IEEE Transactions on Instrumentation & Measurement, Vol: 57(9) pp: 1840-1851

AdHapticA: Adaptive Haptic Application Framework

Mauricio Orozco[1] Rosa Iglesias[2] and Abdulmotaleb El Saddik[1]
[1]Multimedia Communications Research Laboratory (MCRLAB)
School of Information Technology and Engineering (SITE)
University of Ottawa, ON, Canada
[2]Ikerlan Research Centre, Spain
E-mail : {morozco, abed}@mcrlab.uottawa.ca, RIglesias@ikerlan.es

Abstract
Research performed in the area of haptics has produced some remarkable results with a variety of haptic devices. However, haptic-based applications are designed to consider only a particular haptic device. Therefore, the functionality of a haptic-based system is limited by the chosen device's features, such as workspace, device's inertia, friction, number of points of interaction, number of degrees of freedom and maximum exertable force. In short, these haptic-based systems are limited for use with a certain haptic device. Without a doubt, the need for a software tool to adapt existing haptic-based systems to the capabilities of another haptic device is evident. On the other hand, one of the main advantages of using haptic devices is the possibility of saving data during the haptic interaction. Our proposed framework, called AdHapticA, deals with both issues: it automatically adapts a haptic-based system to be used with another haptic device and saves the corresponding haptic data for quantitative evaluation.
The AdHapticA framework can be used to study the feasibility of certain haptic devices to meet the requirements of an application. A case study is presented to evaluate single-point interaction and hand exoskeleton haptic devices for authentication purposes by using the same virtual scenario. The applicability of the proposed framework is shown and the results obtained from the haptic data are captured when a particular application is performed with either single point (desktop device) or multi-point interaction devices (hand exoskeleton). Therefore the results have shown that current hand exoskeleton devices are less suitable for tasks that require a certain level of precision like haptic-biometric based tasks.

1. Introduction

From the linguistic point of view, the word haptic is derived from the Greek verb "hapthestai" meaning "to touch". Haptics refers to the expanding discipline that is associated with the study of the sense of touch through human computer interaction. The science of haptics has received enormous attention in the last decade [25]. Applications in different disciplines such as robotics, computer graphics, cognitive science, psychophysics and neuroscience have contributed to its development. As a result, vendors have thrived on creating haptic-related products to be deployed commercially for research and development purposes, as well as serving as prototypes. A haptic-related product refers to the combination of a haptic device and the Software Development Kit (SDK) or Application Programming Interface (API). In the last decade, such haptic-related products were considered expensive products and only a few vendors commercialize them; however, these days the haptic technology is becoming widespread and it has fostered interest in new providers [1-5]. Consequently, new haptic devices have come forth, along with well-known and already-used haptic devices, such as, the PHANToM devices for single-point interaction from Sensable Technologies Inc. [6] and the CyberForce system for multiple-point interaction or the hand exoskeleton device from the Immersion Corporation [7].

In pursuit of making haptic technology more applicable on a daily basis, several applications have been proposed and designed based on specific or available haptic devices. And, in parallel, many haptic devices have been developed for use in particular applications. Therefore, haptic research has focused on designing and/or evaluating several sensory-based prototypes of different characteristics and capabilities for use in multiple applications. Applications of this technology have been rapidly spread to various devices applied to Graphical User Interfaces (GUI's), games, multimedia publishing, scientific discovery and visualization, arts, editing sound, editing images, the vehicle industry, engineering, manufacturing, tele-robotics, tele-operation, education, training, as well as medical simulation and rehabilitation. As Virtual Environments (VEs) are strongly linked to haptic-based applications, most of these applications make use of those computer-simulated environments. Haptic-based virtual environment applications are considered in this research.

As a consequence, the adaptation of existing haptic-based applications to new innovative devices is desirable, for instance, to evaluate different features and capabilities of other haptic devices for use in a specific application. However, the lack of a standard format for APIs poses a great challenge. The main goal of the Adaptive Haptic Application framework (*AdHapticA*) is to easily adapt a haptic device to an existing haptic-based system. This should be similar to when a mouse is connected to a computer and the system provides different or additional capabilities (i.e. hyper-fast scroll wheel or precision laser tracking) according to the mouse's features. Generally, adding new haptic devices to existing haptic-based systems relies on reusing existing software components, readapting them to new functionalities, and customizing the virtual scene. As we will discuss later in the paper, *AdHapticA* is a framework with mechanisms that deal with translating the requirements of a haptic-based application into a virtual environment regardless of which haptic device or API is being used. *AdHapticA* enables the user to switch between different APIs (multiple API switching) allowing the user to harness the abilities of different haptic APIs.

The 'Plug and Play' feature of the *AdHapticA* system allows the addition of a new haptic device without requiring code implementation. This is desirable for current researchers or academics - non developers or programmers - whose research makes use of the haptic technology. Generally, a variety of haptic devices (from different providers) can be found in research laboratories., With the *AdHapticA* system, an application based on a specific haptic device could be used with another device while the other one is not available. Another example is for neuroscientists who attempt to carry out several experiments with patients and they want to see which device will fulfill the interaction demands for a successful performance. On the other hand, this framework attempts to provide users with an easy-to-use tool for designing and building new visual-haptic scenes. Furthermore, the *AdHapticA* system allows researchers to capture and save haptic data during users' interaction for posterior measurement analysis. These data are captured depending on the device features, for instance, position, force, torque, orientation or angles of the user's fingers.

Through this approach we envision a system that can be flexible and adaptive to the current haptic technologies already available. To achieve it, *AdHapticA* considers the concept of designing and adapting haptic-based applications according to the current haptic and software technology, in the same pipeline. This objective is achieved by creating a layered system which comprises four components: the Application Factory (AF), the Software/Hardware Adaptation component (SHA), the Intelligent Agent Engine (IAE), and the Behavioral Data Repository (BDR). This framework can also be seen as prototype research in looking for a common framework/standard for haptic systems in terms of hardware and software development.

Additionally, a case study, which motivated the development of the *AdHapticA* system, is reported. From an existing haptic-biometric system for individuals' authentication, which is used with the Reachin system [8] – combining the Desktop PHANToM device and stereo viewing -, our attempt was to study the suitability of the same system with a hand exoskeleton device, the CyberForce system [7]. The results show the applicability of the proposed system, as well as the unsuitability of current hand exoskeleton devices to be used for individual authentication. The rest of the paper is organized as follows: Section 2 gives an overview of the state of the art of the frameworks proposed to support multiple haptic devices in a virtual environment. Section 3 describes our proposed architecture and Section 4 presents a case study where a particular hapto-based application is selected to be used with two different haptics devices. Section 5 presents the testing protocol design, which serves to make a decision related to the case study. Section 6 shows the results obtained by supporting two different devices on the same application (case study). Finally, section 7 describes the conclusions reached from the proposed architecture. A good range of haptic devices exist that possess the potential to offer users a rich experience in a virtual reality environment. Thus, we envisioned a system that could be flexible and adaptive to the current haptic technologies already available. This implies integrating many fields such as software engineering models, software development kits, application programming interfaces, and graphic and haptic rendering techniques, which characterize the proposed architecture at different levels.

