The game is organized in three different phases, shown in Figure 4.8, where the child has to: (1) find an hidden cube, by touching its vertexes without being able to see it; (2) explore all the faces of the cube, to find rocket parts hidden inside, and explore all vertexes to find batteries placed there; then, (3) s/he needs to assemble the rocket by pick and placing parts in the correct order.

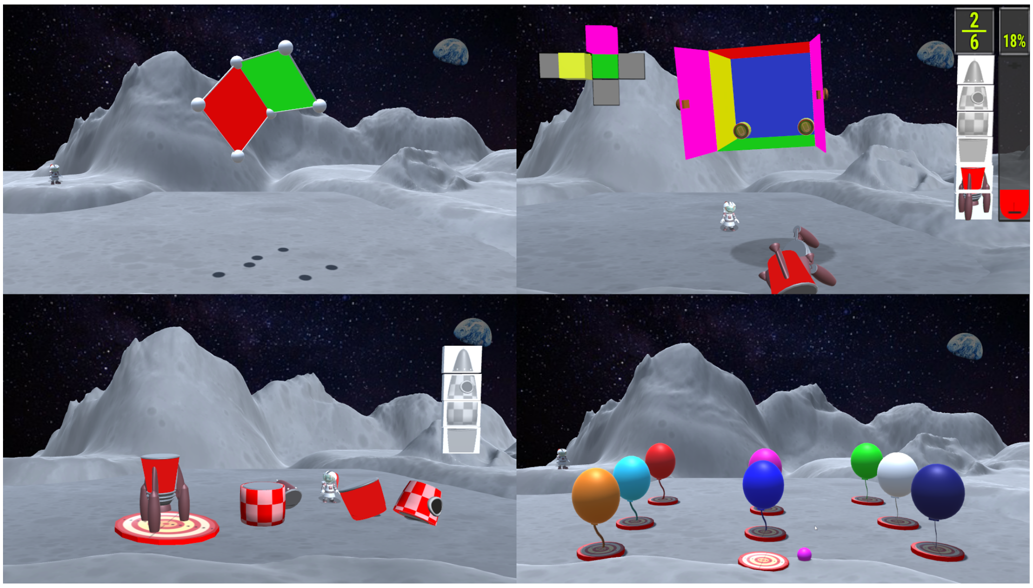

Figure 4.8: The three game phases of the SpaceShape game and the Balloon Game (Bottom Right) from LearnTPM. **Top Left**: Find the vertex of the hidden cube. **Top Right**: Open the faces of the cube to find rocket parts. **Bottom Left**: Assemble the rocket. **Bottom Right**: Pop the balloons.

To evaluate the benefits of the game, a familiarization game, the BalloonGame (shown in Figure 4.8, Bottom Right) has been developed and used as a control condition.

The benefit of the two games was evaluated in the weDRAW Project by the U-VIP and UCL groups. The evaluation protocol for the SpaceShape game involved two

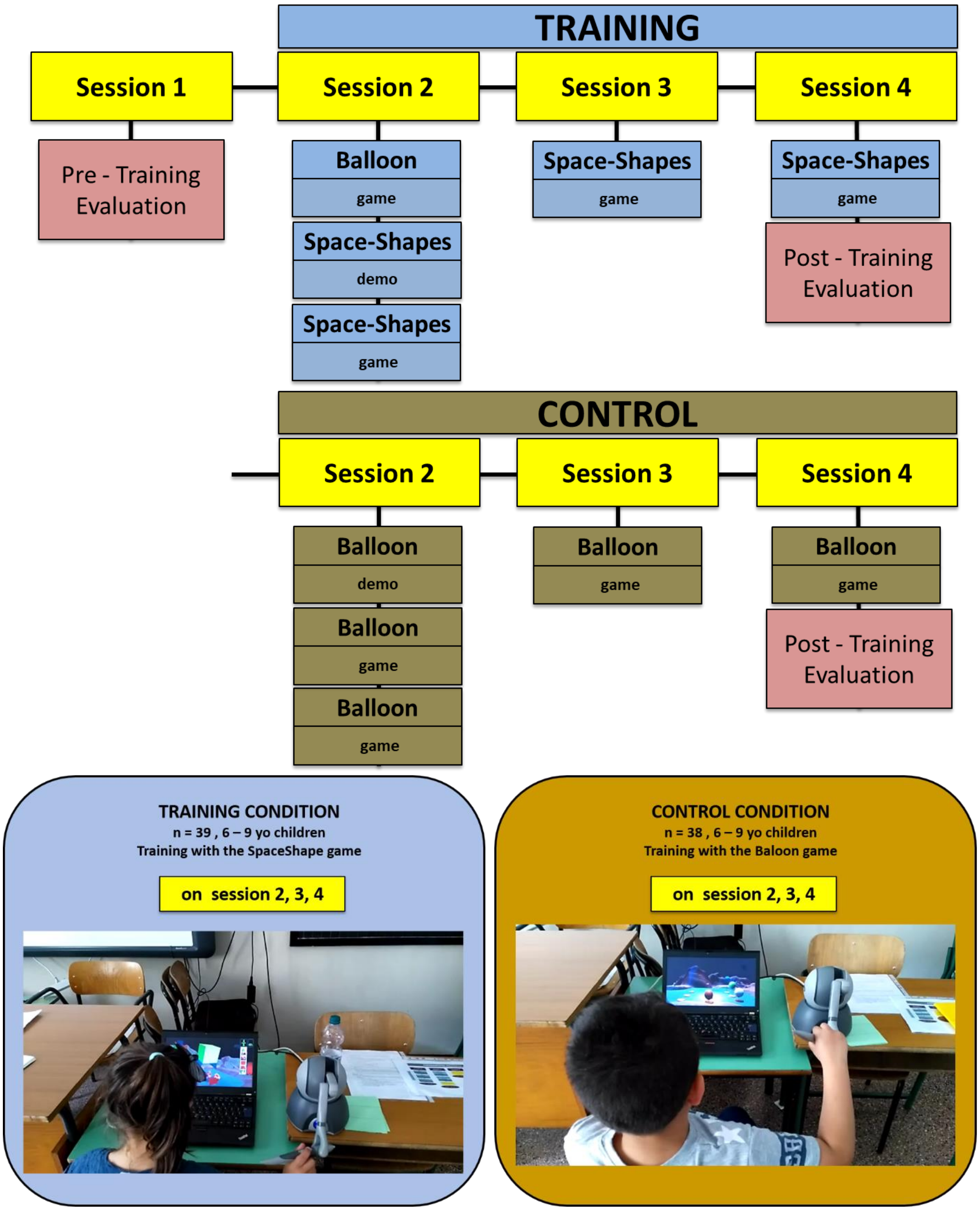

Figure 4.9: Evaluation of the SpaceShape Game. A total of 77 children took part to the experiment. Both the *control* and *training* conditions were organized in four sessions. The first session consisted of a pre-training evaluation on spatial skills. The second session was for familiarization: both group played the BalloonGame once to familiarize with the haptic device; the training group then played the SpaceShape twice, while the control group played again the BalloonGame matching the training group play time. The different games were played also in Session 3 and 4; the latter included a post-training evaluation on the same tasks as the first session.

groups of about forty children between 6 and 9 year old: a *control group* that played only the BalloonGame and the *experimental group* played the SpaceShape game. Both groups participated to one pre-training evaluation session and three training sessions spanning a two-week interval (shown in Figure 4.9). The third training session included a post-training evaluation. The evaluation included questionnaires and tasks that aimed at assessing relevant spatial and representation skills (mental rotation, diagrammatic representation, and proportional reasoning). Preliminary analysis showed that the performances of both groups had improved between the pre- and post-training sessions. While this result is interesting, it is difficult to know whether this improvement is due to the training with the games or to being more familiar with the tasks used to test the performance of the children in the post-training session. Training with the SpaceShape game seems to improve the performance on some tests, such as mental representation and proportional reasoning. While the analysis is still preliminary, these results are promising since this game was designed to train these specific concepts (fraction and 2D/3D spatial representation).

4.2.5 Other Applications

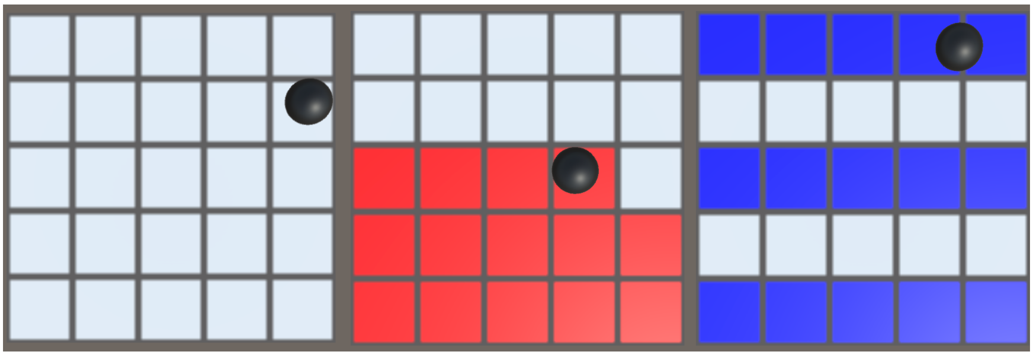

Figure 4.10: Exploration of an horizontal 5×5 grid with the haptic device. The controlled variables were the *feedback* (None, Visual or Auditory) and the *cue* (None, Visual or Auditory). **Left**: Beginning of the trial. **Center**: Visual *feedback*, no *cue*. Tiles change color once explored. **Right**: Horizontal visual *cue*, no *feedback*.

We have used the HPGE plugin in various contexts of application during those three years, including the following cases:

Texture Rendering. Two experiments investigating perception of textures as rendered by CHAI3D algorithm. More details in Chapter 6.

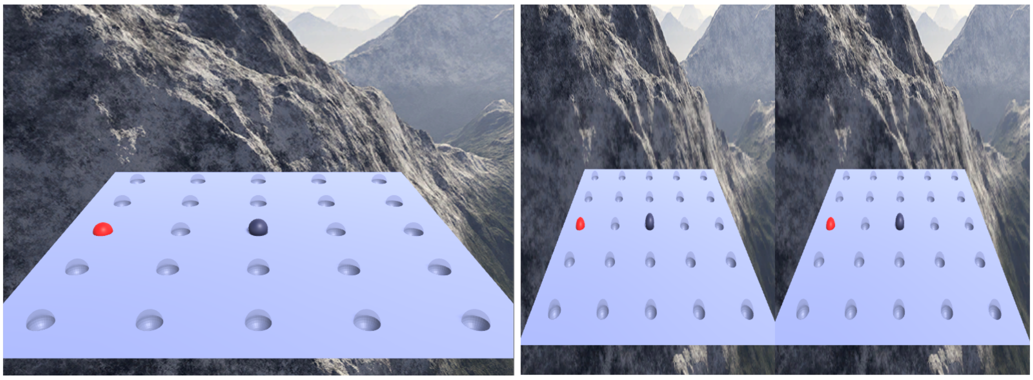

Figure 4.11: **Both**: Target reaching task. The end-effector is represented in the VE with the gray ball at the center of the workspace. The target, in red, could appear on one of the four cardinal points. **Left**: target reaching as seen on the 2D monitor. **Right**: 2D representation of VE as seen by the two eyes with the stereoscopic display. The background has been chosen to ease image fusion by the two eyes.

Haptic Devices Performances. Comparison between texture rendering with two different haptic devices (Force Dimension's Omega and SensAble's Phantom Omni); more details in Chapter 7.

Surface Exploration. Comparison in exploration strategies between normally sighted and visually impaired children of 2D horizontal grids. Figure 4.10 shows the VE used. Two factors were controlled: the feedback (visual and auditory) and exploratory cues (visual and auditory). Either the feedback or the cue was used in each trial. The feedback refer to the information given to the participant about exploration progresses. With a visual feedback (shown in Figure 4.10, Center) explored tiles change color. With an auditory feedback, a sound is played with the exploration of new tiles. The cue refer to the information that is always present on the scene and that might influence the exploration strategy of the participant: cues could either be horizontal, vertical. A total of nine trials for each participant were presented. Progresses were monitored by having three trials (first, fifth and ninth) with no feedback and no cues.

Stereoscopic+Haptic. Evaluation of the role of stereoscopic displays and haptics in a reaching task. The VE used in the experiment is shown in Figure 4.11. A grid placed horizontally was divided into twenty-five tiles. The participant had to move the haptic device from the starting position, placed at the center of the grid, towards the target. The target could be placed in any of tile at the four cardinal points, and was randomized at each trial. After the target was reached, the participant had to return to the center position before the following trial could begin. The reaching time was compared between two visual conditions and three haptic conditions (2×3). The visual representation could either be a 3D view on a 2D monitor or 3D view on a stereoscopic display. Haptic feedback could either be (1) an hole in the horizontal surface, (2) only the horizontal surface or no (3) feedback. The experiment was done in collaboration with University of Genova (Fabio Solari and Manuela Chessa); results are still being analyzed.

5 Experiment on Multisensory Integration

The content of this chapter is based on a draft of a paper titled "The visual, audio, haptic and multimodal perception of ellipses' shape in elementary school children" that Balzarotti and Baud-Bovy wrote and will be submitted for review to a peer reviewed journal.

5.1 Introduction

Our daily life is a multi-modal experience. Depending on the situation, each sense might provide unique, complementary or redundant information; depending on the situation, integrating these information might be useful or not. For example, integrating visual with audio information by reading lip movements is useful when listening to somebody talking in a noisy environment. On the other hand, when reading a text in a noisy environment, we must focus our attention on the visual information and block audio information. Our ability to exploit the wealth of information provided by all senses depend crucially on our ability to select relevant information and process it according to the situation.

Since sensory cues are generally noisy, it makes sense to combine all available information to obtain a more precise estimate. In fact, multiple studies have shown that redundant multimodal sensory cues are integrated optimally according to Bayesian principles; the uncertainty of the final estimate is minimized by giving a weight to each sensory cue that is proportional to its reliability (reviews in Ernst and Bülthoff, 2004; Kersten et al., 2004). For example, Ernst and Banks (2002) demonstrated that the visual and haptic information about the size of hand-held objects is integrated in an optimal manner. Similarly, Alais and Burr (2004) demonstrated the optimal integration between visual and audio information in the localization of the sound source. It is important to note that this type of integration can break down depending on multiple factors that regard the stimuli, the experimental situation and/or the observer (reviews in Spence, 2011; Welch and Warren, 1980).

For this study, it is important to note that optimal integration takes time to develop (Gori et al., 2008; Mjsceo et al., 1999; Nardini et al., 2008; 2010, but see Jovanovic and Drewing, 2014), unlike some other forms of integration such as cross-modal transfers or the McGurk effect that occur already in infancy (e.g., Rosenblum et al., 1997; Streri and Gentaz, 2004). In simple tasks such as judging the size or orientation of an object in the visual and haptic modality, the transition occurs around eight years of age (Gori et al., 2008). Interestingly, the sensory modality used before this transition depends on the task and is not always the most precise one (Gori et al., 2008). In a task involving the fusion of two visual cues, Nardini et al. (2010) found that integration emerged around twelve years while younger children used the best cue. Finally, in a spatial navigation task, Nardini et al. (2008) found integration in adult but not in children with eight years or less. In this latter task, the performance of children in the multimodal condition was worse than the performance in the best unimodal condition.

There is also evidence that decisions can be sped up in multimodal conditions. However, the interpretation of results in these tasks is complicated by the fact that some speed gains are expected in the bimodal condition even in absence of integration by the so-called race models (Miller, 1982; Otto and Mamassian, 2012). Therefore, it is not clear whether the gain observed at a young age in multimodal condition reflects multimodal integration (i.e. the pooling or co-activation models as known in this literature) or not (Schröger and Widmann, 1998). Interestingly, a recent study suggests that some degree of integration or pooling is necessary to explain the speed gains exhibited by four year-olds children but that the pooling is incomplete and, while it increases with age, it does not reach 100 % in adulthood (Nardini et al., 2016). The development of integration observed in speed tasks contrast with the results of the previous tasks that focus on precision, where multisensory integration emerges at a later age but is also fully achieved in adulthood.

In this study, we investigated two questions. The first question is whether children and adults integrate multi-sensory information when perceiving the shape of an object. The second question is whether their sensory preferences and evaluations of their performance reflect the actual performance. To address the first question, we examined whether the precision of the response, as measured by their ability to discriminate between two similar shapes, increases when the information is multimodal with respect to unimodal conditions. In this study, the shape information could be delivered in the proprioceptive, audio and visual modalities and the multimodal condition combined the three sensory modalities. For the second question, we asked the children at the end of the study which condition they preferred and which one they found the easiest.

The task in this study required children and adults to judge whether ellipses with various eccentricities were elongated vertically or horizontally. In all three modalities, the shape was presented sequentially by moving a point along the perimeter of the ellipse. The point corresponded to (i) the hand position in the proprioceptive modality, to (ii) a moving light in the visual modality, or to (iii) a time-varying sound that encoded the position of the moving point in the audio modality. Equalizing the difficulty of the three conditions is important to be able to test the optimal integration hypothesis because this hypothesis predicts that the benefits of integrating in the multimodal condition is greatest when the sensory cues are equally uncertain. When this is not the case, the optimal integration predicts that the performance in the multimodal condition should be similar to the one observed in the unimodal condition with the most reliable information. In the latter case, it is therefore not possible to distinguish experimentally between the optimal integration hypothesis and the winner-take-all hypothesis, which posits that the best sensory modality is used. Finally, it might be noted that the optimal integration hypothesis cannot predict a performance in the multimodal condition that is worse than the best unimodal performance. Such an observation would reflect not only an absence of integration according to the optimal integration model but also that unreliable sensory cues interfere with the most reliable sensory cues.

5.2 Methods

5.2.1 Participants

Three groups of participants took part to the experiment: a group of 38 second-year elementary school children (21 females and 17 males, mean $\pm$ SD: 7.66 $\pm$ 0.35 years), a group of 46 fifth-year elementary school children (27 females and 19 males, 10.46 $\pm$ 0.34 years) and a group of 16 adults (11 females and 5 males, 22.42 $\pm$ 1.48 years). Children were recruited at an Italian elementary school in Genova, Italy. Adults were for the most part University students. The study was approved by the local ethical committee (ASL 3, Regione Liguria). All participants were recruited on a voluntary basis and signed an informed consent form. An adapted consent form was used for the children after obtaining the signed consent of the parents or legal tutors on a different form. The experiment for the children took place during school hours in a separate room of the school where the setup was installed. Two children (7.19 and 7.90 years-old) were tired and did not complete the experiment; their data is excluded from the study. For the adults, the experiment took place at the Istituto Italiano di Tecnologia (IIT) in Genova.

5.2.2 Experimental Setup

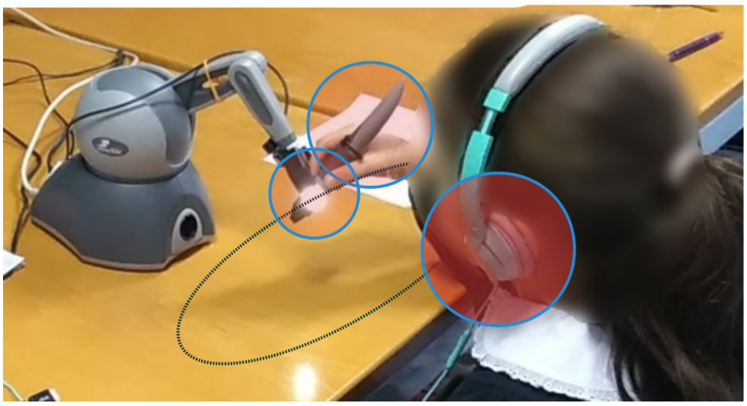

Figure 5.1: Picture of the experimental setup. Haptic feedback was provided by the Phantom Omni haptic device, which was held in hand by the participant. In the visual condition, the participant could watch the device moving by itself. Two LEDs were lit in the visual condition. Audio feedback was provided with headphones (Bose SoundTrue on-ear).

The participant sat in front of a table with a small haptic device (Geomagic Touch; formerly known as the SensAble Phantom Omni device) placed on top (shown in Figure 5.1). The device was aligned with the participant's midline and the height of the seat was adjusted with cushions if needed. Two software-controlled LEDs were placed on each side of the end-effector. During the experiment, the participant wore stereo headphones (Bose SoundTrue on-ear) connected to the computer controlling the haptic device and the LEDs. A custom program controlled the movements of the end-effector, the two LEDs and the audio signal.

5.2.3 Tasks and Stimuli

In all conditions, the stimulus consisted in a point moving along the perimeter of an ellipse in the horizontal plane. The task was to judge whether the ellipse was "horizontally" or "vertically" oriented (two-alternative forced choice, 2AFC). To simplify the understanding of the task with the children, the responses "horizontal" and "vertical" were used to designate the lateral and sagittal directions respectively. The dimensions of the ellipse axes correspond to:

$$\begin{aligned} A_h &= S \cdot E \\ A_v &= S \cdot (1 - E) \end{aligned} \tag{5.1}$$

where the parameter $S = A_h + A_v = 15.2$ cm is a measure of the size of the ellipse. The eccentricity E of the ellipse

$$E = \frac{A_h}{A_h + A_v} = \frac{A_h}{S} \tag{5.2}$$

ranged from 0.05 to 0.95 and was manipulated during the experiment.

The center of the ellipses was approximately 30 cm in front of the body and aligned with the mid-body axis. The point on the perimeter moved with a period of 3 s and the instantaneous velocity followed the 2/3 Power Law in order to avoid perceptual distortions (Viviani et al., 1997). The ellipse perimeter was covered twice during each trial (stimulus presentation duration: 6 s).

The experiment included four conditions:

- In the **visual** condition, the device moved along the ellipse's perimeter by itself with LEDs, placed near the end-effector, lit. The participant simply watched the motion of the device.

- In the **haptic** condition, participants grasped the haptic device with their dominant hand (with the index finger placed on the end-effector) and closed their eyes. In this condition, the haptic device moved the hand along the ellipse's perimeter.

- In the **auditory** condition, participants closed their eyes and listened to a sound produced by the stereo headphones that encoded the position of a virtual point moving along ellipse's perimeter. The sound was a pure tone (440 Hz). The volume encoded the position of the point along the sagittal direction $V(t) = \frac{1-y(t)}{2}$ where $y(t)$ ranged from 0 (near the subject) to 1 (far from the subject) and the L/R balance encoded its lateral position:

$$\begin{aligned} V_R(t) &= x(t) \cdot V(t) \\ V_L(t) &= (1 - x(t)) \cdot V(t) \end{aligned} \tag{5.3}$$

  where $x(t)$ ranged from 0 (left workspace limit) to 1 (right workspace limit).

- In the multimodal condition, the stimulus was delivered in all sensory modalities (visual, haptic and auditory) simultaneously.

Within each condition, the ellipse eccentricity for each trial was selected with the QUEST adaptive method that uses previous responses of the participant to select the value of the next stimulus in an optimal manner (Watson and Pelli, 1983). In this experiment, we used two QUESTs for each condition, which targeted a probability of "horizontal" response of 0.16 and 0.84 respectively, in order to obtain stimulus values near the inflexions point of the psychometric curves (King-Smith and Rose, 1997). For each trial, the condition and the QUEST within the condition were randomly selected until 20 trials for each QUEST were completed. The experiment included therefore `4 * 2 * 20 = 160` trials.

5.2.4 Experimental Procedure

Before the experiment began, the experimenter explained the task and showed to the participant a vertical and a horizontal ellipse drawn on a sheet of paper. Participants who failed to identify correctly the orientation of the ellipses were explained the meaning of horizontal and vertical. The sheet of paper was left on the table during the experiment. The task explanation was followed by a training phase that included two or more 30 s-long familiarization trials with a horizontal ($E = 0.6$) and vertical ($E = 0.4$) ellipses in order to ensure that the participant understood the task and the sonification strategy well. During the training phase, a 15.6'' IPS monitor showing the ellipse was placed flat on the table below the haptic device. The participant grasped the end effector, which could move freely along the perimeter of the ellipse. The sonification scheme was explained and the participant was told to pay attention at how the sound changed with the hand position along the ellipse perimeter. At the end of the familiarization trial, the participant was asked if the ellipse was oriented horizontally or vertically. In the second trial, the orientation of the ellipse was switched. The familiarization trials could be repeated several times if deemed necessary but this was rarely needed. The experimental phase followed the training phase. Before each trial, the participant was told the condition verbally and reminded to grasp the device and close the eye in the haptic condition if necessary. The presentation of the stimuli lasted 6 s, during which the point moved twice around the ellipse. At the end of the trial, the participant reported the perceived ellipse orientation verbally. A trial was repeated if the participant could not respond because of an external interference (e.g., noises) or a distraction, which happened rarely. The experiment lasted approximately 40 minutes, including the familiarization phase.

At the end of the experiment, children were asked which was their favorite and easiest modalities.

5.2.5 Optimal Integration Hypothesis

The optimal integration hypothesis for three sensory cues posits that sensory signals is the weighted average of the three cues

$$e_{123} = w_1 e_1 + w_2 e_2 + w_3 e_3 \tag{5.4}$$

where e_1, e_2 and e_3 represent the visual, audio and haptic cue about the ellipse eccentricity. Bayesian and Maximum Likelihood Estimation (MLE) principles (Ernst and Banks, 2002) can be used to show that value of the variance of the multimodal signals is the value of the weights w_i is proportional to their reliabilities $R_i = \frac{1}{\sigma_i^2}$:

$$w_i = \frac{R_i}{\sum R_i} = \frac{1/\sigma_i^2}{\sum 1/\sigma_i^2} \tag{5.5}$$

where the σ_i^2 are the variances of the unimodal cues, which can be estimated by computing discrimination thresholds in conditions where only one sensory cue is present. It is then easy to show that the variance of the multimodal cue that corresponds to the optimal weights is

$$Var[x_{123}] = w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + w_3^2\sigma_3^2 = \frac{1}{R_1 + R_2 + R_3} = \frac{\sigma_1^2\sigma_2^2\sigma_3^2}{\sigma_2^2\sigma_3^2 + \sigma_1^2\sigma_3^2 + \sigma_2^2\sigma_3^2} \tag{5.6}$$

under the assumption that the noise in the sensory channels is uncorrelated.

5.2.6 Data Analysis

For each subject and condition, we estimated a psychometric function, i.e. the probability of judging the ellipse as “horizontal” as a function of its eccentricity,

$$\Pr(R = \text{Horizontal}|E) = \frac{1}{1 + \exp(-(a + bE))} \tag{5.7}$$

where E is the ellipse eccentricity and a, b are the location and scale parameters of the logistic function. The psychometric functions were fitted to the responses of the subject by maximum likelihood using a Generalized Linear Model (GLM) with logit link function and binomial error in each condition separately.

Then, we computed the Point of Subjective Equality (PSE), i.e. the ellipse eccentricity that corresponding to a probability of 50 % of "horizontal" responses, and the Discrimination Limen (DL), i.e. the difference between the PSE and the ellipse eccentricity that corresponds to a probability of 84 % of "horizontal" responses:

$$\begin{aligned} PSE &= -\frac{a}{b} \\ DL_\pi &= log\left(\frac{\pi}{1-\pi}\right)\frac{1}{b} \end{aligned} \tag{5.8}$$

where $\pi = 0.84$.

In several instances, visual examination of the psychometric functions revealed that participants did not perform the task well. We defined that a performance was *satisfactory* when the PSE was inside the range of eccentricity used in the experiment, $[0.05; 0.95]$, and the discrimination threshold is in the range $[0.0; 0.5]$. All performances that did not satisfy these conditions were deemed to be *unsatisfactory*.

Data was analyzed with the R programming language (R Core Team, 2019) (v3.6.1, rev 76782). The PSEs and DLs were analyzed using robust statistical methods based on 20 % trimmed mean implemented in WRS2 package (Mair and Wilcox, 2019).

5.3 Results

Table 5.1: Distribution of unsatisfactory performances (N: Number of participants)

Group	N	Auditory	Haptic	Visual	Multimodal	Sum
7-8y	38	18	0	0	2	20
10-11y	46	8	0	0	0	8
Adults	16	1	0	0	0	1
Sum	100	27	0	0	2	29

Performance was markedly worse for the 7-8 year-olds children in the auditory condition (see Table 5.1): only about half of the 7-8 year-olds children (20/38) performed the task satisfactorily in this condition. The number of satisfactory auditory performances increased to more than 83 % for 10-11 year-olds (38/46) and almost 94 % for the adults (15/16). A chi-square test confirmed that the number of unsatisfactory performances in the auditory condition changed with the age group ($\chi^2(2) = 13.65$, $p = .001$). In total, 89 % of the unsatisfactory performances (34/38) happened in the auditory condition. For the two children groups, χ^2 tests confirmed an effect of the sensory modality, reflecting the fact that the number of unsatisfactory performances was larger in the auditory condition (7-8y: $\chi^2(3) = 52.51, p < .001$ and 10-11y: $\chi^2(3) = 25.09, p < .001$).

For the adult group, there was only one unsatisfactory performance in the auditory condition. Statistical analyses in the following sections compare the performance across the non-auditory conditions to keep the children who performed unsatisfactorily in the auditory condition in statistical analysis of the DLs and of the PSEs.

5.3.1 Points of Subjective Equality

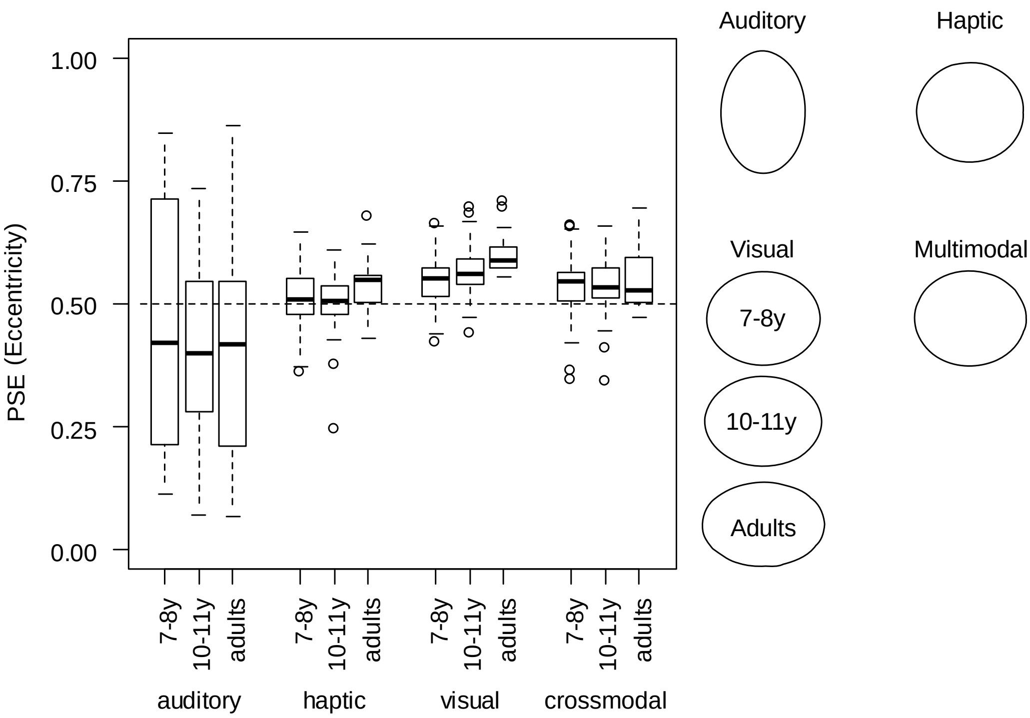

Figure 5.2: **Left**: Distribution of the PSEs. **Right**: Shape of the ellipses that corresponds to median PSEs.

Figure 5.2 Left shows the distribution of the PSE for satisfactory performances. A robust two way mixed analysis of variance (ANOVA) on the trimmed PSE means indicated a main effect of sensory feedback ($F(2, 47.827) = 31.029$, p $<$ .001) but no interaction ($F(4, 40.603) = 2.5569$, p $=$ .0532) or age effect ($F(2, 39.034) = 1.609$, p $=$ .213).

Table 5.2: One-sample test of the PSEs based on the Modified one-step M-estimator (MOM) (H_0: $\mu = 0.5$). P values are computed by bootstrap

Condition	Age Group	Effective Size	MOM	95% CI	P value
Auditory	7-8y	20	0.439	0.244 0.563	0.285
	10-11y	38	0.425	0.358 0.494	0.036
	Adults	15	0.408	0.220 0.556	0.318
Haptic	7-8y	38	0.521	0.493 0.543	0.176
	10-11y	46	0.51	0.494 0.528	0.206
	Adults	16	0.537	0.507 0.564	0.016
Visual	7-8y	38	0.545	0.526 0.567	0.000
	10-11y	46	0.56	0.548 0.580	0.000
	Adults	16	0.583	0.573 0.625	0.000
Multimodal	7-8y	36	0.549	0.517 0.564	0.009
	10-11y	46	0.535	0.523 0.551	0.000
	Adults	16	0.54	0.505 0.582	0.024

Table 5.2 reports the results of robust one-sample tests based on the MOM to test whether PSE differed from circular shapes (H_0: eccentricity $= 0.5$). In general, the MOMs in the auditory condition was below 0.5, indicating a tendency to perceive vertically oriented ellipses as circular. The test of the PSEs in this condition was statistically significant only for the 10-11y group, which might reflect the larger sample size and minor variability of this group. In contrast, the MOM in the other conditions tended to be above 0.5, with the difference between the largest in the visual conditions. To find out whether age affected the PSE in each condition, we performed separate one way ANOVA on trimmed means. This test is significative for the visual condition ($F(2, 28.161) = 5.585$, p $= .009$, $\xi = 0.540$).

5.3.2 Discrimination Thresholds

Figure 5.3 Left shows the distribution of the DLs for each age group and sensory condition after excluding unsatisfactory performances. Despite the fact that removing the unsatisfactory performances introduces a bias that favor the auditory condition, the left panel shows clearly that DL values are worse in the auditory conditions than in the other sensory conditions for all age groups.