2. Related work

The idea of having a framework that facilitates the development of existing haptic applications based on a specific device to support multiple haptic devices in a virtual environment is not trivial and some researchers have attempted to achieve it [8] [6] [9] [10]. The reason is that each haptic device supports a specific API that includes specific graphic libraries, haptic rendering algorithms, compatible collision detection algorithms, and, in some cases, limited support for general graphical tools. Some research has been developed with the idea of integrating different haptic audio visual VEs and hardware technologies - haptic devices - into one common framework.

One of the pioneers in building an API, which supports two devices, was Reachin Technologies. These devices were the Desktop Phantom [6] and the Delta [1]. Through the same API, programmers could implement new applications to be used with only those devices. Later on, after the commercial success of the Phantom devices, SensAble Technologies Inc. launched the OpenHaptics SDK to be used with their own devices (i.e. Premium, Desktop and Omni) [6].

In 2004, El-Far and his colleagues introduced a concept called the UNISON framework to standardize the development of haptic-visual applications [9]. Their work was focused on providing a set of services regardless of the choice of graphic or haptic software and hardware for an e-commerce application. Such a framework lacks the adaptability to support different haptic devices in multiple haptic visual applications, not only for the e-commerce application.

Recently, architecture has been designed with the idea of easily and efficiently developing software with haptic capabilities [11]. The proposed architecture in [11] imports an existing API to provide software developers with a large set of pre-coded capabilities in .NET. This architecture has been oriented as a development tool without considering the relationship between the capabilities of an API and a VE. However an open source haptics project called CHAI 3D is the closest technology to meet the fundamental requirements of our proposed work [10]. CAHI 3D framework supports several commercial single-point interaction haptic devices. The CHAI 3D haptic/graphic libraries enable the rapid development of device-independent multimodal virtual worlds and provide a good start for building a generic framework to enable users to work in the same environment with at least two different haptic devices. CHAI 3D wraps up different APIs into a more generic API those allowing different devices to be used. When these different APIs are wrapped, some functionalities are lost [11] *AdHapticA* on the other hand, strives to eliminate the lost of functionalities at the low implementation level by allowing at a very high level, users to choose the exact haptic API implementation to use with the device.

Thus, none of the proposed frameworks explicitly details the capabilities of interaction depending on the chosen haptic device. In addition, the issue of adapting a haptic-based application from a single-point interaction (as found by any PHANToM haptic device) to a multiple-point interaction (i.e. the CyberForce system) is not known. The *AdHapticA* system addresses this challenge of supporting single and multiple-point interaction. Moreover, it allows users to store haptic data according to the haptic device used for posterior quantitative analysis.

3. Adaptive Haptic Application Framework: AdHapticA

In some cases, the limitations of a haptic device (i.e. workspace, exertable force, points of interaction and so on) and the available supporting software platform restrict the benefits of an application. For instance, the possibility of using a hand exoskeleton device instead of a single-point interaction device is a clear example. The usage of a hand exoskeleton device could enhance user interaction and provide researchers with greater information content for better evaluation. The proposed system aims at supporting the creation of haptic audio visual environments that can be used with different haptic devices according to the user's needs or availability. When designing a framework like *AdHapticA* in the haptic domain, customizing current technology is an evident challenge that needs to be tackled, especially for the great variety of haptic devices and their related APIs. For example in the framework design, extensible and adaptive features are needed for the usage of different haptic APIs and the management of their behavior.

Designing a framework with white-box characteristics enables programmers to write code for extending or refining the applications, built on top of it, to accomplish the requirements of a particular application [12]. Therefore, the *AdHapticA* framework is built with this principle by defining interfaces or interface classes, in each of its components that are needed to achieve the feature of adaptability. Furthermore, for predefined haptic devices and existing haptic-based applications, the *AdHapticA* framework provides a 'plug-and-play' feature, where the system automatically adapts the haptic device. Through the evident need to merge the current haptic technology in a single pipeline and create flexibility in order to be adaptive to the user's needs, we envision a system conceptually visualized in Figure 1.

As depicted in Figure 1, our objective can be achieved by creating an architecture with four main components: Application Factory (AF), the Software/Hardware Adaptation component (SHA), the Intelligent Agent Engine (IAE), and the Behavioral Data Repository (BDR). The fundamental characteristic of the AF component is to facilitate the creation of new haptic-visual applications. Current applications such as navigating a virtual maze, signing a cheque or lifting a cup are also included. Users can utilize such applications or modify them by adding new virtual objects. The SHA component is in charge of detecting the type of haptic and graphic libraries for the specific connected device. Therefore, the IAE component is needed to accommodate the haptic device API with software implementations according to the application and the connected haptic device. Important tasks in this component include, activating or enabling the methods integrated with the haptic API selected and its relationship with the functions to graphically and haptically modulate the object's properties, such as the stiffness parameter. In addition, the goal of the BDR component is to record, store and retrieve haptic data generated during users' interactions. In the following subsections, we illustrate the fundamental features of each component in the proposed framework.

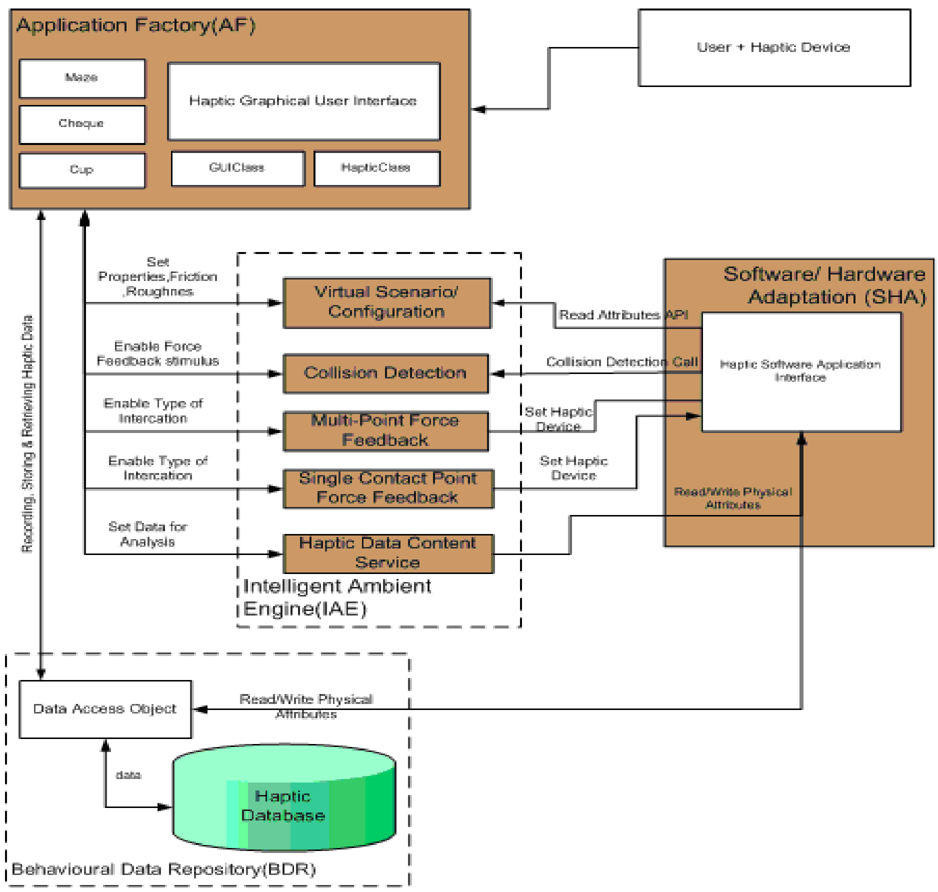


Figure 1. Overall architecture of the Adaptive Haptic Application framework (*AdHapticA*).