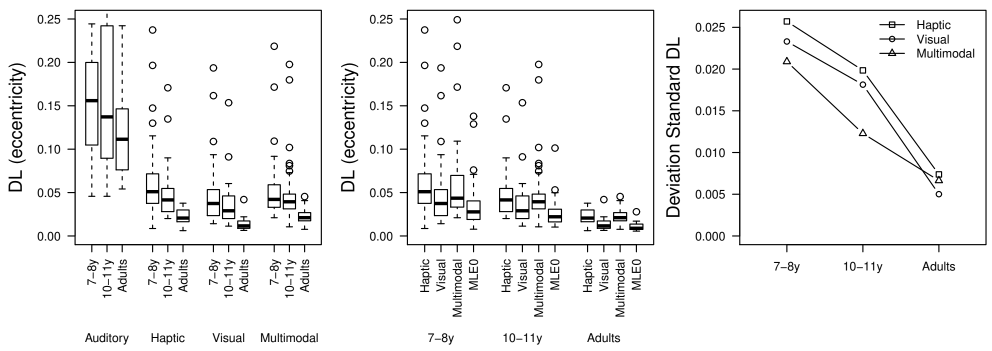

Figure 5.3: Discrimination Threshold in different sensory and age groups. **Left**: Effect of age on the DL for each sensory modality. **Center**: Comparison between the DL in two unimodal conditions, multimodal condition and the Maximum Likelihood Estimation (MLE0). The MLE0 is always lower than both any unimodal condition and the multimodal condition. **Right**: The standard deviation of the DL decreases with age.

Figure 5.3 Center shows values of the threshold in all feedback conditions except the auditory condition and the value of the threshold predicted by the optimal integration hypothesis (MLE0) in the multimodal condition. Only the data from four participants who did not perform well in the multimodal and/or haptic conditions are excluded.

Table 5.3: Post-hoc comparison between trimmed means. P values are adjusted for family-wise error rate

Condition	Contrast	Statistic	P value
haptic	7-8y/10-11y	0.011	0.057
	7-8y/adults	0.03	$<$ 0.001
	10-11y/adults	0.02	$<$ 0.001
visual	7-8y/10-11y	0.006	0.164
	7-8y/adults	0.025	$<$ 0.001
	10-11y/adults	0.019	$<$ 0.001
multimodal	7-8y/10-11y	0.005	0.301
	7-8y/adults	0.023	$<$ 0.001
	10-11y/adults	0.018	$<$ 0.001

First, we carried out a two-way mixed ANOVA on the trimmed means of the DL with the *age group* as between-subjects factor (7-8y, 10-11y and adults) and the *sensory feedback* as within-subjects factor (visual, haptic and multimodal). We found no significant interaction between age and feedback ($F(4, 41.459 = 0.639$, p $= .638$), but a significant effect of both age ($F(2, 38.670) = 37.753$, p $< .001$) and sensory modality ($F(2, 52.168) = 22.066$, p $< .001$) on the DL. One-way ANOVAs on the trimmed means of the DL confirmed the effect of age within each feedback condition (haptic: $F(2, 31.879) = 19.214$, p $< .001, \xi = 0.711$; visual: $F(2, 36.821) = 29.277$, p $< .001, \xi = 0.651$; multimodal: $F(2, 35.702) = 22.887$, p $< .001, \xi = 0.671$). The values of ξ above 0.5 correspond to large effect sizes (Mair and Wilcox, 2019). Pairwise comparisons in Table 5.3 show that adult performed better than the two children groups. The differences between 7-8 and 10-11 year-olds were not statistically significant however.

Table 5.4: Robust post-hoc comparisons for paired data. P values are adjusted for family-wise error rate with Hochberg method

Age group	Contrast	Statistic	P value
7-8y	visual vs. haptic	-0.015	0.013
	visual vs. multimodal	-0.009	0.048
	haptic vs. multimodal	0.006	0.245
10-11y	visual vs. haptic	-0.010	0.000
	visual vs. multimodal	-0.010	0.001
	haptic vs. multimodal	0.001	0.750
adults	visual vs. haptic	-0.009	0.005
	visual vs. multimodal	-0.009	0.002
	haptic vs. multimodal	0.000	0.937

Robust one-way repeated-measure ANOVAs confirmed that the audio, visual and multimodal discrimination thresholds varied across sensory modalities within each age group (7-8y: $F(1.961, 41.186) = 7.824$, p $= .001$; 10-11y: $F(1.971, 53.220) = 6.863$, p $= .002$; adults: $F(2, 18) = 8.330$, p $= .003$). Follow-up pairwise comparison tests show that visual thresholds are lower than haptic and multimodal thresholds for all groups (see Table 5.4). In contrast, the differences between the thresholds in the haptic and multimodal conditions are not statistically significant.

5.3.3 Multisensory Integration

According to the optimal integration hypothesis, the discrimination thresholds in the multimodal condition should be lower than the thresholds in any other sensory modality alone, which was not the case, as observed previously. The fact that the thresholds predicted by optimal integration hypothesis were much smaller than the thresholds observed in the multimodal condition (see Figures 5.3) was confirmed statistically (Yuend's test on trimmed means for dependent samples: $p < .001$ for all age groups).

An alternative hypothesis to the Optimal Integration Hypothesis is the winner-take-all hypothesis, which posits that the performances in the multimodal condition should be the same as the best unimodal sensory modality. The observation that thresholds in the multimodal condition is higher than in the visual condition goes against this hypothesis (see Table 5.4).

Given the observation that the thresholds in the multimodal and haptic conditions were similar for all age groups, a third hypothesis is that participants based their responses in the multimodal condition on the haptic cue rather than on the visual one. A possible way to test this hypothesis is to examine whether individual multimodal discrimination thresholds are better correlated with haptic multimodal than visual discrimination thresholds.

Table 5.5: Winterized coefficients of correlation between individual discrimination thresholds for each age group. P values are computed by bootstrap ($*$: $p < 0.05$, $**$: $p < 0.01$; $***$: $p < 0.001$)

	7-8y	10-11y	Adults
visual vs. haptic	0.54**	0.53***	0.37
multimodal vs. haptic	0.65***	0.26	0.34
multimodal vs. visual	0.67***	0.46**	-0.03
multimodal vs. MLE	0.75***	0.40**	0.11

Table 5.5 reports the winsorized coefficients of correlation between the individual thresholds in the multimodal condition and the other feedback conditions for each age group. The table also includes the correlation of coefficients between the multimodal condition and the value predicted by the MLE hypothesis. The coefficients of correlation are in general larger for the children group and weakest for the adults. While it might be tempting to interpret this, it should be noted that the values of the discrimination threshold varied much less in the adult group (as shown in Figure 5.3 Right). As a result, it is possible that threshold measurement errors together with the observation that threshold are less variables in the adult populations could suffice to explain the lower correlation observed in the adult population. For the children groups, the thresholds in the multimodal condition are less correlated with the haptic condition than the visual condition, which does not

support the hypothesis that the haptic cue dominated the visual cue in the cross modal condition.

5.3.4 Questionnaires

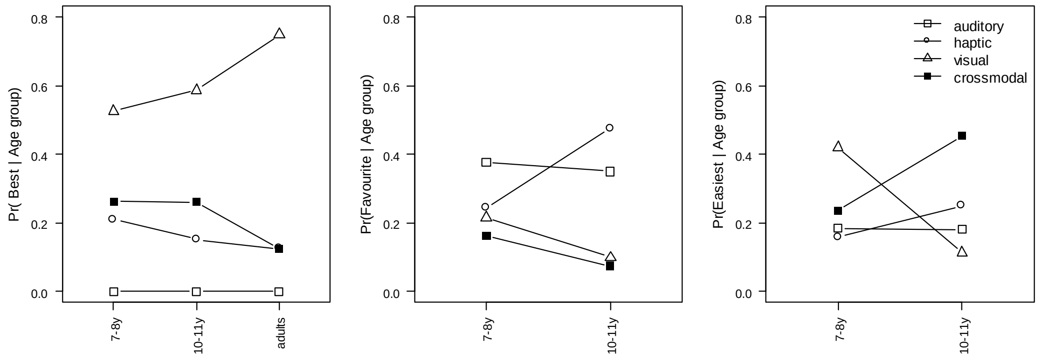

Figure 5.4: **Left Panel**: Frequency of participants for each age groups whose lowest Discrimination Limen was in in a specific modality. **Center Panel**: Frequency of participants which reported a feedback condition as being their *favorite*. **Right Panel**: Frequency of participants which reported a feedback condition as being the *easiest*.

Children reported their favorite and easiest sensory modalities at the end of the experiment. To compare these responses with an objective measure of performance, we defined the best sensory modality as the sensory modality with the lowest discrimination threshold for each participant.

The best sensory modality was the visual one for 64 % of the participants, followed by multimodal in 21 % of cases and haptic for the remaining 15 %.

First, we examined the two-way contingency tables for each possible response variable to find out whether the distribution of the best (B), favorite (F) or easiest (E) responses changed across age groups (A). We fitted the observed frequencies with a Poisson regression model including an interaction between the response and age group variables and calculated type-II likelihood ratio tests. In this model, a statistically significant interaction indicates that conditional distributions of the four possible responses change across age groups while a statistically significant main effect associated with the response variable indicates that the distribution of participants across conditions is not uniform.

1. As noted previously, Figure 5.4 Left shows that the largest group of participants performed best in the visual modality at all ages and that this proportion tended to increase with age. The main effect of associated with the

response variable was highly statistically significant (LRT: $\chi^2(3) = 97.673, \mathrm{p} < .001$). However, the interaction was not statistically significant (LRT: $\chi^2(6) = 1.777, \mathrm{p} = .939$).

2. Figure 5.4 Center shows interestingly that children preferred the two modalities (auditory and haptic) that were objectively the most difficult ones (main effect: $\chi^2(3) = 16.938, \mathrm{p} < .001$). For the 7-8 year-olds children group, the preferred condition was the auditory sensory modality while the preferred sensory modality for the 10-11 year-olds children group was the haptic one. The preference for the haptic modality increased with age to the detriment of the other three modality. The LRT testing whether the distribution of the four responses changed across age groups was not statistically significant (LRT: $\chi^2(3) = 5.913, \mathrm{p} = .116$). However, a LRT between the independence model and a more parsimonious model assuming an age group effect only on the distribution of haptic modality was statistically significant ($\chi^2(1) = 4.538, \mathrm{p} = .033$).

3. Finally, Figure 5.4 Right shows that the sensory modality that children found easiest changed between the two age groups: while youngest children found the visual modality easier, the older children found the multimodal condition easier. The Poisson model confirmed that age had an effect on the distribution of the responses (interaction: $\chi^2(3) = 11.458, \mathrm{p} = .009$). The main effect of the response variable was not statistically significant ($\chi^2(3) = 5.3942, \mathrm{p} = .145$).

Second, we examined whether there was an association between the best sensory modality and the two subjective responses:

1. Comparing the best and easiest response shows that the two objectively worse experimental conditions (haptic and audio) are also judged to be more difficult than the other conditions. However, there is an obvious difference between the best and easiest responses for the older children, who perform best in the visual condition but find the multimodal condition easier than all other conditions. A sequential Poisson regression model analysis indicated that age group and easiest responses are jointly independent from the best responses (joint independence [AE, B] vs. complete independence [A, E, B] models: $\chi^2(3) = 11.302, \mathrm{p} = .01$). The data did not support conditional independence ([AE, B] vs [AE, AB]: $\chi^2(9) = 3.795, \mathrm{p} = .924$) or homogeneous association [AE, AB, EB] models.

2. We did not find an association between the best and favorite responses. In fact, the best model in this case what the complete independence model [A, B, F]. The data do not support more complex models including two-way interactions.

3. Finally, we also did not find an association between favorite and easiest responses. In this case, the data supported a model where the age group and easiest response were jointly independent from the favorite response (joint independence [AE, F] vs. complete independence [A,E,F] models: $\chi^2(3) = 11.302, \mathrm{p} = .01$). As before, the data did not support conditional independence ([AE, F] vs [AE, AF]: $\chi^2(3) = 6.015, \mathrm{p} = .111$) or homogeneous association models.

To summarize this last part, these analyses suggest the objective measure of performance – that is, the best sensory modality – and the sensory modalities that was preferred or found to be the easiest are independent. There analyses also confirm that the sensory modality that was judged to be the easiest is the only variable that is affected by age in a consistent manner.

5.4 Discussion

Ellipse shape discrimination is best in the visual condition. The first objective of the experiment was to compare the perception of the shape of an ellipse presented in various sensory modalities. To that end, we measured the PSE, i.e. the perceptual biases of the participants and the discrimination thresholds, or, in other words, their capacity to discriminate the eccentricity of ellipses in three unimodal conditions where only audio, visual or haptic feedback was present. For all age groups, the results show that the auditory condition was the most difficult and the visual condition was the easiest one. This result was in part expected because shape discrimination is a spatial task and the visual modality is the preferred modality for spatial tasks (Bermant and Welch, 1976; Jack and Thurlow, 1973; Kitagawa and Ichihara, 2002; Repp and Penel, 2002; Vroomen et al., 2001).

Obviously, this result is limited to the specific stimuli, sensory modalities and presentation methods used in this study, since the task difficulty might differ depending on how the stimulus is presented in each condition. That said, it should be noted that we tried to equalize the difficulty of the task in the three modalities.

First, the task required that the participant reconstructed the shape of the ellipse from trajectory of a moving point in the three modalities. As a matter of facts, the visual stimulus – the LEDs on the end-effector – was a moving point in the visual modality like in the audio or haptic modality; an easier alternative would have been a static two-dimensional representation of the ellipse which would have made the task easier in the visual modality.

Second, the haptic device guided the hand of the participant in the haptic modality rather than letting the participant explore the shape freely. Previous research suggests that this passive modality of exploration yield better haptic performances than free exploration (Gori et al., 2012).

Third, the sonification strategy was chosen after informally testing different sonification schemes, where we compared the encoding used in this study with, for example, encoding that combined the volume and/or the pitch of a tone. Unfortunately, we did not find previous studies that would provide guidance about how to best encode the position in the audio modality. It is possible that the bad performance in auditory condition might reflect a suboptimal audio encoding of the position of the moving point instead of worst capabilities in the audio modality.

Perceptual biases. The analysis of the PSEs revealed some biases, which were stronger in the audio and visual conditions. There was considerable inter-individual variability in the audio modality, reflecting perhaps the difficulty of the task; the central tendency however corresponds to a vertically oriented ellipses. This observation might suggest that the volume difference between the left and right influenced the perceived position of the moving point along the horizontal axis. Given the sonification scheme used in this study, it should be noted that the volume in each ear was proportional to the product of the positions along the horizontal and vertical direction.

For the visual modality, the PSEs corresponded to horizontally-oriented ellipses suggesting that all groups might have (over) compensated for perspective. This over compensation might be related to a misperception of the slant of the plane in which the end-effector moved. Interestingly, this compensation was stronger for adults, which might reflect the fact that these mechanisms are still developing and not yet mature at the in 11 year-olds, the age of the oldest children involved in this study (Burr and Gori, 2012; Diaconescu et al., 2013; Dionne-Dostie et al., 2015; Ernst, 2008).

Children preferences and subjective judgments of difficulty. We examined how age affected the preferred sensory modality and subjective judgment of difficulty and compared them to the condition which was performed best.

First, the responses reflect individual differences between the children: all four conditions were mentioned by some children as the favorite or as the easiest one. At the same time, the distribution of the preferences and difficult judgments was not uniform, indicating that some sensory modalities were preferred or judged as easier by more children than other ones.

Second, we did not find evidence of association between the favorite and easiest modality, or between these two variable and the best sensory modality. This indicates that children did not based the preference for a sensory modality on what appeared to be the easiest one and vice-versa. These analyses also indicated that subjective and objective judgments of difficulty are not related in any obvious manner.

A closer look at the data helps to understand better the absence of relation between subjective and objective judgment of difficulty. First, it is interesting to note that a group of children found the audio modality to be the easiest one even though, for all children, the task was performed less well in the auditory condition than in other conditions. Second, when considering the effect of age on the difficulty judgment, the proportion of children who found the visual modality the easiest went from the largest one for 7-8 year-olds to the smallest one for 10-11 year-olds. This clearly differs from the actual performance, showing that the proportion of children who did best in the visual modality increased from the 7-8 year-olds to the 10-11 year-olds group. A caveat that might explain this discrepancy in part is that the fact that the visual modality was judged to be the easiest by the smallest group of 10-11 year-olds children should not be interpreted as indicating that the children who responded differently found it to be the most difficult one. In fact, the multimodal condition, which was judged as the easiest by the largest group of 10-11 year-olds children includes visual feedback and it is possible that the children who found another condition easier also found the visual condition to be easy but not as much.

With respect to the favorite sensory modality, we already noted that most children did not prefer the visual sensor modality, which was the best one for most children. Arguably, the two sensory modalities that were preferred by the children (haptic and audio) were the least familiar ones. One possible explanation is the “novelty”, which has the effect to increase the preference for novel stimuli (Skidgell et al., 1976); this effect also increases with age (Mendel, 1965; Witryol and Valenti, 1980), which might explain the increase in preference for the haptic modality in the older group. As a matter of facts, this study was the first time that children had been exposed to a haptic device. The sonification strategy of the movement used in the auditory condition was also new to them. In contrast, using the movement of a point visually, even fixed on the haptic device, to judge the shape of the ellipse might be more familiar give that the visual modality is typically used to present geometrical information and moving objects.

Multimodal sensory integration. The final objective was to find out whether participants integrated the unimodal cues optimally in the multimodal condition and whether optimal integration changed during development. The results of this study do not support the hypothesis that unimodal cues are optimally integrated in the multimodal condition, even for the adults. As a matter of facts, the results in the multimodal condition were worse than the best unimodal condition, in contraction with the optimal integration hypothesis. This result is in contrast with the results of previous studies involving in size and orientation discrimination tasks that have shown that integration processes improved during development (Gori et al., 2008).

A theoretical explanation for the absence of integration might be that the different sensory cues might not appear to be congruent. There is a large set of studies that show that optimal integration depends on a variety of low-level factors such as spatial and temporal congruence and high-level factors such as semantic congruence (Chen and Spence, 2017; Laurienti et al., 2004; Spence, 2007; Warren et al., 1981; Welch and Warren, 1980). However, it should be noted that the LEDs were placed near the end effector to help the subject focus the attention on the device and to provide a visual stimulus co-localized with the haptic stimulus, since it is believed that co-localization might improve multi-sensory integration (Neil et al., 2006).

The results are also incompatible with the hypothesis that participants used the best sensory cue since the discrimination thresholds were on average larger in the multimodal condition than in the visual conditions.

It should be noted that previous studies have shown that people might not always used the best available cue. For example, Gori et al. (2008) found that young children used the haptic modality to judge the size of an object even though the discrimination threshold were lower in the visual modality. These authors argued that the haptic modality provided more accurate information in this task and that it was used to calibrate the visual modality. They also found that slightly older children and adults used both sensory cues in an optimal manner, which was not the case in our study. While the thresholds in the multimodal condition and the thresholds predicted by the optimal integration hypothesis should in principle agree and not simply correlate, correlation is often used as evidence for integration. Similarly, one would expect that the thresholds in the multimodal conditions correlate with the thresholds in the unimodal condition that corresponds to the sensory cue used in the multimodal condition.

This pattern of results does not provide evidence supporting the hypothesis that the participants in this study used the haptic modality as the main sensory cue in the multimodal condition. Indeed, the correlation between the threshold in the haptic and multimodal conditions was less strong that the correlation with the visual or MLE thresholds.

What happens with age? The most notable age-related effect in this study is the decrease of the discrimination thresholds in all conditions with age. The largest difference was found between the 10-11 year-olds group and the adults. While the 10-11 year-olds were on average better than the 7-8 year-olds children, the difference was not statistically significant. Results show that the performance in this task continue to improve after the end of elementary school. Further studies should examine at which age the performance reach an adult level.

The improvement in the multimodal condition does not appear to reflect a development of multimodal integration processes since adults, no more than children, appear to integrate sensory cues in this condition.

An open question is whether this improvement reflects a development of perceptual or cognitive processes. One possibility is that adult perform better because they perceive the position of the moving point more precisely. Another possibility is that they are better able to represent the ellipse shape and/or to compare vertical and horizontal extent of the represented ellipse.

6 Experiments on Perception of Virtual Textures

One long-standing question in perception regards the relationship between physical attributes in the world, such as the weight or the size of an object, and the corresponding percepts, and how the percepts are affected by changes in physical attributes.

This fundamental question is at the origin of the Psychophysics, which marked the historical turn of Psychology from introspective methods to scientific ones (Fechner et al., 1966). In many cases, this question does not have a simple answer because the percepts, such as the perceived weight of an object or its heaviness, are influenced by multiple physical dimensions. For example, it has been known for long that the size of an object can influence its heaviness (the size-weight illusion; Charpentier, 1891).

In this section, we report the results of two studies on the haptic perception of VEs. In the weDRAW Project, the question of the virtual textures emerged when discussing how to render the surface of the virtual cube in the SpaceShape game and whether it might be useful to differentiate them by associating each surface with a different texture and/or color. While the current version of the game does use different texture, the CHAI3D algorithms in the HPGE library can simulate virtual textures.

Virtual textures are artificial textures that are created by force-rendering algorithms that simulate the contact with objects in virtual environments. The physical interaction and the subjective experience when touching and exploring these virtual objects depend on the values of many parameters which need to be set in the VE. For virtual textures, we identified four parameters that could potentially affect the perception: (1) the Stiffness, (2) the coefficient of Dynamic Friction, (3) the Spatial Frequency of the texture and (4) Level of Contrast of texture elements.

The goal of both studies was to find out how these parameters influenced the perception of virtual textures. This might be useful to design virtual textures that are perceptually salient and/or contrast with each other, for example. In the first study (Section 6.1), we manipulated only one parameter at the time to find how much each parameter influenced the texture perception. In the second study (Section 6.2), we manipulated several parameters together. The results of the first experiment have been published in 2018 on Haptics: Science, Technology, and Applications (Balzarotti and Baud-Bovy).

6.1 First Study

Psychological dimensions such as perceived roughness might differ from the physical attributes of the touched surface such as its friction coefficient. Previous research has aimed at identifying how many psychological dimensions are involved in the perception of textures and at relating them to the physical attributes of the textures used. Only a brief overview of this research is given in this introduction. Most research on haptic perception has focused on the perception of the texture of real materials. Holliins et al. (1993) conducted a seminal work on the perception of real textures. In their study, they analyzed the perceived distance between 17 real materials. Using MDS techniques, the perceived distance between the stimuli could be represented in three-dimensional space, where the two main perceptual dimensions corresponded to roughness-smoothness and hardness-softness (see also Hollins et al., 2000; see Okamoto et al., 2012 for a recent review). Perceiving textures and/or material properties rendered by a force-feedback haptic device present several notable differences. First, the surface is touched with a hand-held probe rather than with the finger. In such a condition, the information about the surface is conveyed by vibrations transmitted by the probe to the hand. The cues related to the deformation of the skin and imposed by the surface are missing since the finger is not in contact with the surface. Interestingly, Yoshioka et al. found that using a probe did not prevent perceiving a variety of textures. The perceptual dimensions obtained with the probe were similar (though not identical) to those obtained with the bare finger, even though mechanisms that are involved are certainly different. Moreover, changing the force range and/or movement speed did alter markedly roughness judgment (see also Klatzky et al., 2003). Another issue for virtual textures is that the contact with the object is simulated. Unfortunately, grounded force-feedback devices and, in particular, low-cost devices are unable to render high frequencies (Campion and Hayward, 2005; Culbertson et al., 2013). From a texture perception point of view, this should strongly limit the realism of the textures that can be achieved with this technology. Otaduy and Lin (2005) implemented a texture-rendering algorithm that replicated the motion of the probe when interacting with low-frequency gratings. Tests conducted with algorithms yielded results comparable to those of Klatzky et al. (2003). Kuchenbecker and colleagues have developed models and algorithms to render real-world textures by transmitting texture-specific high-frequency vibrations to the user. Early work using a high-fidelity force feedback device validated the approach by showing that a user could use these vibration patterns to discriminate between several textures (Kuchenbecker et al., 2005). Later, this approach was tested with a low-cost haptic device equipped with a small linear motor to render high frequencies (Culbertson et al., 2014; McMahan et al., 2010). This approach yields promising results in terms of the ability of such a system to render textures realistically. Unfortunately, this technology is not yet part of commercially available haptic devices and most haptic software development kits do not support it yet.

6.1.1 Objective

The general objective of this study is to find how to best render textures with the parameters of existing haptic software development kits and low-cost haptic devices. To that end, we conducted a study to assess the impact of CHAI3D force- and texture-rendering parameters on texture perception. CHAI3D is an open source (Free Software released under Revised BSD 3-clause), freely available set of C++ libraries for computer haptics, visualization and interactive real-time simulation (Conti et al., 2005). CHAI3D originated in 2003 from research at Stanford University and is widely used library to develop haptic applications. In CHAI3D, haptic objects have a primary form (or geometry) which is usually specified by a mesh. In addition, the haptic objects can be endowed with a rich set of material properties such as stiffness, static and dynamic friction, two magnetic effect parameters (magnetic distance and magnetic strength) and textures, which are specified by providing a texture image, and a texture level parameter. Reviewing the texture-rendering algorithms is out of scope of this paper (but see Lin and Otaduy, 2008). For our purpose, a high-level description might suffice. CHAI3D uses the "finger-proxy" algorithm developed by Ruspini et al. (1997), which is similar to the "god-object" algorithm proposed by Zilles and Salisbury (1994), to render the contact force with a virtual object. In this algorithm, a virtual spring connects a proxy position, which is constrained to remain on the surface of the virtual object, to the actual position of the end-effector inside the object. The stiffness parameter controls the stiffness of the spring, which prevents the end-effector from penetrating more in the object. Finally, CHAI3D also implements the "stiction" algorithm, an extension of the god-point algorithm that can account for frictional effects (Salisbury et al., 1995). In this algorithm, the static and dynamic friction coefficients have an effect on the tangential force necessary to move the proxy and, thus, also on the interaction force. The main parameters for rendering a haptic texture are the spatial characteristics of the grayscale image (color images are transformed into a grayscale) and its level. In CHAI3D, the addition of a texture to an object will locally modify the interaction force both in the normal and tangential direction according to the gradient of the image. The extent of the texture-related changes depends on the texture level. In a sense, the texture level is the haptic equivalent of the contrast of the image in the visual domain. A variety of techniques might be used to describe spatial characteristics of an image that might be relevant for texture, such as the angular second moment of the gray-level co-occurrence matrix (GLCM) or the spectral centroid (Culbertson et al., 2014). The goal of this preliminary study is to build scales for the main parameters involved in the rendering of virtual haptic textures and to compare these scales to find out which parameter has the largest impact on texture perception. To that end, we systematically manipulated the texture parameters available in CHAI3D, trying to cover the range of values that a low-cost haptic device could render. More precisely, we manipulated the stiffness, the dynamic friction, the texture level and the texture image. We used the pairwise-comparison method to present a pair of textures to the observer who had to judge the similarity/dissimilarity of the pair on a scale ranging from "equal" to "completely different". Then we used MDS techniques to build the psychological scales that reflect the perceived similarity of the stimuli as done in many previous studies on textures perception (Holliins et al., 1993; review in Yoshioka et al., 2007). Because the number of pairs increases quickly with the number of stimuli, we decided to focus on subscales where only one parameter was varied at a time with respect to a reference stimulus and analyze the perceptual dimension for each parameter independently. Crucially, however, all pairs were presented together in a random order so that the subscales would have the same unit and could be compared. The main results are that some parameters have a major impact on the texture, which is reflected by a larger extension of the corresponding scale. We also found that two perceptual dimension might be necessary to account for the similarity judgments even when one rendering parameter was manipulated.

6.1.2 Methods

Eighteen volunteers (10 Females, 8 Males, Age range: 24-32 year-old, mean±SD: 27.7±2.7 years) took part to the experiment. All participants signed an informed consent form, which was approved by the local ethical committee (ASL 3, Regione Liguria). Each participant was asked to sit in front of a 15.6" IPS monitor. The flat screen lied on an inclined surface, with a 30-degree slope, to facilitate the vision of the screen. A Phantom Omni device (Geomagic) was placed above the screen on the same surface (as shown in Figure 6.2, Left), so that its workspace was exactly above the monitor. The participant was asked to wear headphones with active noise-cancelling enabled. To further prevent participants from hearing the Phantom Omni device motors' noise, the headphones played a white noise. The haptic device was controlled by a custom haptic plugin for Unity3D ® game engine. This haptic device plugin used CHAI3D to render textures and objects haptically. Game objects with textures are created in Unity and then transparently mapped to objects with the same geometry and position in the CHAI3D haptic world. The feedback force was computed in a high frequency thread (>1 kHz) running in the haptic plugin. The position of the objects was fixed. The software ran on a desktop computer (processor Intel Pentium G3450, 4 Gb RAM) with Windows ® 7 Enterprise operating system. The task was to judge the similarity/dissimilarity of textures presented pairwise. The monitor displayed two gray rectangular surfaces (16 cm by 5.33 cm, see Figure 6.2, Right), The textures were not rendered visually. Haptically, the gray surfaces corresponded to two slightly raised parallelepipeds placed above a background plane. At the beginning of each trial, a red dot started moving horizontally on one of the two surfaces, at a frequency of 1 Hz. The user had to follow the red dot while touching the virtual surface with the end-effector, the position of which was indicated by a black dot. The user could explore the first surface as long as desired but, after a 5s-long contact with the surface, the red dot switched to the other surface to instruct the participant to start exploring the second texture. After moving and exploring the second probe for at least 5s, the two rectangles disappeared as soon as the participant lifted the stylus. In general, the participants followed the cues to switch surface and end the trial without delay. Then, the participant had to respond to the question "How similar are the two stimuli?" by moving a visually displayed slider with the haptic device. The slider's extremities were labeled "Equal" and "Completely Different". The initial position of the slider was randomized. After 45 trials there was a small pause to let the participant to relax. Two Omni haptic devices were used in the experiment and switched during the pause to avoid excessive warming of the motors. The device order was counterbalanced across participants.

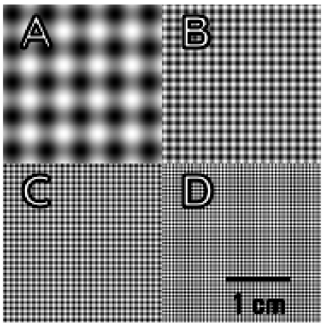

Figure 6.1: 2D-sine Textures Spatial Frequency. Values are shown in Table 6.1.

Table 6.1: Texture Characteristics

Texture	Spatial Freq. (cycle/mm)	GLCM (2nd Mom.)
S	—	1.00
A	0.19	0.16
B	0.75	0.12
C	1.13	0.14
D	1.68	0.12

The textures were grayscale 200×600 texels images with different frequency content. Four textures (A, B, C and D, Figure 6.2, Left) were generated by a combination of sinusoids along the x and y axes, with the center at half of the grayscale (mid-gray value of 128), and their gray values covered the whole scale (min value = 0, max value = 255). The fifth texture (S) is completely gray and corresponds to a smooth surface (see Figure 6.2, Right).

Table 6.2: Textures stimuli parameters

Name	(#) Values	Reference value	# pairs with different / same stimuli
Texture Level	(4) 0.0, 0.3, 0.5, 0.7	0.5	6 / 4
Stiffness	(4) 0.2, 0.4, 0.6, 0.8	0.6	6 / 4
Texture Image	(5) A, B, C, D, S	C	10 / 5
Dynamic Friction	(4) 0.0, 0.2, 0.4, 0.6	0.0	6 / 4

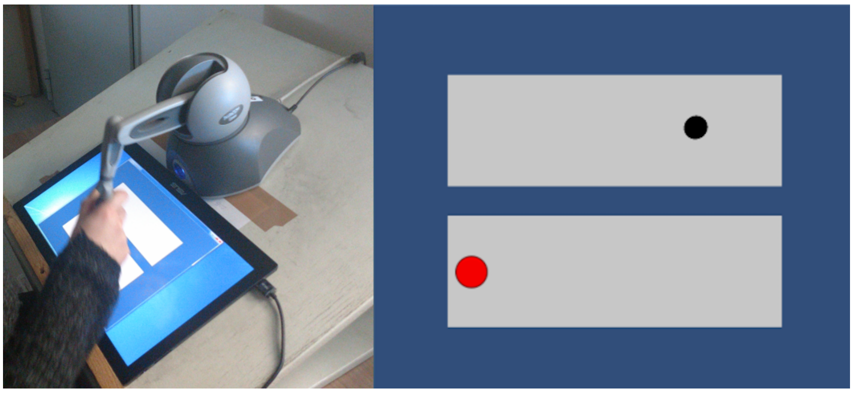

Figure 6.2: Experimental setup. **Left**: the setup with the Omni device. **Right**: Visual feedback during texture exploration (see text).

Along with the texture image, we manipulated the texture level, the stiffness and dynamic friction. Static friction was set to zero. Altogether, the study included 17 different stimuli, which were obtained by changing one and only one parameter value while keeping the other ones fixed at their "Reference Value" (see Table 6.2). To limit the total number of different stimuli pairs in this study, we combined stimuli that differed by one of the parameters pairwise, which yielded a total number of 28 pairs with different stimuli and 17 pairs with the same stimulus. Each pair with different stimuli was presented twice for each participant, with each stimulus presented once on the top surface and once on the bottom one. In addition, at the beginning of the experiment, we also presented all stimuli once, with the same stimulus on both surfaces, to let participants familiarize with the procedure. As a result, the total number of trials was 90.