3.1 Application Factory

Initially, we envisioned a set of applications that address interesting fields of research such as rehabilitation procedures, authentication and verification of hand movements, and gaming performance, among others. Such applications can represent a research test-bed in areas related to neuroscience, neurophysiology, pattern recognition, and haptic perception. We have developed the following applications: navigating a maze, signing a cheque, and lifting a cup. The first two applications were created to be used with the Reachin system but also with Open Haptic from the PHANToM family, whereas, the cup and maze application makes use of the CyberForce system. The maze and cheque applications have been used for authentication purposes [13] and the task of lifting a cup has been used for rehabilitation of stroke patients [14].

The Application Factory (AF) component sets up a graphical display based on a Windows environment by using the OpenGL library. Currently, the AF allows the user to set the application by loading models from different formats and mapping the haptic requirements by using a defined menu, which is activated according to the device in use. The AF implements an interface that defines methods and contains variables that any subclass would have to define and implement to add haptic functionality to the objects rendered in the graphical scene. Moreover, to make the framework extensible, this component supports different file formats for rendering the graphical scene. The 3D StudioMax is one of the most widely used software for creating 3D virtual scenes, thus a 3DSLoader is provided to load and render 3D StudioMax (3DS) models [15]. 3DS models can be created in user-friendly commercial modeling systems; therefore, the object is not required to be modeled in the OpenGL model format (the majority of the APIs are based on the OpenGL format). The OpenGL format is also supported.

3.2 Software and Hardware Adaptation

3.2.1 Software

Currently, each haptic device uses a different API and thus when programming a haptic-based virtual environment, users have to create a virtual scene that is dependent on the API and whose haptic interaction depends on the haptic device' features. When creating a virtual scene, the user can add haptic properties to the created objects (i.e. stiffness, friction, texture) according to the device. Therefore, the application is limited to what the API and/or the haptic device are capable of. Due to this fact, existing applications cannot be easily adapted to a different device. Therefore, it is evident that there is a need for a component that maps device-API's capabilities and incorporates them into haptic-based applications. The solution is to have a software adaptation component to wrap haptic API functionalities based on the haptic device activated for the current application.

Based on the above, we defined a Software/Hardware Adaptation (SHA) interface that has a main goal of wrapping multiple haptic APIs, such as the OpenHaptics API and the Reachin API, to provide uniform access to these APIs. The haptic libraries have corresponding sets of functions and classes. To organize the entire collection of APIs into one package, we created abstract classes for each type of device (OpenHaptics, Reachin, etc.) and then derived concrete subclasses of each of those classes for each supported device. In this case, it is critically important to ensure that the created device objects are those for the desired API. An abstract HapticFactory class defines methods to create an instance of each abstract class that represents functionality in the device. Eventually, there will be concrete applications factories, which are concrete subclasses of the abstract factory. These 'sub-factories' will implement specific methods to create instances of the concrete device classes for the specific API.

CHAI 3D is the only development software that supports a variety of different haptic devices not pertaining to the same family (Table 1). It allows developers to build haptic-based applications for a predefined set of single-point interaction haptic devices. To our knowledge, there is no framework that supports both multiple-point and single-point haptic devices, so far. As a first step, the *AdHapticA* framework addresses this issue.

TABLE 1- APIs and haptic devices supported

Support Device	CHAI 3D	VirtualHand	Reachin	GHOST	OpenHaptics
Phantom Omni	Yes	No	No	No	Yes
Phantom Desktop	Yes	No	Yes	Yes	Yes
Phantom Premium	Yes	No	No	Yes	Yes
CyberForce	No	Yes	No	No	No
Other	Yes	No	No	No	No

3.2.2 Hardware

The need for a mechanism that enables the application's functionality according to the connected haptic device is clear. Different haptic devices have different messaging policies to be detected by the current operating system of the computer. The SHA component is able to recognize the device that has been connected and pass the information to the IAE component, which enables the requirements needed for that particular haptic device. For example, in the case of the CyberGrasp exoskeleton, the SHA configures the system to call the third party applications to calibrate the device before the graph scene can display the proxy representation of such a device in the VE. The hardware element of the SHA component is defined as an abstract level. It has been implemented in software to link the physical hardware specifications of the haptic devices with its corresponding APIs. In other words, this component enables high level programming commands to establish communication with low level programming of hardware. Thus, the hardware component not only permits the set up of the environment characteristics for the plugged haptic devices plugged, it also allows to the haptic applications to gain access to hardware information so that they can use such resources to set properties of the virtual environment such as stiffness, friction, texture etc, which are related to the device type. The hardware component can be seen as a small kernel between AF and the resources from the haptic devices and from the computer itself such as memory and I/O devices.

It is worth mentioning the differences between single and multiple point interaction from the viewpoint of the commercially available haptic devices. A desktop haptic interface such as any PHANToM device has an end-effector in the form of a stylus, which operates in a physical 3D environment. This device can capture the 3D position, the amount of force and torque applied, and the current orientation of the stylus. The API is capable of providing force feedback resistance and friction when the real stylus comes in contact with a virtual solid object. The same type of single-point interaction can be found in devices such as the Omega or Delta [1] or the MPB 6S Freedom [4]. On the other hand, as an example of multi-point interaction force feedback devices, the CyberForce system is the only one commercially available (see Section 4.3.1 for further details).

3.3 Intelligent Ambient Engine

There is an evident need to model different virtual scenarios and those to be adaptive to different individual's purposes without the need of a changing code. The haptic-based application is adapted according to haptic device specifications, similar to when a mouse is connected to a computer and the features of scrolling or tracking are enabled. The context-aware or IAE component provides the user with the appropriate virtual scene and all the requirements for haptic interaction through the connected haptic device. Generally, the virtual scene is scaled according to the haptic device's workspace and the haptic interaction is enabled. With the relevant software and hardware, it will be possible to offer the user an optimum experience of a haptic-induced virtual world. This IAE component has been built to be easily updated, extended and modified. The user interacts with the whole framework via a *GUIClass* class from the AF component that prompts the user for high level requirements (such as the interaction type/device, the virtual environment components, data recording, etc.), and generates a set of rules or specifications for the application. The IAE goes beyond traditional interfaces and is meant to personalize and adapt automatically to particular user behavior patterns or interests (user profile), and different contexts (application by means of Ambient Intelligence techniques [12]. The IAE component, shown in Figure 2, is envisioned to receive application requirements from the AF component and a persona user profile and apply intelligent algorithms to generate concrete and low level requirements that are used by the SHA component.