The seventeen familiarization trials were always presented first. Then the remaining 73 ($= 2 \times 28 + 17$) pairs were presented in a random order in a manner that insured that all 28 pairs with different stimuli would be presented once before the second presentation. The randomization was different for every participant. The familiarization trials were included to allow the participants to experience all stimuli and calibrate their responses. Participants were informed that the stimuli were the same. All these trials have been removed from the analyses. The similarity ratings were codified by a number between 0 (equal) and 1 (completely different). The similarity responses were normalized by dividing the responses by the mean of each participant across all stimuli. Then, we compared the distribution of the response and built a similarity matrix for each parameter, pooling responses from all participants. For the MDS analysis, the similarity matrices were made symmetric by averaging the responses to the presentation of each pair of stimuli and by setting the diagonal of the matrix to 0. Then we used classical (metric) MDS with the additive constant to build the scale (Torgerson, 1958). We found no difference when the additive constant was not included or when using non-metric MDS techniques (results not shown). To measure the goodness-of-fit we used Equation GOF.

$$\mathrm{GOF} = \frac{\sum_{i=1}^{p} |\lambda_i|}{\sum_{i=1}^{n} |\lambda_i|} \tag{6.1}$$

where p is the dimension of the scale and n the number of stimuli, and which represents a measure of agreement for the proportion of the general distance matrix explained by the k-dimensional space (Mardia, 1978). All analyses were conducted in R (R Core Team, 2019).

6.1.3 Results

Participants tended to use the whole scale to respond. The inferior range limit was always zero and the superior range limit was 0.82 or above. The mean of the similarity responses for each participant ranged from 0.32 to 0.69, with an across-subject mean ($\pm$ SD) of 0.46 $\pm$ 0.10. This shows that the familiarization procedure was successful at uniformizing the similarity ratings. We did not find any effect of the starting position of the slider on the participants' responses. All MDS analyses were conducted with the normalized responses as is standard practice; very similar results are obtained when using the raw ratings.

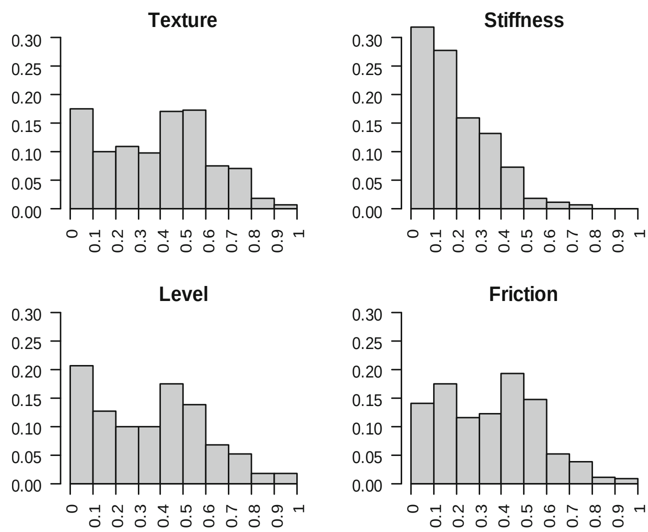

Figure 6.3: Distribution of the normalized similarity responses.

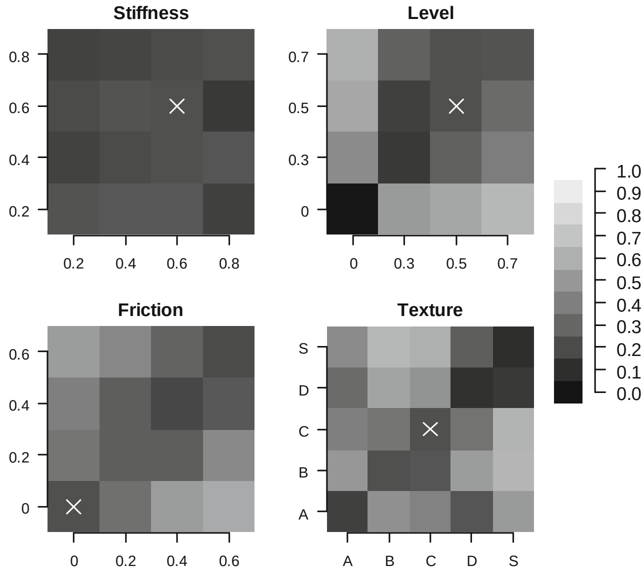

Figure 6.4: Similarity matrices foreach texture parameter. The grayscale scale represents the "equal" responses in black (value 0) and the "completely different" responses in white (value 1). The white cross identifiescomparison between the reference stimulus that is common to all matrices.

Figure 6.3 shows the distribution of the normalized responses from all participants for each texture parameter. The mean distance was markedly lower for the stiffness parameter than for the others, as confirmed by the Wilcoxon non-parametric tests ($p<0.05$, pairwise paired Wilcoxon tests with Bonferroni correction). The means were not statistically different for other pairs of parameters but Figure 6.3 suggests that distributions have a different shape, which is confirmed by χ^2 tests with Bonferroni correction (10 bins, $p<0.05$). In particular, Figure 6.3 suggests that responses distributions are somewhat bimodal, with the exception of stiffness, which is unimodal and right-skewed toward zero. Figure 6.4 shows the similarity matrices before averaging the responses for the two presentations of the same stimulus pairs. The column indicates the characteristics of the stimulus placed on the top side of the screen while the rows indicates the characteristics of the stimulus placed on the bottom side of the screen. All matrices are approximately symmetric, which indicates that similarity judgments were little influenced by the position of the textures. The main diagonal is significantly darker than the upper or lower triangular parts of the matrix for the level, friction and texture parameters (paired t-test, $p<0.01$), which reflects the fact that perceived difference was small for pairs with the same stimuli. The stiffness matrix appears less regular but is also darker and less contrasted; the diagonal is not significantly darker then the upper and the lower triangular parts. As it might already be expected from the histograms, the stiffness parameter does not seem to lead to clearly differentiated textures. To gain further insight on the perceptual differences between these textures, we analyzed these matrices with MDS techniques. First, we built a scale for each parameter by analyzing each matrix separately, assuming that the perceived distance between two textures could be represented on a single dimension. The main interest of this analysis is to compare the extent of the subscales across parameters and the position of the stimuli within each subscale. It was not possible to build a global scale encompassing all stimuli because we did not present all possible pairs of stimuli in this study. However, all subscales have the same (arbitrary) unit because all similarity judgments were obtained with the same response scale and all stimulus pairs were presented together in a random order.

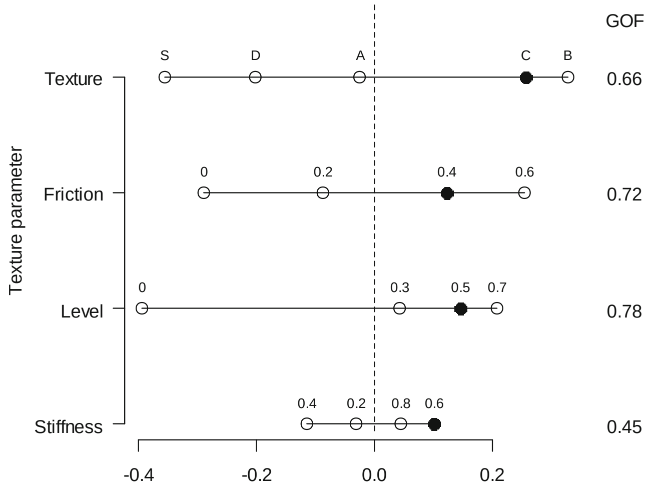

Figure 6.5: Unidimensional scales of texture parameters.

Figure 6.5 shows the subscales obtained for each parameter. The range of the subscale for the stiffness is clearly smaller than for the other subscales, confirming the fact that the textures with different stiffnesses were little differentiated. The position of the stimuli within each subscale indicates the perceived distances between them. Interestingly, the subscale for the texture level indicates that there is a large perceptual distance between a completely smooth surface (texture level = 0) and slightly rugged surfaces (texture level = 0.3), suggesting that a logarithmic scale might have been more appropriate to samples this physical dimension. In contrast, the perceived distance between texture levels varying from 0.3 to 0.7 is smaller. Note that two participants reported emphasizing the difference if one of the two stimuli was smooth. The goodness-of-fit of these one-dimensional subscales is not very high (less than 0.8, as shown in Figure 6.5). In particular, the goodness-of-fit for the stiffness is markedly lower than the other parameters, which suggests that this one-dimensional representation does not account well for the distances in the similarity matrices. The stimuli within the level and friction subscales had the same order as the rendered parameters. This was not the case for the stiffness subscale where, as noted previously, stimuli tended to appear similar. For the texture subscale, the correlation with the spatial frequency (r = 0.42) and GLCM second momentum (r = −0.69) were not statistically significant (DOF = 3, p>0.05).

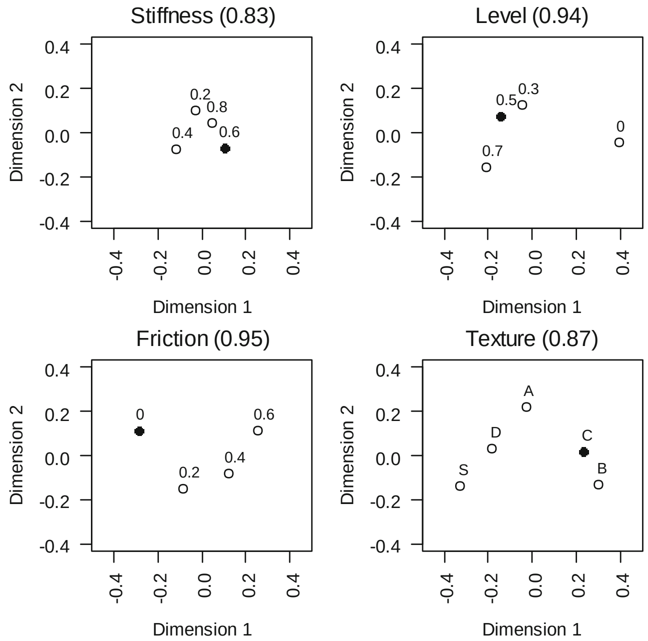

Figure 6.6: Two dimensional representations of the perceived distance between stimuli. Thegoodness-of-fit is indicated between parentheses. The black dot represents the reference stimulus which is present in all subscales.

Figure 6.6 shows the results of the two-dimensional representation of the stimuli for each parameter. The goodness-of-fit is above 0.8 for all parameters. In particular, it has markedly increased for the stiffness and texture parameters. For the texture level and friction parameters, the improvement reflects the isolation of the smooth or frictionless stimulus (0) while the other ones are relatively well aligned (see also Figure 6.4). For the texture parameter, the two middle frequency textures (B and C) are separated from the other texture, which include the lowest (A) and highest (D) frequency with the smooth surface (S). It is possible that the highest frequency texture resembles more to the smooth surface than the intermediate frequencies used in this study.

6.1.4 Discussion

The objective of this study was to assess the perceptual impact of various parameters used to render textures. The subscales for the texture level and the texture pattern had the largest extension, indicating that variations of those features had the largest perceptual influence. For texture level, the grouping of the non-smooth textures suggests that lower values (or a logarithmic scale) could and should be used, even though values smaller than 0.3 did not appear very different from a smooth surface at first. For the texture pattern, the large extension and relative homogeneous distributions of the stimuli in the one- and two-dimension subscales are interesting (and unexpected) result because the choice of the patterns is not optimal. As a matter of fact, the spatial frequency of the sinusoidal pattern ranged from 0.19 to 1.7 cycle/mm), which corresponds to a period ranging from 5.3 mm and 0.6 mm. Since the scanning movement velocity was, on average, 150 mm/sec, this spatial frequency corresponds to a temporal frequency ranging from 28 Hz to 253 Hz, which is clearly above the rendering abilities of the Omni device. Despite this fact, the different texture stimuli did not appear as indistinguishable noise. While it is unlikely that the Omni device rendered this pattern in an exact sense, this result shows that the vibratory patterns produced by the Omni device could be distinguished. From an applied point of view, this is quite interesting as it opens the range of values that might be used. Stiffness was the rendering parameter that affected perceptions the least. There might be several reasons why this is the case. First, it should be noted that users tend to rely on final force to discriminate stiffness (Tan et al., 1995) and to maintain a constant penetration force when stroking virtual surfaces, in particular if the virtual surfaces have a stiffness that is in the middle range such as in our experiment (Choi et al., 2005; Walker and Tan, 2004). Second, it is also known that the proprioceptive system does not give reliable information on the absolute position of the hand in space (Jones and Hunter, 1990). While the hand could be seen in our experiment, the precise distance from the screen was not easy to judge visually. In fact, in the sagittal direction, it has been shown that the depth direction depends more on proprioception than on vision (van Beers et al., 2002). These two observations together suggest that the participants to our study did not differentiate between the different stiffnesses because they basically felt the same interaction force when exploring these surfaces. It is possible that stiffness might have had a major impact if the participants had been allowed to tap the surfaces with the stylus, an exploratory strategy that yields considerable information about surface compliance (LaMotte, 2000). Another interesting result of this study is the marked goodness-of-fit improvement that came from adding a second dimension. While the increase was largest for the stiffness parameter, all stimuli remained grouped together and the improvement might simply reflect a lack of discriminability between the stimuli in the first place. Otherwise, this improvement suggests that changing the value of one rendering parameter might have an impact on more than one perceptual dimension, confirming that texture perception like roughness is a complex perceptual phenomenon (Kornbrot et al., 2007).

A limitation of this study is that it was not possible to build a global scale with all stimuli together, which would have permitted to see whether the different rendering parameters impacted the same perceptual dimension or not. Obtaining such a scale would require to obtain similarity judgments between pairs of stimuli that differ by more than one parameter. The results of this study might be used to select these stimuli in future work. Another limitation of this paper is that we did not analyze the actual physical interactions occurring during the exploration. It would be interesting to know the vibratory patterns associated with the different textures in light of the results of this study.

6.2 Second Study

As noted at the beginning of the chapter, percepts are often influenced by more than physical attributes. For example, in the size-weight illusion (Charpentier, 1891), the perceived weight of the object is larger if the size of the object is smaller. A limitation of the previous study was that only one parameter was manipulated at a time, which prevents knowing whether the perceived difference between virtual textures belongs to same or different psychological dimensions.

Classical studies with real textures suggest the perceived distances between stimuli could be represented in a 3D psychological space, where the two main dimensions correspond to roughness-smoothness and hardness-softness dimension respectively (Holliins et al., 1993; Hollins et al., 2000; Okamoto et al., 2012).

The goal of this second experiment was to find the number of dimensions and the perceptual organization of the virtual textures. To that end, we manipulated simultaneously three of the four parameters we used to define virtual textures in the first experiment: Spatial Frequency, Contrast Level and Dynamic Friction. We used four different values for each parameter that was manipulated, which yielded a set of $4^3 = 64$ stimuli. We did not manipulate Stiffness because the first experiment indicated that this parameter was less well differentiated or perceived than the other three. For this reason, the Stiffness was kept constant across all stimuli.

In the first experiment, we used the paired comparisons method to obtain the dissimilarity judgments. In this method, each possible pair of stimuli is presented to the subject in succession. For n stimuli, there are $n(n-1)/2$ pairs possible. While this method has the advantage of simplicity, presenting each pair one after the other can take a very long time. For the second experiment, using this method with 64 stimuli would require scaling `64 * 63 / 2 = 2016` pairs of virtual textures in succession, which would take about 7 hours to be completed.

Various methods have been proposed to collect dissimilarity data more efficiently.

To feasibly acquire this dissimilarity matrix, we tried two different methods; the Free Sorting Task (described in Section 6.2.1) and the Spatial arrangement method (SpAM) (described in Section 6.2.2).

6.2.1 Free Sorting Task: a Pilot Experiment

In the "Free Sorting Task", the subject is asked to sort the stimuli in groups or categories. Then, a dissimilarity score of 0 is given to stimuli that are in the same group, and 1 to stimuli in different groups (see Chollet et al., 2014 for a detailed description of the method and its variants). This methods does not impose a dimensional structure as the subject is free to define as many groups as desired. One problem with the method is that it does not provide a fine measure of the dissimilarity since stimuli are either inside the same group or not (but more time-consuming solutions that solve this problems have been proposed, like the Descendant Hierarchical Sorting task, as described by Chollet et al.).

In this pilot we tried to adapt the Free Sorting Task method (shown in Figure 6.7) to a task with an haptic device. To shorten the duration of the experiment, we used two monitors; the first monitor displayed a 8 by 8 checkerboard that represented the virtual textures arranged in a random order, the second monitor was a flat screen placed under the haptic device and represented a large rectangle which could be used to explore the virtual texture and 10 squares representing 10 possible groups. In the haptic VE, the checkerboard was placed horizontally above the texture plane.