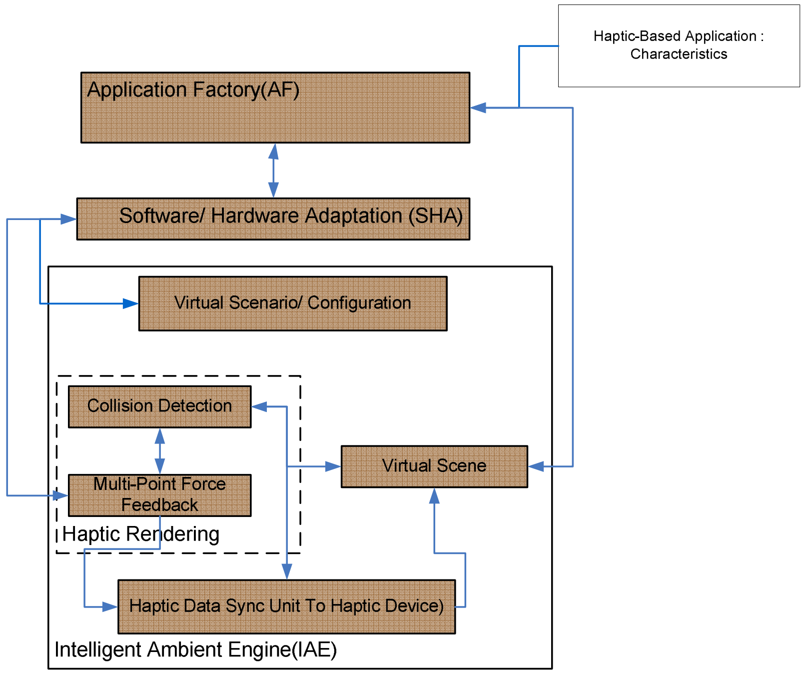
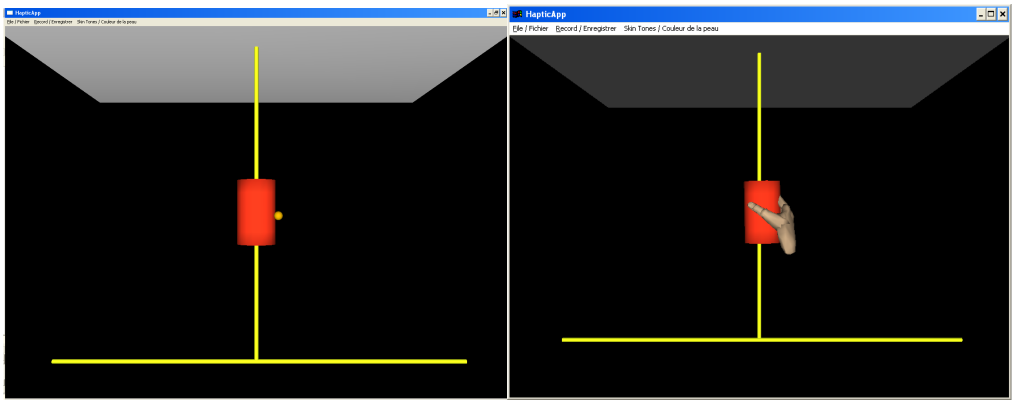


Figure 2. Intelligent Ambient Engine component (IAE) and its relationships with other components.

The IAE component makes the separation of the haptic and graphic rendering process by managing the haptic rendering and virtual scene functions via abstract classes *Virtual Scene* and *Haptic Rendering*. IAE manages the haptic rendering process as an API dependent element by instanced functions implemented in the SHA component. Meanwhile, the graphical and application functionalities remain as a single entity. The fundamental contribution of the IAE component is to map the haptic API's functionality in an approach defined by associating generic names to the functions that could be generalized for most haptic APIs. Functions such as rendering haptics, or changing the proxy colour (virtual representation of the haptic device), would all have an associated representation in the policy of the IAE component stated in the Virtual Scenario/Configuration interface. For example, haptic applications based on the exoskeleton device, which use the Virtual Hand Toolkit [7], has a function to change the skin colour of the hand represented in the VE. Thus, in a single point interaction device such as the PHANToM device, its API [6] also defines a function to change the colour of a proxy instead of the hand. Based on this principle, the IAE creates an interface in an object-oriented fashion, which enables it to call upon those functions that are implemented by specific APIs by their associated generic name. The functions that are not available in the API implementation would still be called, but not implemented.

The integration of the haptic and graphic rendering process with the application under different API umbrellas is required to ensure that the functions that handle the haptic calculations are matched with the functions implemented in the IAE interface. In addition, the view of the environment is handled by the IAE component by calling a function to scale the models or objects that appear in the graphic scene to a defined size. Furthermore, IAE is able to perform troubleshooting of the mismatches in the haptics and graphic scenes. In other words, the scenario when a user moves the device **stylus** up and down, the **proxy** on the screen moves in and out. This means that the graphical coordinate system and the haptic device coordinate system do not match. Thus, IEA defines a function that adjusts both coordinate systems by rotating along the x, y or z axis to match the presence of the concurrent coordinate systems.



(a) (b)

Figure 3. Screenshot of the haptic application 'Lifting a cup' using the Omni haptic device (a) and the CyberForce system (b)

Thus, the IAE component requires different services or user requirements to handle haptic properties of virtual objects according to the application selected (i.e. a virtual environment with deformable, static, dynamic virtual objects). Based on such requirements for the virtual scene, the IAE component creates the software engineering models for designing the application. Figure 3 shows the application of the virtual cup (a) for the PHANToM Omni device and (b) for the CyberForce system. This application was originally created for the CyberForce system, and it was adapted by using *AdHapticA*. With the Omni device, once the user touches the cup, it is attached to the displayed virtual end-effector and the user can lift the cup and feel the weight. In the case of the CyberForce system, the user is able to lift the cup by grasping it. As it is shown in the case study, an authentication function was implemented in this component to verify a user or reject impostors based on the haptic interaction.

3.4 Behavioral Data Repository

Everyday people interact with different devices, such as checking e-mail messages,, driving a car, or using mobile phones. These devices have become a part of our daily environments. It is probable that almost everybody has a unique way of opening a door or typing a message using a keyboard [6] [17]. Intrinsically, we create historical data about our performance through interactions with daily devices that is never recorded. Recording and storing a user's data when interacting with haptic systems are the primary objectives of the BDR component.

The use of the data recorded with the current set of applications has potential relevance to discover hand movements that can be classified as predefined user patterns or for rehabilitation. In our proposed system, the data is formed by a set of arrays containing the output attributes obtained from the haptic event. Initially, the data source is generated as flat files and then it is modified to represent an object-oriented database. The data files recorded from the implemented applications register a variety of physical attributes according to the haptic device used. In the case of a single-point interaction, this information is the 3D world coordinates of the stylus's position (x, y, z), force exerted, and torque applied from the haptic-based application with the associated devices. In the case of a multi-point interaction, other parameters can be captured that are associated with the VirtualHand SDK (from the CyberForce system), such as the bending angle of each finger during grasping or holding a virtual object.

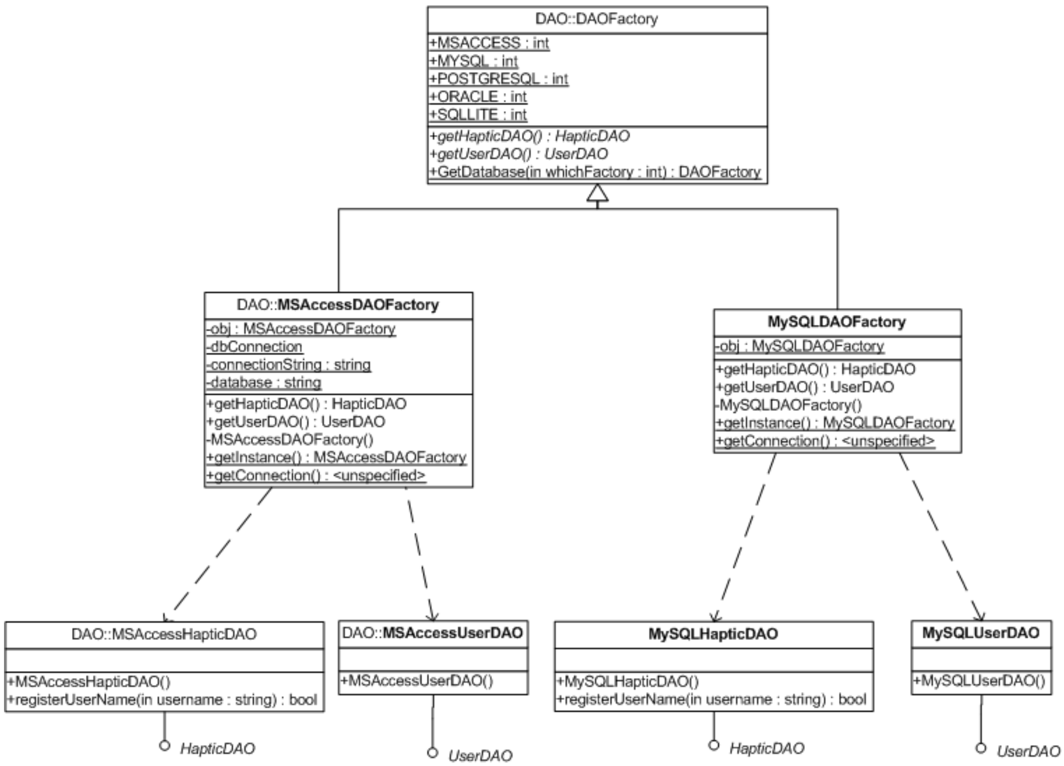


Figure 4. Class diagram of the Behavioural Data Repository Subsystem that enables recording, storing and retrieving of haptic data.

Within this component, it is defined as a Data Access Object Factory (DAOF) interface, which is the abstraction and encapsulation of all accesses to the data source (Figure 4). The DAOF manages the connection with the data source to obtain and store data. The DAOF implements the access mechanism required to work with the data source via either MySQL or MS ACCESS, although other database toolkits (i.e. Oracle, PostGreSQL, SQLlite) exist.

Table 2 shows an example of an array of data that could be captured from a PHANToM haptic device. It is understood that the physical attributes captured from haptic interfaces vary from device to device and its related software API.

TABLE 2 - Example of a recorded file from the haptic system

Trial	Timestamp (ms)	Position X	Position Y	...	Force X (N)
1	0	0.23344	0.56768		0.00456
1	…	…	…		…
1	0.01230	0.37676	0.98989		0.03767

4. Case Study: single-point or hand exoskeleton devices for authentication

The *AdHapticA* system emerges as a consequence of considering the possibility of the CyberForce system for individual authentication. From an existing haptic-biometric system for authentication, which is used with the Reachin system [8] – combining the Desktop PHANToM device and stereo viewing -, our attempt was to study the suitability of the same system with a hand exoskeleton device, the CyberForce system [7].

4.1 Motivation

Without a doubt, the hand is the most functional and varied instrument that human beings use to interact with the real world. Many issues have been raised in the field of computer science with the introduction of the use of haptic interfaces to mimic the real world into a VE. The PHANToM Desktop device is the most widely used interface for the haptic community. It provides high precision, low friction and low inertia as main features. It has been successfully implemented in a variety of art and design applications, such as sculptural modeling for digital content creation and fine arts [6] and interactive haptic painting with a 3D virtual brush concept called Dab [16]. Likewise, the PHANToM Desktop device has been used in systems that involve an identity authentication process with successful performance [17].

On the other hand, the use of the whole-hand as input in a VE has gained much interest. Hand exoskeleton haptic devices can provide a multi-point interaction where tasks such as grasping and holding objects in VEs can be characterized in a more natural way. The gesture recognition discipline has provided the reference point for the introduction of CyberGlove technology as an input device. Studies in gesture recognition have provided a quantitative representation of the level of sensitivity that such gloves can provide. Kessler et al. have investigated the level of accuracy in recognizing fingers' joint angles under different flexion measurements by using the CyberGlove device [18]. They have also concluded that noise and hand size do not affect the accuracy level. However, calibration affects performance in accurately recognizing the fingers' joint angles involved in a particular gesture. Recently, Chen et al. have demonstrated that the measurements with the CyberGlove device with a dynamic gesture approach are quite precise [19]. The recognition success rate of their scheme using the CyberGlove has reached 90% accuracy on given gestures.

In 1999, in the field of dynamic signature verification, Tolba presented the first attempt to use a virtual reality glove as an input device to capture behavioral data from users performing their hand signature [20]. The acquisition data from the glove was based on the optical-fiber sensor situated at each finger, which captured 256 different positions. Results from Tolba showed 0% error rate with 100% confidence by combining 21 correlated features with a 6 x10 matrix.