When the stylus was raised above the checkerboard plane, it was possible to select one of the cells by touching it for a short time interval. The selection of a cell caused the checkerboard plane to disappear in the VE and let the subject feel the selected texture on the lower plane. Then, the participant could associate the textures by touching the corresponding square. A letter representing the selected group was

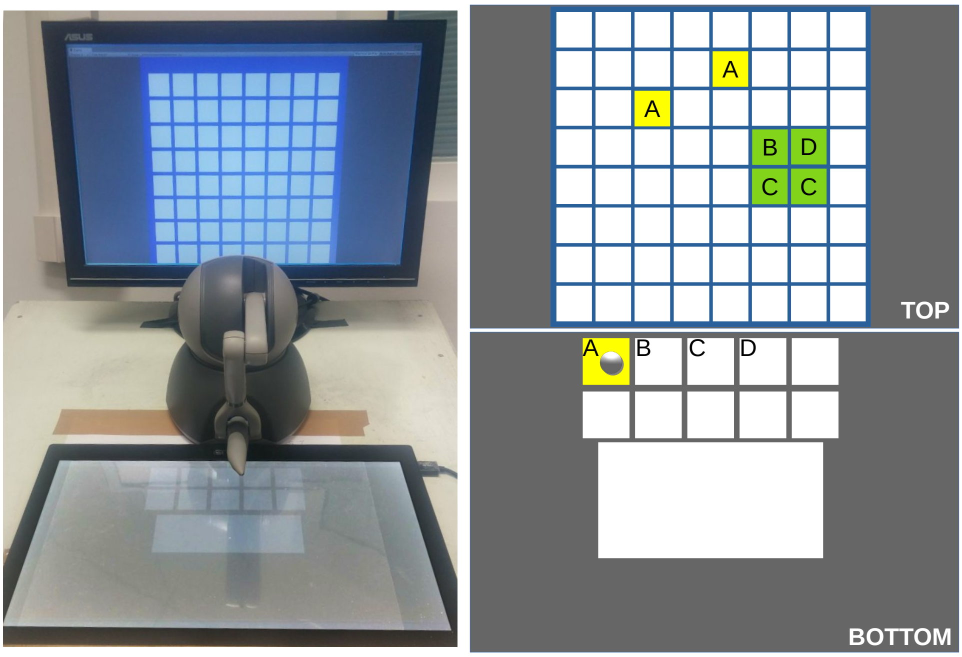

Figure 6.7: Pilot experiment. A Free Sorting Task where the subject has to assign each of the 64 textures to a group. In the upper screen the subject can see the matrix of all 64 stimuli. In the monitor below the subject can see touch a surface and assign it to the desired group.

displayed in the cell array to indicate the group membership of the cell. Moreover, by using the button on the stylus, it was possible to move a texture from one group to another group, in cases in which the subject changed his mind.

Subjects were required to associate all textures with a group. The task could be completed using only a subset of the 10 available groups. The results of the pilot experiments revealed several problems. First, the participant had problems remembering the properties of virtual textures in each group and often needed to re-explore the texture in the groups, which took time. Second, participants found that the interaction needed to select the textures and associate them with a group was too complex. For these reasons, we decided not to use this method in our experiment.

Instead, we decided to use a method that is more similar to the SpAM method (described in Section 6.2.2). The experimental procedure is described in Section 6.2.3.

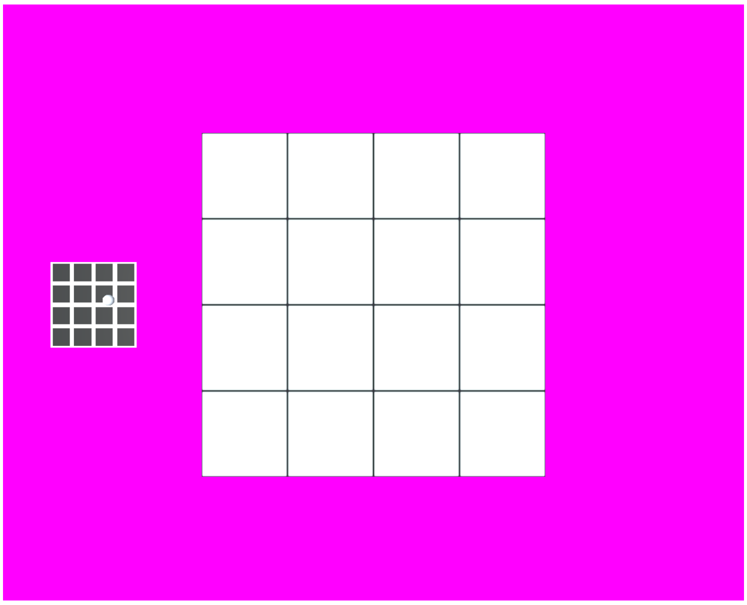

Figure 6.8: Virtual Texture Perception Experiment. On the left, the subject can select a texture to explore. On the right, s/he can assign the texture to one of the cells of the grid. Similar textures should be placed near and different ones should be placed far.

6.2.2 The Spatial Arrangement Method

Another efficient way of collecting a dissimilarity matrix is the Spatial arrangement method (SpAM), which was first proposed by Goldstone (1994). Participants have to place the stimuli on a surface, so that the distance between them would reflect the perceived dissimilarity. In principle, this method allows one to obtain graded information on the dis/similarity of the whole set of stimuli simultaneously. A potential problem with this method is that it imposes a two-dimensional structure on the dissimilarity judgment since it involves arranging the simulation on a two-dimensional plane.

The methods of the triads is a variant or special case of the SpAM method, where the observer must arrange the position of three stimuli and the distance between them is taken as a measure of their dissimilarity. Since only three objects are involved, the method does not impose any constraint on the relative distance between the objects. Moreover, the method might be more efficient than the method of paired comparison because it allows one to collect information on three pairs at each trial.

Hout et al. (2013) compared the SpAM method, the method of paired comparisons and the triad method using visual stimuli that included both two-dimensional and three-dimensional structures. Since the SpAM method involves arranging objects on a two-dimensional plane, they expected that the method would perform well for two-dimensional stimuli but were unsure about three-dimensional structures. They found that the three methods provided solutions with roughly comparable organization. The SpAM method was superior to the other methods for two-dimensional stimuli. For three-dimensional stimuli, it generated solutions that were comparable to the other techniques. In their experiments, participants arranged sets of 25-27 stimuli in about 5 minutes, compared to the 25-30 minutes necessary for the pairwise method or the triads method.

In our experiment, we needed to include 64 stimuli. This large number implies 2016 judgments of dissimilarity are needed to fill completely the dissimilarity matrix. While in the original method (Goldstone, 1994) participants were asked to place (visual) stimuli on the screen with their proximity proportional to their similarity, we were constrained by the small haptic workspace. To deal with this limitation, we divided the 64 textures into 16 subsets (or *cliques*) of 16 textures, with each stimulus present in 4 different cliques. Those repetitions are needed because the problem of finding an optimal set of cliques that covers all possible pairs with the minimum number of repetitions is a NP hard problem (Wu and Hao, 2015).

If the pairs in each clique were all different, one would obtain `16 * 15 / 2 = 120` similarities for each clique. The full set of 16 cliques would cover a total 1920 different similarities, which would correspond to 95 % of the complete dissimilarity matrix. Unfortunately, having all 16 cliques of unique pairs is theoretically impossible, and some pairs needs to be presented more that once.

This means that with 16 cliques we could not cover the whole dissimilarity matrix (i.e. each subject's dissimilarity matrix is sparse), but does not constitute a problem for the experimental design. First, the information in the similarity matrix is highly redundant given that the responses are usually based on a representation of the stimuli with a small number of dimensions. Second, the 16 cliques were all different across subjects, meaning that it was possible to reconstruct a full dissimilarity matrix from the data of all subjects.

At the end of each trial, all the Euclidean distances between the X and Y position of the textures on the grid were used as a measure of the similarity judgment.

6.2.3 Methods

Participants 20 right-handed subjects (F = 13, M = 7, age = 23.25 ± 2.86 years) took part to the experiment. Subjects were recruited on a voluntary basis at Università Vita-Salute San Raffaele.

Task For each trial, the participant had to place 16 textures on a 4 by 4 grid so that similar textures would be close together and dissimilar textures would be far from one another. The grid covered most of the available workspace, which has a maximum depth of about 13 cm, but avoiding positions too close to the border where the quality of the haptic rendering decrease considerably (Tatti et al., 2014). Initially, the textures were represented by 16 small squares on the left side of the grid (as shown in Figure 6.8) and could be picked and placed on the grid using the haptic device. Once a texture was selected, it was possible to explore it freely on the left grid. This allowed the subject to compare fast between different textures so that they had to rely less on working memory. Texture could be placed on the right grid at will. It was also possible to change the position of a texture as many times as desired.

Stimuli Each texture was defined by three parameters: contrast, spatial frequency and friction, which could each take one of four possible levels, which yielded a total of 64 textures. Geometrically, it is possible to represent the 64 textures in a 3D space, where each dimension corresponds to a different parameter. In this space, the 64 textures can be represented as a small cube embedded in a large cube. Each virtual texture was represented by a small tile (2.5 cm by 2.5 cm) in the VE. The size of the tile allowed the participant to explore the virtual texture by making small circular movement with the stylus.

Table 6.3: Parameter used for generating stimuli, with respective values

Parameter	Values
Level	0.0, 0.3, 0.5, 0.7
DynamicFriction	0.0, 0.2, 0.4, 0.6
Frequency	0.0, 0.5, 1.0, 2.0

6.2.4 Data Analysis

An oversight in designing the stimuli, which became clear only when analyzing the data was that the original set of 64 stimuli included only 40 **different** stimuli. As a matter of fact, when the texture level is 0, it is not possible to feel the ridges that are present in textures with non-zero frequencies and when the frequency is 0, the texture level has no effect. In other words, a Frequency or a Texture Level of 0 correspond to a smooth surface and it is not possible to distinguish them since they are rendered in the same manner. As a results, for each level of friction, the stimulus set included only 10 different stimuli: a smooth surface and 9 virtual textures that corresponded to the combinations on non-zero spatial frequencies and non-zero contrast levels. In total, the stimulus set included only 40 different of stimuli.

For the MDS Analysis, we used both a 64 by 64 matrix and a 40 by 40 dissimilarity matrices. Both approaches yield very similar results. In the case of the 64 by 64 matrix, we found that the stimuli that corresponded to the same stimuli were close one to another. In the case of the 40 by 40 matrix, these stimuli are represented by a single point, which had the same location in the perceptual space. For the sake of brevity, we present only the results based on the 40 by 40 matrix.

Each clique provided a subset of all possible pairs. For each clique, we computed the Euclidean distances between each pair of stimuli. When the same pair was present in multiple cliques across subjects, the mean distance was used. In fact, while in our experiment the stimuli in the 16 cliques corresponded collectively to approximately 70 % of all different pairs possible, each pair was presented to at least 5 subjects and on average to 12 subjects. 20% of the pairs was presented at least once to each subject.

Stimuli with a Frequency or a Texture Level of 0 where considered to be the same, resulting in a 40 by 40 dissimilarity matrix with a theoretical minimum and maximum distance of 1 and $3\sqrt{2} \approx 4.24$ respectively.

6.2.5 Results

The observed minimum and maximum mean distances in the dissimilarity matrix were 1.04 and 2.97 respectively.

Metric MDS. Dissimilarity matrices were used to compute a Metric MDS with a number of dimensions (N) up to 10. Values for the Stress function are shown in Figure 6.10. The decrease is high when adding the second dimension, while is low

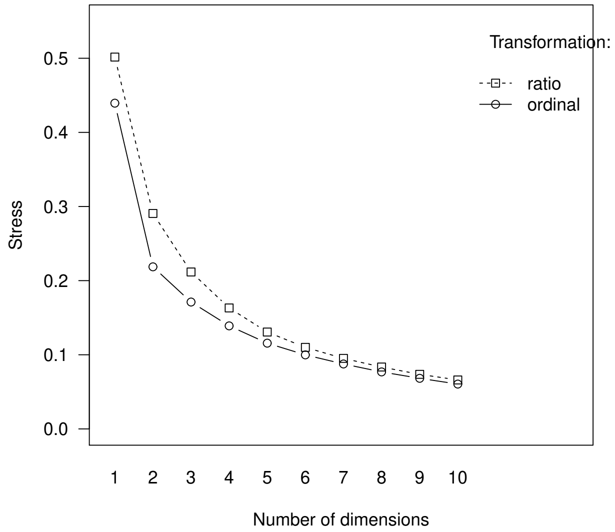

Figure 6.9: Stress for metric MDS solutions up to 10 dimensions. Two dimensions describe the perceived differences in the stimuli well.

for the third dimension on, suggesting that a model with two dimensions is good to describe the perceived differences in the stimuli.

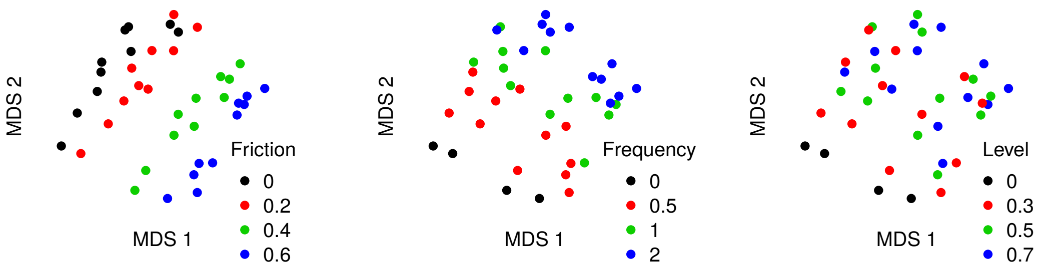

Figure 6.10: Metric MDS for two dimensions. The color code represent value of the three parameters.

This solution with two dimensions is shown in Figure 6.10. The scatterplots in the three panes show the same data, with colors that depends on the three parameters. It is possible to note that one dimension reflects well the changes in the Dynamic Friction values (first pane) and the other dimension reflects changes in the Frequency. Changes in the Texture Level partially overlap with the Frequency, and except when the Texture Level is equal to 0 (smooth surface) differences are far less clear. Classic MDS Analysis gives the same results.

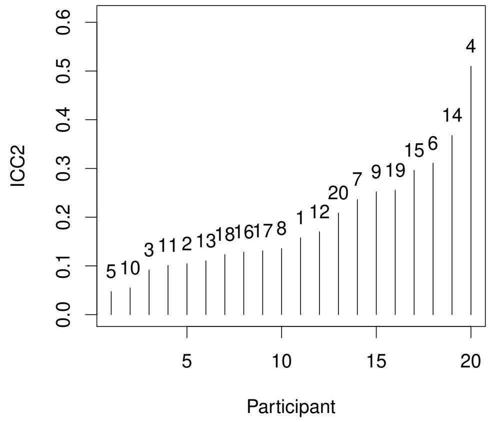

Figure 6.11: Consistency (ICC) of the ratings of the same pairs in different subsets for each participant.

Consistency of texture placement within participants. Each subject was presented some pairs multiple times. Those repetitions can be used to evaluate the consistency of the responses for each subject. We used Intraclass Coefficient (ICC) to measure the consistency of the repeated pairs, in order to estimate the reliability of the placement of textures. Results are shown in Figure 6.11. These measures suggest a low agreement between the relative placement of two different textures across grids. A reason is that the placement of two textures is strongly constrained by the geometry (all 16 textures must be put on a 2D grid) and the presence of different textures in different grids (not all grids where the same pair is presented are the same).

Position dependency of objective parameter values. It is possible to investigate whether a stimulus parameter varied smoothly as a function of the position on the grid. If this is the case, this would mean that the participant paid attention to the dimensions of the stimulus that corresponds to that texture parameter. To that end, we examined how well stimulus position predicts each parameter of the virtual textures by regressing the value of texture parameter on the coordinates of the virtual texture in the grid:

$$\frac{\pi_i}{pi_{max}} = \beta_i + \beta_{x_i} x_i + \beta_{y_i} y_i + \beta_{x_i} x_i y_i \quad (6.2)$$

where π_i/π_{max} is the i-th parameter (Level, Dynamic Friction or Spatial Frequency) rescaled by its maximum value, x_i and y_i are the coordinates of the texture and β are the coefficients of the regression. Parameter values were rescaled by their maximum value so that residuals might be compared across parameters (normalizing by centering and dividing by standard deviation gives the same results).

For each grid, we computed the Residual Sum of Squares (RSS) for each parameter separately to assess how well the texture position on the grid reflected the texture parameter. Results are shown in Figure 6.12. The RSS varied across subjects, in particular for the Dynamic Friction and Frequency parameters. Subjects with small RSS paid more attention to the stimulus dimension that corresponds to this parameter.

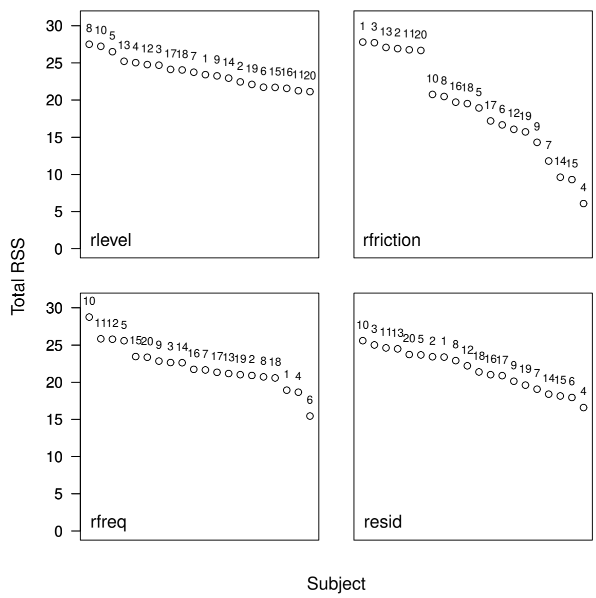

Figure 6.12: Total RSS of separate regressions of texture parameters on texture position for each participant. The subjects (horizontal axis) are ordered accordind to the sum of the RSS across all grids. The last panel shows the average RSS across the three texture parameters.