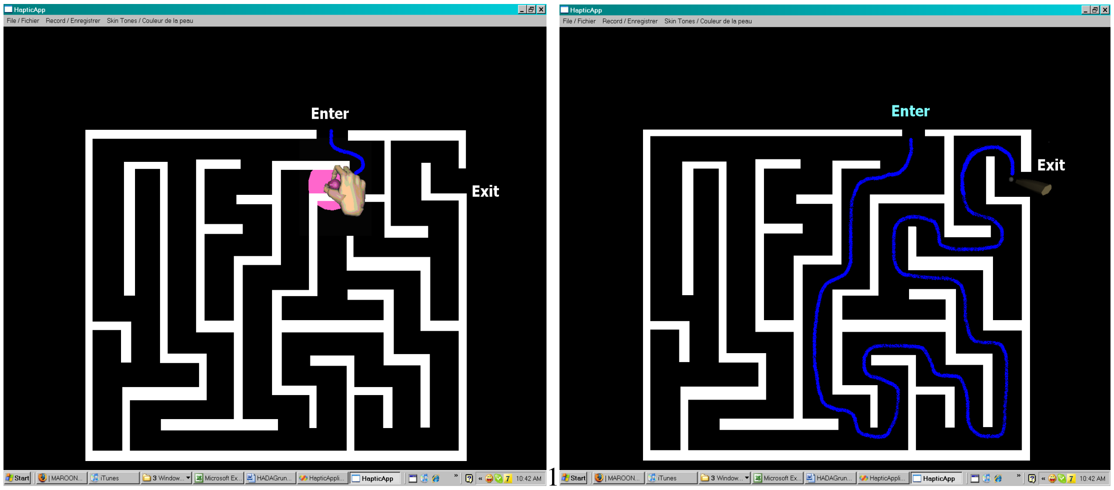
Based on the previous research, both the PHANToM Desktop device and the CyberForce system, which involve the use of the CyberGlove, seem to be suitable to be used for authentication purposes as biometric input. A fundamental assumption is that there is a possible quantification of the human motor skills through the haptic data generated while a user is haptically interacting within a VE. Such haptic data can be used as biometric identifiers if there are natural differences between the psychomotor patterns exhibited by individuals in virtual domains and if the data can describe uniqueness for each individual. Thus, the haptic data can generate haptic profiles that can be used for further analysis (i.e. individual authentication).

4.2 Experimental Design

The *AdHapticA* system allows the same application to be adapted according to the haptic device connected into the system. In our case, the task of navigating a maze was chosen to be used with the Reachin system with the Desktop device and the CyberForce system. This application consists of a 3D maze, which was first created to be used with the Desktop device (Figure 5 b). The virtual end-effector corresponds to the haptic device's end-effector that allows users to navigate the maze. When the CyberForce device is connected, the virtual scene is only comprised of the maze and the virtual hand, therefore a virtual wand is needed to navigate the maze (Figure 5 a). The collisions between the maze's wall and the wand are counted and provide the user with a realistic feeling of navigating through a maze. The performed path is visually represented using a line path (colored in blue) throughout the maze.

There was only one correct path to exit the maze. The ability to solve the maze was not judged, but the way in which participants navigated the maze was. While navigating the maze, the *AdHapticA* system recorded data related to participants' hand motions by using the DAOF interface within the BDR component. Therefore, such data were used as a means of testing individual abilities or for describing psychomotor patterns, among other features. In our case, the stored data was processed using MS ACCESS.

The user must move the stylus or grasp the virtual wand from the entry ("Enter") to the Exit arrow without crossing walls using the single point interaction device or the multiple-point interaction unit.



(a) (b)

Figure 5. The color codes of the Maze Recording Process (a) A user interacting within the maze with the CyberForce system (b) A user interacting with the PHANToM Desktop device

The haptic-biometric experimental set consisted of 32 different participants divided into two groups, A and B at different times. Prior to the task, they were introduced to basic concepts of haptic interaction and they were familiarized with the application (to avoid undesired results). Group A was integrated by 22 participants to handle the haptic-based task with the PHANToM Desktop device and the Reachin system and 10 participants formed Group B who used the CyberForce system. Each participant carried out the experiment 10 times on different days.

4.3 Haptic Devices

The *AdHapticA* system adapted the virtual maze application to be used with the CyberForce system. It wrapped the API for the Desktop device by using a defined function in the SHA component. In that way, the possibility of interaction with the CyberForce system was enabled. The VirtualHand SDK is used by the CyberForce system [7] and the OpenHaptics API for the Desktop device [6]. The framework mapped the functions between both APIs to provide the equivalent virtual scene and haptic interaction according to the device. A haptic class was implemented for each API, making sure to provide similar haptic functionality.

4.3.1 Hand Exoskeleton Device: the CyberForce system

The CyberForce system consists of three pieces of hardware: the CyberGlove, the CyberGrasp and the CyberForce armature. The CyberGlove collects data that is related to the hand, such as the bending of the joints of each finger. The data collected is used to display a realistic representation of the hand on screen. The device provides up to 22 high-accuracy joint-angle measurements. The sensors on the hand provide a resolution of about 0.5 degrees (typical standard deviation between glove wearing). The CyberGrasp device is an exoskeleton capable of generating force feedback on the fingers. The CyberGrasp can give the user a feeling of actually grasping an object or each finger tapping on an object. The device can generate forces that are roughly perpendicular to the fingertips. The force on each finger can be specified individually. The CyberForce armature has two functions, one of which is tracking the hand movement. The CyberForce armature provides six degrees-of-freedom movement and is capable of measuring hand rotations and translations. The other function of the CyberForce is to generate force feedback that simulates inertia. For example, users can feel objects' weight or collisions with walls. The CyberGlove is attached to the CyberForce armature and a CyberGlove is worn in conjunction with the CyberGrasp.

4.3.2 PHANToM Desktop Haptic Device

Single-point interaction haptic devices, such as the PHANTOM Desktop, provides a positional resolution ranging from 450 dpi (dots per inch) to up to more than 1000 dpi (~0.02mm). Generally, the PHANToM haptic interface presents high-precision positional characteristics among others [6] [21]. It also offers a range of motion for hand movements that pivots at the wrist. Some PHANToM devices provide six degree-of-freedom positional sensing through digital encoders. They deliver 3 degrees-of-freedom positional sensing and 3 degrees-of-freedom force feedback capabilities such as the Premium models 1.0, 1.5, and 3.0.

5. Evaluation Criterion

5.1 Authentication

After performing the experiments, the raw values measured from different participants completing the maze were analyzed to search feature similarities among different trials of the same person (intra-personal data) and feature differences among different people (inter-personal). The precision of a biometric system is reliant on the choice of the features used to form biometric profiles [17] [18]. For accurate authentication, the features must be chosen so that they are unique for each individual. Authentication decisions are made by a variety of metric classifiers, for instance, measuring the Matching Score (MS) between two different profiles.

An authentication function implemented in the IAE component is called the Decision Domain Function (DDF), which accepts a user by establishing their identity, and rejects a user if this establishment cannot be formed. To identify a biometric profile, the MS between that profile and one of the template profiles must be less than or equal to an upper bound, τ. The performance of the authentication system varies with the choice of this upper bound. If τ is chosen to be large, many impostors will match with template profiles, also called the False Acceptance Rate (FAR); whereas, if τ is chosen to be small, many genuine users will fail to match with their template profile. The False Rejection Rate (FRR) refers to the latter case, where the system falsely rejects a genuine user (Figure 6).