A Principal Component Analysis (PCA) on the RSS of the three stimulus parameters is shown in Figure 6.13. The first two principal components explain 78.8 % and 15.4 % of the total variance respectively. It can be seen that the first PC corresponds to Friction and the second one to Frequency, and that subjects performed differently on these axes.

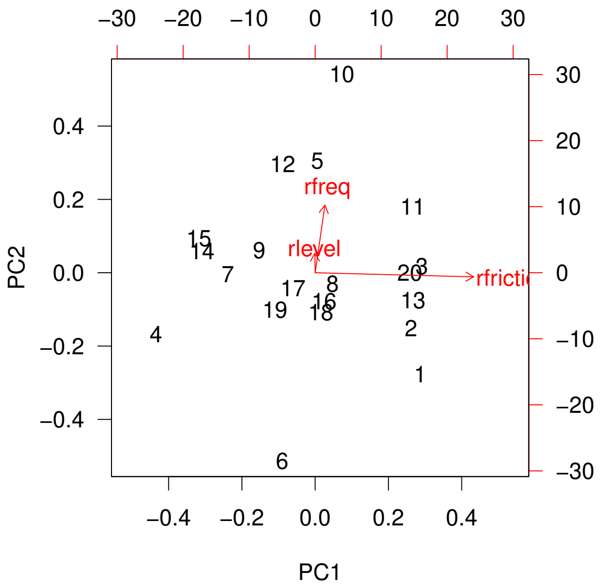

Figure 6.13: PCA of RSS for the RSS of the stimulus parameters for each participant. Red arrows represent texture parameters. The first Principal Component corresponds to the Friction, the second component to the Spatial Frequency.

6.2.6 Discussion

The results of this experiment show that perceptual differences between virtual textures are well represented in a two dimensional space. The first dimension of this space corresponds to two parameters of the virtual textures, i.e. the Spatial Frequency and Contrast Level, while the second dimension corresponds to the third parameter, Dynamic Friction.

The main results, from an applied point of view, is that one cannot use Spatial Frequency and Texture Level independently to produce virtual textures that are perceptually different: increasing spatial frequency has the same effect as increasing the contrast.

The first dimension obtained in this experiment is similar to the smoothness-roughness dimension that is typically obtained in MDS analysis with real textures or materials (Hollins et al., 2000; Okamoto et al., 2012).

The second dimension, which corresponds to friction in our experiment, appears different from hardness-softness, the second dimension of MDS with real objects. A possible reason is that we did not manipulate the stiffness of the contact surface in this experiment. In fact, the Friction dimension is probably better related to the stickiness dimension, which has appeared as a third dimension in some experiments with real textures (Bensmaïa and Hollins, 2005; Holliins et al., 1993).

From a methodological point of view, we introduced a new method to collect similarity data for multidimensional scaling with virtual textures. Clearly, the time gained with this method was smaller than the gain obtained by Hout et al. (2013), who found that arranging 25-27 visual stimuli displayed on a monitor screen took roughly only 20% of the time necessary to obtain the same information using the method of pairwise comparisons. In our study, the time used to complete the experiment was on average 55 minutes, about 13% of the time that would be expected if the pairwise comparison method had been used for the same number of pairs.

The SpAM method has been mainly used in visual domain, where it allows the observer to see and compare all stimuli simultaneously. In this task, however, the need to touch the stimuli in reduce the efficiency of the method in the haptic modality because the stimuli must be explored serially, one after the other. This constraints is particularly strong when using a haptic device such as the Omni Phantom device, which has only a single point of contact. For the haptic domain, this same difficulty is present also with the paired comparison method. The advantage of the SpAM method over the paired comparison task in the haptic domain might be given by the fact that instead of having to alternate the exploration/perception phase and judgment phase, the SpAM method might change the task to a feature search task. The participant can explore sequentially many surfaces rapidly until two similar are found, and repeat this task until all stimulus are placed, thus reducing the judgment overhead. Also, once two similar surfaces are found, because of the transitive property it is not strictly necessary to compare each of them with all the remaining ones.

Trying to adjust the number of stimuli presented for each trial might further reduce the exploration time.

Given the oversight that caused us to use 40 different stimuli in stead of the 64 expected, it is possible to reduce the number of cliques from 16 to 8 while maintaining approximately the same coverage of the full dissimilarity matrix. We are currently repeating this experiment with the same 40×40 matrix, but never presenting the same stimulus twice in the same grid and thus reducing the number of cliques from 16 to 8. The advantages are twofold. First, it halves the experiment time. Second, when the same stimulus is compared a distance of 0 cannot be chosen (it is not possible to place two stimuli in the same position on the grid), hence same-stimuli dissimilarity was artificially increased.

7 Conclusions

This research was part of the weDRAW European H2020 Project. As such, we focused on *education*, *serious game* and *multisensory integration*. Since our lab works on haptic perception and technologies, our role inside the weDRAW Project was to help in the development of an integrated platform that needed to provide visual, auditory and haptic sensory feedback.

There are many reasons why haptics had a central role in the weDRAW Project, including the role that haptic plays in learning theories such as *constructivism* and *embodied cognition* and the advantages that using a VE might provide over real world situations. However, to really exploit those VEs, further research is needed. First, there is still lack of an easy way to add force-feedback to VEs. While many haptic software libraries exist, their features and ease of use is still far from the one provided by game engines. Second, how to convey information with the haptic sense and how those information are combined with other senses is still unclear.

We addressed those problems in two ways. First, we realized two haptic plugins, each aimed at different software platforms. The first plugin provided low-level haptic rendering integration to EyesWeb, a software platform focused on data generation and analysis for multimedia installations. The haptic library created, H4F, could interface with EyesWeb through the OSC protocol. The second plugin was needed as requirements inside the project changed. The switch to the Unity3D game engine and the need for more complex haptic rendering algorithms required the development of a second library, HPGE. This library can be easily integrated with this Unity3D. We decided to keep a hierarchical structure that abstract away from Unity3D itself so that with no changes to the core of the library it should be possible to add support for other game engines.

Second, we applied those technologies in two research fields; multisensory integration and virtual haptic rendering perception. The aim of the multisensory integration experiment was to try to partially answer to the question on how information coming from different senses are integrated. As the focus of the weDRAW Project was on Geometry and Mathematics, our experiment was on orientation perception.

Others experiments presented in this thesis regard haptic texture rendering and perception. Those studies can have an important role in order to understand current capabilities and limitation of haptic devices and haptic algorithms, but also, in the future, to provide basic knowledge on how much information can be transmitted to users in a multisensory VE. With this knowledge it will be possible to build rich learning environments that take advantages of *gamification* and multisensory feedback to ease learning.

7.1 Work in Progress

Studies presented in this thesis had some limitations. To address these limitations and to try to answer to other interesting questions, we run other experiments during the last PhD year. Data have been collected and will be analyzed in the future weeks. Details on those experiments are given below.

Haptic Perception. The two texture rendering perception experiments presented has some limitations. The first experiment is limited by the fact that we were not able to compare pairs of all the possible stimuli, as we needed to reduce the time required for similarity judgments. To overcome this problem, the second experiment used a different method, a modified version of the SpAM. We asked participants to place similar stimuli near and different stimuli far on a grid. However, an oversight in the design of the experiment led to including the same stimulus multiple times in each grid. We have completed a replication of the study without the extra stimuli. Preliminary analysis suggests that results are the same, while the experiment duration is reduced to about forty minutes.

Rendering Quality. Another limit shared by both this studies is that haptic rendering can be affected by the position of the stimulus in the workspace, as suggested by previous studies. To investigate this possibility, we did a similarity judgment task with the same stimuli as the previous experiments. We manipulated the position of the stimulus explicitly and compared results between two different haptic devices. Preliminary analyses suggest that stimulus location and the haptic device used have a limited effect.

Sonification. The main limit of this the ellipse multisensory experiment was the difficulty with which stimuli were perceived in the auditory condition. This difficulty cannot be ascribed to the auditory modality itself as much as it can be ascribed to the sonification strategy used. We now evaluated different sonification strategies in two different experiments; a bisection task and a localization task. Sonification strategies included changing the pitch, volume, left-right balance and beep frequency of a sine wave. Performances were compared between musicians and non-musicians.

Stereoscopic Vision and Haptics. Preliminary testing with the SpaceShape game with children (see Section 4.2.4) revealed that children need some practices to control the movements in the 3D VE. The main difficulty was to control the depth of the movement in the VE, which stem from the absence of strong depth cues on the perceived scene on the 2D display. The BallonGame was designed to address this issue, by letting the children experience the VE before playing the SpaceShape game. In a follow-up study, we investigated the effect of stereoscopic vision in adults with the BalloonGame and with the experiment described in Section 4.2.5, Stereoscopic+Haptic. Both these experiments were run in collaboration with University of Genova.

Acronyms

ANOVA	analysis of variance 55–58
API	Application Programming Interface 12, 13, 24, 27, 30, 31, 34–36
DL	Discrimination Limen 54–58
DoF	Degree of Freedom 9–11, 13
GUI	Graphical User Interface 24–26, 39
H4F	Haptic For Fun 27, 30, 31, 33, 91
HPGE	Haptic Plugin for Game Engines 9, 33, 44, 67, 91
IDE	Integrated Development Environment 25, 26, 36, 39
IIT	Istituto Italiano di Tecnologia 1, 4, 50
LED	Light-Emitting Diode 27, 31, 33, 50, 51, 63, 65
MDS	Multidimensional Scaling 5, 68, 70, 73, 74, 76, 85, 86, 89
MLE	Maximum Likelihood Estimation 53, 59
MOM	Modified one-step M-estimator 56
OSC	Open Sound Control 26, 28, 30, 91
PCA	Principal Component Analysis 88, 89
PSE	Point of Subjective Equality vi, 54–56, 62, 63
SDE	Software Development Environment 23
SG	serious game v, 2, 4, 5, 21, 34, 42
SpAM	Spatial arrangement method 80–83, 90, 92
UCL	University College of London 2, 4, 42
UDP/IP	User Datagram Protocol 28
VE	Virtual Environment 8, 11, 12, 14, 15, 17–23, 27, 39, 45, 67, 80, 84, 91–93
VR	Virtual Reality 4, 9, 10, 17–19, 21–23, 25

Glossary

C

constructivism

A theory about the nature of learning that focuses on how humans make meaning from their experiences 19, 22, 91

F

force-feedback device

A mechanical device that can produce forces, used to stimulate user's haptic sense v, 3, 4, 8–11, 21–23, 25, 27, 33, 39

free software license

Software Licenses that give the user the freedom to run, copy, distribute, study, change and improve the software 13, 69

H

haptic device

An electromechanical device involving a physical interaction with the user v, 7, 8, 10–14, 16, 18, 22, 27, 28, 84, 92

S

software library

A collection of readily-available software functions that can be used from other programs 4, 5, 12, 13, 16, 24, 25, 27, 28, 30, 31, 33–36, 39, 41, 67, 69, 91

W

weDRAW

European Project, coordinated by Monica Gori v, 1–5, 10, 19, 23, 27, 29, 42, 67, 91

Bibliography

David Alais and David Burr. The ventriloquist effect results from near-optimal bimodal integration. *Current biology*, 14(3):257–262, 2004.

Nicolò Balzarotti and Gabriel Baud-Bovy. Hpge: an haptic plugin for game engines. In *International Conference on Games and Learning Alliance*, pages 330–339. Springer, 2018a.

Nicolò Balzarotti and Gabriel Baud-Bovy. Effects of chai3d texture rendering parameters on texture perception. In Domenico Prattichizzo, Hiroyuki Shinoda, Hong Z. Tan, Emanuele Ruffaldi, and Antonio Frisoli, editors, *Haptics: Science, Technology, and Applications*, pages 138–149, Cham, 2018b. Springer International Publishing. ISBN 978-3-319-93445-7.

Nicolò Balzarotti and Gabriel Baud-Bovy. The visual, audio, haptic and multimodal perception of ellipses' shape in elementary school children. 2020.

Barbagli, F., Frisoli, A., Salisbury, K., and Bergamasco, M. Simulating human fingers: a soft finger proxy model and algorithm. pages 9–17, 2004.

Lawrence W Barsalou. Perceptual symbol systems. *Behavioral and brain sciences*, 22(4):577–660, 1999.

Gabriel Baud-Bovy and Nicolò Balzarotti. Using force-feedback devices in educational settings: A short review. In *Proceedings of the 1st ACM SIGCHI International Workshop on Multimodal Interaction for Education*, MIE 2017, pages 14–21, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5557-5.

Sliman Bensmaïa and Mark Hollins. Pacinian representations of fine surface texture. *Perception & psychophysics*, 67(5):842–854, 2005.

Robert I Bermant and Robert B Welch. Effect of degree of separation of visual-auditory stimulus and eye position upon spatial interaction of vision and audition. *Perceptual and Motor Skills*, 43(2):487–493, 1976.

Jim Blascovich. *Social Influence within Immersive Virtual Environments*, pages 127–145. Springer London, London, 2002. ISBN 978-1-4471-0277-9.

Breuer, Johannes and Bente, Gary. Why so serious? on the relation of serious games and learning. *Journal for Computer Game Culture*, 4 (1):7–24, 2010.

David Burr and Monica Gori. Multisensory integration develops late in humans. In *The neural bases of multisensory processes*. CRC Press/Taylor & Francis, 2012.

Gianni Campion and Vincent Hayward. Fundamental limits in the rendering of virtual haptic textures. In *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*, pages 263–270. IEEE, 2005.

Antonio Camurri, Shuji Hashimoto, Matteo Ricchetti, Andrea Ricci, Kenji Suzuki, Riccardo Trocca, and Gualtiero Volpe. Eyesweb: Toward gesture and affect recognition in interactive dance and music systems. *Computer Music Journal*, 24(1): 57–69, 2000.

Antonio Camurri, Paolo Coletta, Giovanna Varni, and Simone Ghisio. Developing multimodal interactive systems with eyesweb xmi. 06 2007. doi: 10.1145/1279740.1279806.

Augustin Charpentier. Analyse experimentale de quelques elements de la sensation de poids. *Archive de Physiologie normale et pathologiques*, 3:122–135, 1891.

Yi-Chuan Chen and Charles Spence. Assessing the role of the 'unity assumption'on multisensory integration: A review. *Frontiers in psychology*, 8:445, 2017.

Seungmoon Choi and Hong Z Tan. Effect of update rate on perceived instability of virtual haptic texture. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 4, pages 3577–3582. IEEE, 2004.

Seungmoon Choi, Laron Walker, Hong Z Tan, Scott Crittenden, and Ron Reifenberger. Force constancy and its effect on haptic perception of virtual surfaces. *ACM Transactions on Applied Perception (TAP)*, 2(2):89–105, 2005.

Sylvie Chollet, Dominique Valentin, and Hervé Abdi. Free sorting task. *Novel Techniques in Sensory Characterization and Consumer Profiling. Valera P. & Ares, G.(Eds.). Boca Raton: Taylor and Francis*, pages 207–227, 2014.

Jonathan Cole and Jacques Paillard. Living without touch and peripheral information about body position and movement: Studies with deafferented subjects. 1995.

François Conti, Federico Barbagli, Dan Morris, and Christopher Sewell. Chai 3d: An open-source library for the rapid development of haptic scenes. 2005.

Cuevas-Rodriguez, Maria, Poyade, Matthieu, Reyes-Lecuona, Arcadio, and Molina-Tanco, Luis. A vrpn server for haptic devices using openhaptics 3.0. pages 73–82. Springer, London, 2013.

Heather Culbertson, Juliette Unwin, Benjamin E Goodman, and Katherine J Kuchenbecker. Generating haptic texture models from unconstrained tool-surface interactions. In *2013 World Haptics Conference (WHC)*, pages 295–300. IEEE, 2013.

Heather Culbertson, Juliette Unwin, and Katherine J Kuchenbecker. Modeling and rendering realistic textures from unconstrained tool-surface interactions. *IEEE transactions on haptics*, 7(3):381–393, 2014.

Simon Davis, Keith Nesbitt, and Eugene Nalivaiko. A systematic review of cybersickness. In *Proceedings of the 2014 Conference on Interactive Entertainment*, pages 1–9, 2014.

Ton De Jong, Marcia C Linn, and Zacharias C Zacharia. Physical and virtual laboratories in science and engineering education. *Science*, 340(6130):305–308, 2013.

John Dewey. *Democracy and education: An introduction to the philosophy of education*. Macmillan, 1923.

Andreea Oliviana Diaconescu, Lynn Hasher, and Anthony Randal McIntosh. Visual dominance and multisensory integration changes with age. *Neuroimage*, 65:152–166, 2013.

Emmanuelle Dionne-Dostie, Natacha Paquette, Maryse Lassonde, and Anne Gallagher. Multisensory integration and child neurodevelopment. *Brain sciences*, 5 (1):32–57, 2015.