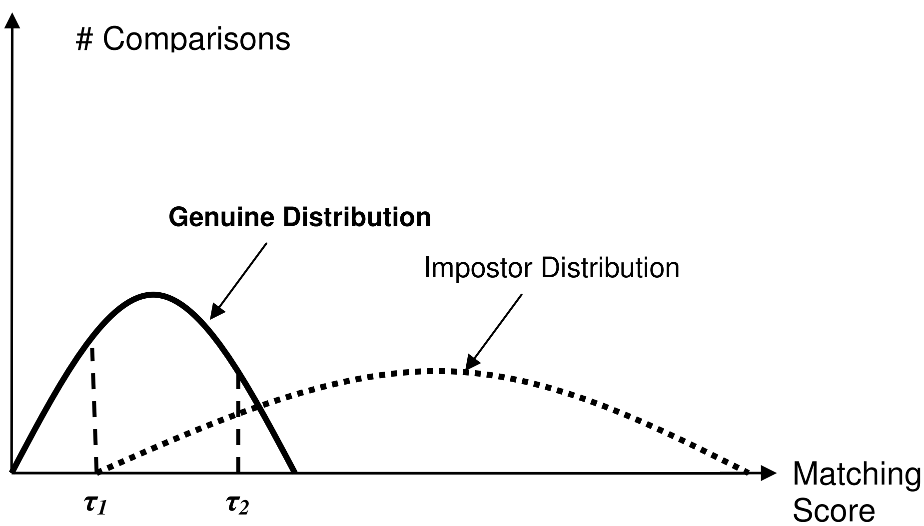


Figure 6. Matching Scores (MS) less than τ_2 would allow access to a significant portion of impostors; whereas, MS less than τ_1 would deny access to a significant portion of genuine users.

Raw signals are sent to the IAE component, which are subsequently processed to construct a user profile recognition template. Thus, the DDF operates in two modes: identification and verification. In the identification mode the DDF evaluates the system performance by comparing profiles of the same user to yield a "genuine" MS distribution. On the other hand, comparisons between dissimilar profiles yield an "imposter distribution". For different values of τ, the evaluation of the FAR is carried out by integrating the imposter distribution from zero to τ. In a similar way, the calculation of the FRR is carried out by integrating the genuine distribution form τ to infinity. The set of points of FRR (τ) and FAR (τ) form the ROC (Receiver Operating Curve).

The verification mode of DDF operates differently. Using the pre-stored template profiles, we determine a different threshold MS for each user. To calculate a specific threshold for a user, we first construct a genuine distribution based only upon comparisons of this user's template profiles. Then, we form an imposter distribution by comparing all dissimilar user profiles to this user's profiles. We construct several sets of threshold values where the FAR is constant amongst the users; hence, allowing the construction of the ROC.

5.2 Biometric Analysis

5.2.1 Single Point Interaction

This user's behavioral authentication is based on human manipulations of the PHANToM Desktop device [6]. The *AdHapticA* system uses position, velocity, force, and torque data, among other features, as biometric identifiers. The BDR component collected this haptic data according to the device connected while the participants were navigating the maze. The rate of sampling is around 0.015 seconds (70 samples/sec). The *AdHapticA* system characterizes the state of a haptic system by measurements of completion, time (t), velocity (v), force (f), torque (n), and angular orientation of the stylus (θ), to form a state vector $\boldsymbol{x} = (t, v_x, v_y, v_z, f_x, f_y, f_z, n_x, n_y, n_z, \theta)$, where the subscripts x, y, z indicate spatial dimensions. For example, v_x is the projection of the velocity vector onto the x axis. In order to evaluate the information content of specific features (which characterizes the uniqueness of the user), we consider state vectors of a reduced dimension. For instance, in the analysis of the information content of velocity data, we consider state vectors of the form $\mathbf{v} = (v_x, v_y, v_z)$.

5.2.2 Multiple Point Interaction

On the other hand, the state of a haptic system, when the CyberForce system is used, can be characterized by measurements of completion time of the task (t), 3D position (x, y, z), and angle of each of the fingers' proximal phalanges (θ). The proximal phalanges are the closest to the hand. As a result, *AdHapticA* defines the state of the haptic system of the CyberForce system based on 9-dimensional features with a vector of the form: $\boldsymbol{r} = (t, x, y, z, \theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$. These state vectors are sampled at a rate of about 20 samples/sec; however, we noticed that the rate of sampling is not a stable parameter. For instance, during the first second of one trial we recorded 150 samples in a second; whereas, later in another trial only 12 samples were recorded in a second. In addition to the vector $\boldsymbol{r}$, a post-processing function was designed into the DDF to analyze the following parameters, which characterize a set of sampled state vectors:

- Time to do the exercise
- $d\theta/dt$ for each finger (variation of a finger angle over time)
- Idle time on each finger (i.e. percent of the data where $d\theta/dt = 0$)
- Accumulated distance over each axis (x, y or z)
- Speed over each axis

Based on the features selected and the post-processing stage, the complete feature set resulted in a 57 dimensional feature vector that contains the following information:

- Mean, standard deviation, relative dispersion, maximum, minimum, and maximal difference of the $d\theta/dt$ distributions for each θ, and for the dx/dt, dy/dt, and dz/dt distributions
- Time taken to do the exercise and percent of this time where $d\theta/dt = 0$ for each θ (idle time)
- Accumulated distances for all of x, y, and z

6. Results: Haptic Attributes and System of Authentication

6.1 Single Point Interaction

"The concept of Dynamic Time Warping (DTW) or alignment has been applied to solve the problems related to data mining, gesture recognition, speech processing, manufacturing, and medicine" [23]. The DTW algorithm can efficiently find an alignment between two sequences, allowing a more sophisticated distance measurement to verify an individual. Thus, we use the DTW algorithm to produce a MS between two biometric profiles by computing the minimum cost of aligning two data sets of time-series measurements represented by the state vectors of the system. We use two-dimensional time-series position data to characterize the virtual maze-trajectory, and therefore to be used in the DTW calculations. The identity of a user is verified by the system if their sample trajectory produces a MS, less than a predetermined threshold, with at least two of the associated template trajectories. As depicted in Figure 7, using the DTW algorithm, 26% of the skilled forgeries (FAR) were accepted at an operation level of 93.8% of Probability of Verification (PV). The PV is equal to 1 – FRR.

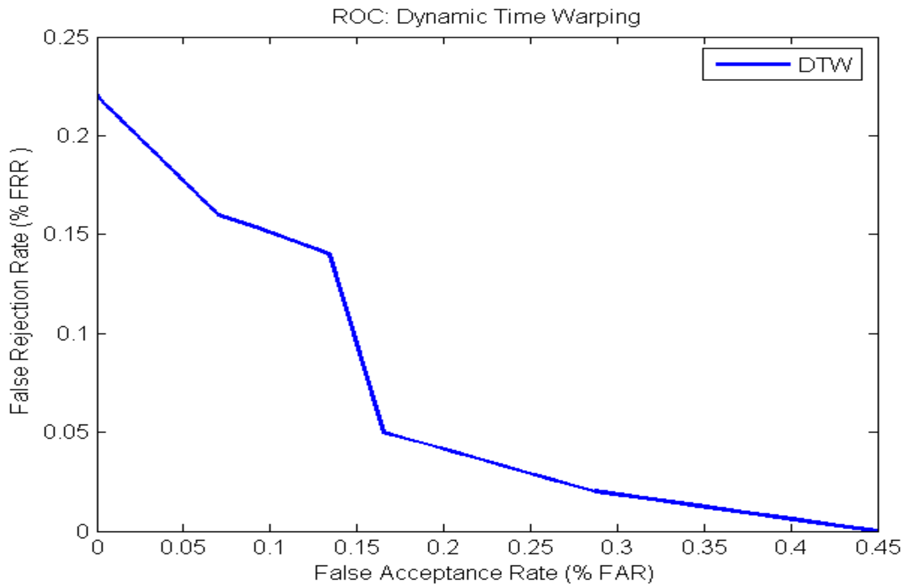


Figure 7. The Receiver Operating Curve (ROC) for the Dynamic Time Warping Method.