Djaouti, Damien, Alvarez, Julian, and Jessel, Jean-Pierre. Classifying serious games: The g/p/s model. *Handbook of Research on Improving Learning and Motivation through Educational Games: Multidisciplinary Approaches*, pages 118–136, 2011.

Sam Duffy, Sara Price, Gualtiero Volpe, Paul Marshall, N Bianchi-Berthouze, Giulia Cappagli, Luigi Cuturi, Nicolò Balzarotti, David Trainor, and Monica Gori. Wedraw: using multisensory serious games to explore concepts in primary mathematics. In *Proceedings of the 13th international conference on technology in mathematics teaching*, volume 13. 13th International Conference on Technology in Mathematics Teaching (ICTMT 13), 2017.

Marc O Ernst. Multisensory integration: a late bloomer. *Current Biology*, 18(12): R519–R521, 2008.

Marc O Ernst and Martin S Banks. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, 415(6870):429–433, 2002.

Marc O Ernst and Heinrich H Bülthoff. Merging the senses into a robust percept. *Trends in cognitive sciences*, 8(4):162–169, 2004.

Gustav Theodor Fechner, Davis H Howes, and Edwin Garrigues Boring. *Elements of psychophysics*, volume 1. Holt, Rinehart and Winston New York, 1966.

Robert Goldstone. An efficient method for obtaining similarity data. *Behavior Research Methods, Instruments, & Computers*, 26(4):381–386, 1994. ISSN 1532-5970.

Monica Gori, Michela Del Viva, Giulio Sandini, and David C Burr. Young children do not integrate visual and haptic form information. *Current Biology*, 18(9): 694–698, 2008.

Monica Gori, Valentina Squeri, Alessandra Sciutti, Lorenzo Masia, Giulio Sandini, and Jürgen Konczak. Motor commands in children interfere with their haptic perception of objects. *Experimental Brain Research*, 223(1):149–157, 2012.

LIU Guanyang, GENG Xuda, LIU Lingzhi, and WANG Yan. Haptic based teleoperation with master-slave motion mapping and haptic rendering for space exploration. *Chinese Journal of Aeronautics*, 32(3):723–736, 2019.

Hamari, J., Koivisto, J., and Sarsa, H. Does gamification work? – a literature review of empirical studies on gamification. pages 3025–3034, 2014.

Felix G. Hamza-Lup, Crenguta M. Bogdan, Dorin M. Popovici, and Ovidiu D. Costea. A survey of visuo-haptic simulation in surgical training. *CoRR*, abs/1903.03272, 2019.

Blake Hannaford and Allison M. Okamura. *Haptics*, pages 719–739. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-30301-5.

Sandra Helsel. Virtual reality and education. *Educational Technology*, 32(5):38–42, 1992.

Hmelo-Silver, Cindy E. Problem-based learning: What and how do students learn? *Educational Psychology Review*, 16(3):235–266, 2004. ISSN 1040-726X, 1573-336X.

Mark Holliins, Richard Faldowski, Suman Rao, and Forrest Young. Perceptual dimensions of tactile surface texture: A multidimensional scaling analysis. *Perception & psychophysics*, 54(6):697–705, 1993.

Mark Hollins, Sliman Bensmaïa, Kristie Karlof, and Forrest Young. Individual differences in perceptual space for tactile textures: Evidence from multidimensional scaling. *Perception & Psychophysics*, 62(8):1534–1544, 2000.

Autumn B. Hostetter and Martha W. Alibali. Visible embodiment: Gestures as simulated action. *Psychonomic Bulletin & Review*, 15(3):495–514, 2008. ISSN 1531-5320. doi: 10.3758/PBR.15.3.495.

Michael C Hout, Stephen D Goldinger, and Ryan W Ferguson. The versatility of spam: A fast, efficient, spatial method of data collection for multidimensional scaling. *Journal of Experimental Psychology: General*, 142(1):256, 2013.

Kai Huotari and Juho Hamari. Defining gamification: a service marketing perspective. In *Proceeding of the 16th international academic MindTrek conference*, pages 17–22, 2012.

Charles E Jack and Willard R Thurlow. Effects of degree of visual association and angle of displacement on the "ventriloquism" effect. *Perceptual and motor skills*, 37(3):967–979, 1973.

Mina Johnson-Glenberg. Immersive vr and education: Embodied design principles that include gesture and hand controls. *Front. Robotics and AI*, 2018, 2018.

Lynette A Jones and Ian W Hunter. A perceptual analysis of stiffness. *Experimental Brain Research*, 79(1):150–156, 1990.

Bianca Jovanovic and Knut Drewing. The influence of intersensory discrepancy on visuo-haptic integration is similar in 6-year-old children and adults. *Frontiers in psychology*, 5:57, 2014.

Petr Kadleček and Supervised Petr Kmoch. Overview of current developments in haptic apis. In *Proceedings of CESCG*. Citeseer, 2011.

Daniel Kersten, Pascal Mamassian, and Alan Yuille. Object perception as bayesian inference. *Annu. Rev. Psychol.*, 55:271–304, 2004.

P Ewen King-Smith and David Rose. Principles of an adaptive method for measuring the slope of the psychometric function. *Vision Research*, 37(12):1595–1604, 1997.

Norimichi Kitagawa and Shigeru Ichihara. Hearing visual motion in depth. *Nature*, 416(6877):172–174, 2002.

Roberta L Klatzky, Susan J Lederman, Cheryl Hamilton, Molly Grindley, and Robert H Swendsen. Feeling textures through a probe: Effects of probe and surface geometry and exploratory factors. *Perception & psychophysics*, 65(4): 613–631, 2003.

Diana Kornbrot, Paul Penn, Helen Petrie, Stephen Furner, and Andrew Hardwick. Roughness perception in haptic virtual reality for sighted and blind people. *Perception & psychophysics*, 69(4):502–512, 2007.

Katherine J Kuchenbecker, Jonathan Fiene, and Günter Niemeyer. Event-based haptics and acceleration matching: Portraying and assessing the realism of contact. In *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*, pages 381–387. IEEE, 2005.

Robert H LaMotte. Softness discrimination with a tool. *Journal of Neurophysiology*, 83(4):1777–1786, 2000.

Paul J Laurienti, Robert A Kraft, Joseph A Maldjian, Jonathan H Burdette, and Mark T Wallace. Semantic congruence is a critical factor in multisensory behavioral performance. *Experimental brain research*, 158(4):405–414, 2004.

S. J. Lederman and R. L. Klatzky. Haptic perception: A tutorial. *Attention, Perception, & Psychophysics*, 71(7):1439–1459, Oct 2009. ISSN 1943-393X. doi: 10.3758/APP.71.7.1439.

Ming C Lin and Miguel Otaduy. *Haptic rendering: foundations, algorithms, and applications*. AK Peters/CRC Press, 2008.

Ming C. Lin, USA Dinesh Manocha, and Jon Cohen. Collision detection: Algorithms and applications, 1996.

Annie Luciani. Virtual reality and virtual environment. In *Enaction and enactive interfaces : a handbook of terms*, pages 299–300. Enactive Systems Book, 2007.

Patrick Mair and Rand Wilcox. Robust statistical methods in r using the wrs2 package. *Behavior research methods*, pages 1–25, 2019.

Kanti V Mardia. Some properties of clasical multi-dimesional scaling. *Communications in Statistics-Theory and Methods*, 7(13):1233–1241, 1978.

William McMahan, Joseph M Romano, Amal M Abdul Rahuman, and Katherine J Kuchenbecker. High frequency acceleration feedback significantly increases the realism of haptically rendered textured surfaces. In *2010 IEEE Haptics Symposium*, pages 141–148. IEEE, 2010.

Gisela Mendel. Children's preferences for differing degrees of novelty. *Child Development*, pages 453–465, 1965.

Zahira Merchant, Ernest T Goetz, Lauren Cifuentes, Wendy Keeney-Kennicutt, and Trina J Davis. Effectiveness of virtual reality-based instruction on students' learning outcomes in k-12 and higher education: A meta-analysis. *Computers & Education*, 70:29–40, 2014.

Tassos A Mikropoulos and Antonis Natsis. Educational virtual environments: A ten-year review of empirical research (1999–2009). *Computers & Education*, 56 (3):769–780, 2011.

Jeff Miller. Divided attention: Evidence for coactivation with redundant signals. *Cognitive psychology*, 14(2):247–279, 1982.

Giovanni F Mjsceo, Wayne A Hershberger, and Ronda L Mancini. Haptic estimates of discordant visual—haptic size vary developmentally. *Perception & psychophysics*, 61(4):608–614, 1999.

Tania K Morimoto, Paulo Blikstein, and Allison M Okamura. [d81] hapkit: An open-hardware haptic device for online education. In *2014 IEEE Haptics Symposium (HAPTICS)*, pages 1–1. IEEE, 2014.

Marko Nardini, Peter Jones, Rachael Bedford, and Oliver Braddick. Development of cue integration in human navigation. *Current biology*, 18(9):689–693, 2008.

Marko Nardini, Rachael Bedford, and Denis Mareschal. Fusion of visual cues is not mandatory in children. *Proceedings of the National Academy of Sciences*, 107(39): 17041–17046, 2010.

Marko Nardini, Jennifer Bales, and Denis Mareschal. Integration of audio-visual information for spatial decisions in children and adults. *Developmental science*, 19(5):803–816, 2016.

Patricia A Neil, Christine Chee-Ruiter, Christian Scheier, David J Lewkowicz, and Shinsuke Shimojo. Development of multisensory spatial integration and perception in humans. *Developmental science*, 9(5):454–464, 2006.

Kofi Nyarko, Tanya Capers, Craig Scott, and Kemi Ladeji-Osias. Network intrusion visualization with niva, an intrusion detection visual analyzer with haptic integration. In *Proceedings 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. HAPTICS 2002*, pages 277–284. IEEE, 2002.

Shogo Okamoto, Hikaru Nagano, and Yoji Yamada. Psychophysical dimensions of tactile perception of textures. *IEEE Transactions on Haptics*, 6(1):81–93, 2012.

Miguel A Otaduy and Ming C Lin. Introduction to haptic rendering. In *ACM SIGGRAPH 2005 Courses*, page 3. ACM, 2005.

Thomas U Otto and Pascal Mamassian. Noise and correlations in parallel perceptual decision making. *Current Biology*, 22(15):1391–1396, 2012.

Partha Sarathi Paul, Surajit Goon, and Abhishek Bhattacharya. History and comparative study of modern game engines. *International Journal of Advanced Computed and Mathematical Sciences*, 3(2):245–249, 2012.

Jérôme Perret and Emmanuel Vander Poorten. Touching virtual reality: A review of haptic gloves. In *ACTUATOR 2018; 16th International Conference on New Actuators*, pages 1–5. VDE, 2018.

Jean Piaget. *The language and thought of the child*, volume 5. Psychology Press, 2002.

Myrthe A. Plaisier and Jeroen B. J. Smeets. How many objects are inside this box? pages 240–244. Institute of Electrical and Electronics Engineers, Inc., 2017.

Veljko Potkonjak, Michael Gardner, Victor Callaghan, Pasi Mattila, Christian Guetl, Vladimir M Petrović, and Kosta Jovanović. Virtual laboratories for education in science, technology, and engineering: A review. *Computers & Education*, 95: 309–327, 2016.

Rachel Proffitt. Gamification in rehabilitation: Finding the "just-right-challenge". In *Handbook of Research on Holistic Perspectives in Gamification for Clinical Practice*, pages 132–157. IGI Global, 2016.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019. URL https://www.R-project.org/.

Bruno H Repp and Amandine Penel. Auditory dominance in temporal processing: new evidence from synchronization with simultaneous visual and auditory sequences. *Journal of Experimental Psychology: Human Perception and Performance*, 28(5):1085, 2002.

Liliana Rincon-Gonzalez, Jay P Warren, David M Meller, and Stephen Helms Tillery. Haptic interaction of touch and proprioception: implications for neuroprosthetics. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 19(5):490–500, 2011.

Lawrence D Rosenblum, Mark A Schmuckler, and Jennifer A Johnson. The mcgurk effect in infants. *Perception & Psychophysics*, 59(3):347–357, 1997.

Diego C. Ruspini, Krasimir Kolarov, and Oussama Khatib. Haptic interaction in virtual environments. *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97*, 1:128–133 vol.1, 1997.

Kenneth Salisbury, David Brock, Thomas Massie, Nick Swarup, and Craig Zilles. Haptic rendering: Programming touch interaction with virtual objects. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 123–130. ACM, 1995.

Erich Schröger and Andreas Widmann. Speeded responses to audiovisual signal changes result from bimodal integration. *Psychophysiology*, 35(6):755–759, 1998.

Shams, Ladan and Seitz, Aaron R. Benefits of multisensory learning. *Trends in Cognitive Sciences*, 12(11):411–417, 2008. ISSN 1364-6613.

Alan C Skidgell, Sam L Witryol, and Philip J Wirzbicki. The effect of novelty-familiarity levels on material reward preference of first-grade children. *The Journal of Genetic Psychology*, 128(2):291–297, 1976.

Charles Spence. Audiovisual multisensory integration. *Acoustical science and technology*, 28(2):61–70, 2007.

Charles Spence. Crossmodal correspondences: A tutorial review. *Attention, Perception, & Psychophysics*, 73(4):971–995, 2011.

Jonathan Steuer. Defining virtual reality: Dimensions determining telepresence. *Journal of communication*, 42(4):73–93, 1992.

Robert J Stone. Haptic feedback: A brief history from telepresence to virtual reality. In *International Workshop on Haptic Human-Computer Interaction*, pages 1–16. Springer, 2000.

Arlette Streri and Edouard Gentaz. Cross-modal recognition of shape from hand to eyes and handedness in human newborns. *Neuropsychologia*, 42(10):1365–1369, 2004.

Johann Taljaard. A review of multi-sensory technologies in a science, technology, engineering, arts and mathematics (steam) classroom. *Journal of learning Design*, 9(2):46–55, 2016.

Hong Z Tan, Nathaniel I Durlach, G Lee Beauregard, and Mandayam A Srinivasan. Manual discrimination of compliance using active pinch grasp: The roles of force and work cues. *Perception & psychophysics*, 57(4):495–510, 1995.

Fabio Tatti, Netta Gurari, and Gabriel Baud-Bovy. Static force rendering performance of two commercial haptic systems. In *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications*, pages 342–350. Springer, 2014.

Warren S Torgerson. Theory and methods of scaling. 1958.

Robert J van Beers, Daniel M Wolpert, and Patrick Haggard. When feeling is more important than seeing in sensorimotor adaptation. *Current biology*, 12(10): 834–837, 2002.

Paolo Viviani, Gabriel Baud-Bovy, and Marco Redolfi. Perceiving and tracking kinesthetic stimuli: Further evidence of motor–perceptual interactions. *Journal of Experimental Psychology: Human Perception and Performance*, 23(4):1232, 1997.

Erica Volta, Paolo Alborno, Monica Gori, Simone Ghisio, Stefano Piana, and Gualtiero Volpe. Enhancing children understanding of mathematics with multisensory technology. In *Proceedings of the 5th International Conference on Movement and Computing*, pages 1–4, 2018.

Jean Vroomen, Paul Bertelson, and Beatrice De Gelder. The ventriloquist effect does not depend on the direction of automatic visual attention. *Perception & psychophysics*, 63(4):651–659, 2001.

Laron Walker and Hong Z Tan. A perceptual study on haptic rendering of surface topography when both surface height and stiffness vary. In *12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS'04. Proceedings.*, pages 138–145. IEEE, 2004.

David H Warren, Robert B Welch, and Timothy J McCarthy. The role of visual-auditory "compellingness" in the ventriloquism effect: Implications for transitivity among the spatial senses. *Perception & Psychophysics*, 30(6):557–564, 1981.

Andrew B. Watson and Denis G. Pelli. Quest: A bayesian adaptive psychometric method. *Perception & Psychophysics*, 33(2):113–120, 1983.

Robert B Welch and David H Warren. Immediate perceptual response to intersensory discrepancy. *Psychological bulletin*, 88(3):638, 1980.

Sam L Witryol and S Stavros Valenti. A developmental comparison of novelty-familiarity levels in first-and fifth-grade children. *The Journal of Genetic Psychology*, 136(2):281–284, 1980.

Matthew Wright. Open sound control: an enabling technology for musical networking. *Organised Sound*, 10(3):193–200, 2005.

Robert Wright. Virtual reality. *The Sciences*, 27(6):8–10, 1987.

Qinghua Wu and Jin-Kao Hao. A review on algorithms for maximum clique problems. *European Journal of Operational Research*, 242(3):693–709, 2015.

Takashi Yoshioka, Sliman J Bensmaia, Jim C Craig, and Steven S Hsiao. Texture perception through direct and indirect touch: An analysis of perceptual space for tactile textures in two modes of exploration. *Somatosensory & motor research*, 24 (1-2):53–70, 2007.

Anis Zarrad. Game engine solutions. In *Simulation and Gaming*, pages 75–87. BoD–Books on Demand, 2018.

Craig B. Zilles and John Kenneth Salisbury. A constraint-based god-object method for haptic display. *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, 3:146–151 vol.3, 1994.