Based on the measurements, the results of our study suggest that the PHANToM Desktop device, like PHANToM, provides an accuracy level high enough to be suited for authentication purposes. Therefore, we can conclude that the use of these single-point interaction devices as a mechanism to authenticate individuals is feasible.

6.2 Multiple Point Interaction

On the other hand, from the biometric viewpoint, it seems that the haptic data captured from the CyberForce system are not accurate enough to be used as biometric identifiers. This conclusion was drawn from the results obtained from the statistical analysis of the 57-dimensional feature vectors from 10 different users. The *relative dispersion* of a statistical distribution, which is a good measure of their variance, was used. The relative dispersion is determined by the standard deviation and mean and it is also called the 'coefficient of variance'.

The average relative dispersion of all 57 dimensions of the feature vectors was 138.35%. This large percentage led us to immediately dismiss the possibility of using such haptic data for biometric purposes. Some parameters showed low relative dispersion values, amongst which were the total time mean of 22.54%, and the idle time mean of 6.4% (Table 3). Although the relative dispersion for theses parameters is low; all users had similar times. Therefore, this feature could not be used to identify them.

TABLE 3- Relative Dispersion of idle times with the CyberForce system

Proximal Angle	**Relative Dispersion of Idle Times**
Thumb	4.5 %
Index	7.0 %
Middle	6.2 %
Ring	7.7 %
Pinky	6.9 %

All users had almost exactly the same percentage of idle time: 95% for each finger. This suggests that either the users were very steady with their hands, or that the measurement processes were affected by the components of the haptic rendering loop implemented by the device itself [7] [24]. Table 4 shows the relative dispersion of features describing the distribution of $d\theta_1/dt$ (the change in the angle of User 1's proximal thumb phalange over time) for a particular user. The figures are high which immediately supports the observations mentioned.

TABLE 4 -Relative Dispersion of the thumb angle over time from User 1 with the CyberForce system

Feature ($d\theta_1/dt$)	Relative Dispersion
Average	17.9 %
Standard deviation	286.0 %
RD	22.8 %
Max	42.5 %
Min	22.5 %
(Max - Min)	320.2%

It seems that the overall performance, and therefore the captured haptic data, of the CyberForce system is affected by different issues. Firstly, the algorithm used for collision detection between the virtual hand object displayed in the virtual scene and, in general, virtual objects (in our case the virtual wand) need to be improved. These observations are supported by the optimized geometry approach selected for faster collision and haptic detection between virtual objects in the VE. Furthermore, the issue of the device calibration for each individual introduces an error in recognizing joint angles that affects the accuracy of the captured haptic data. Altogether, significant accumulative errors in the measurements of the grasping angles of each finger are introduced, which indirectly affects the performance of the authentication process.

7. Conclusions

One of our main goals is to decouple a virtual environment from the haptic device and API in order to support other haptic devices. To accomplish this goal, we have developed the *AdHapticA* framework based on the current haptic technology. It aims at adapting an existing haptic-based system to be used with other haptic devices (with different APIs). In this context, the haptic and graphic components have been separated so that the API can be changed while maintaining the virtual scene

We have also identified several issues of developing haptic-based applications in the same pipeline framework. Firstly, there is no standard for addressing the domain in terms of hardware and software application development. In addition, the difficulty of integrating new software methodologies into the current and new hardware prototypes is still increasing. With this adaptive and generalized approach, we have presented a proof that different haptic devices can be adaptable to the same application, therefore, some guidelines can be provided for abstracting the domain as well. Furthermore, this framework makes use of the data generated on the haptic-visual interactions. Such a source of data can be applied, for example, to an authentication system or to a follow-up rehabilitation procedure.

The selected case study provides a proof of concept for the *AdHapticA* framework, which is the use of different haptic devices for an existing application. The virtual maze application was initially built for the PHANToM Desktop device, and the same application was adapted to be used with the CyberForce system through the *AdHapticA* framework. The idea was to prove whether or not both haptic devices provide haptic data to be used as biometric identifiers for authentication purposes. The haptic data was saved while users were navigating the maze. The results suggest that single-point interaction haptic devices, such as the PHANToM Desktop device, have the potential to be used for individual authentication while interacting with a haptic system. However, the results with the CyberForce system were not promising. The experiment carried out with this device suggests that this technology is a bit far from being used on tasks that require a certain level of precision, such as in biometrics. The relative dispersion distribution of the haptic data generated was very high, which immediately dismisses the possibility of using this data for biometric purposes. This conclusion has been reached on the basis that the haptic data measurements of the fingers' phalanges during a haptic-grasping session were not reliable. Such measurements are affected by the calibration of the system at every session. In addition, in terms of non-accurate measurements, the software limitations are from the collision detection algorithm implemented by the VirtualHand SDK.

Currently, we are validating and extending our framework to incorporate new applications, new APIs and different methodologies for analysis within the IAE component. This will demonstrate that our framework is capable of handling multiple APIs and, at the same time, it takes advantage of the capabilities of different commercial haptic devices. Hopefully, a standard format for defining haptic APIs will soon be reached.

Acknowledgements

This work was partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by the Alexander von Humboldt Foundation. The authors would like to thank Rosa Iglesias for her constructive feedback.

References

[1] Force Dimension: http://www.forcedimension.com/
[2] Moog FCS Ltd: http://www.fcs-cs.com/index.html
[3] Haption: http://www.haption.com/
[4] MPB Technologies: http://www.mpb-technologies.ca/
[5] Novint: http://www.novint.com/
[6] SensAble Technologies Inc.: http:// www.sensable.com
[7] Immersion Corporation: http://www.immersion.com/3d/
[8] Reachin Technologies: http://www.reachin.se
[9] N.R.El-Far, X. Shen, and N.D.Georganas, "Applying Unison, a Generic Framework for Hapto-Visual Application Development, to an E-Commerce Application", Proc. IEEE Workshop on Haptic Audio Visual Environments and their Applications, Ottawa, Canada, October 2004.
[10] F. Conti, F. Barbagli , D. Morris , C. Sewell "CHAI: An Open-Source Library for the Rapid Development of Haptic Scenes" Demo paper presented at IEEE World Haptics, Pisa, Italy, March 2005
[11] Z. Mahboubi and S. Clarke, ".NET API Wrapping for Existing C++ Haptic APIs" IEEE International Workshop on Haptic Audio Visual Environments and their Applications, pp. 67-71, 2006.
[12] C. Szyperski, "Component Software – Beyond Object-Oriented Programming" Second Edition, Addison-Wesley and ACM Press, ISBN 0-201-74572-0, 2002
[13] M.Orozco, Y.Asfaw, A.Adler, S.Shimohammandi and A. El Saddik. "Automatic Identification of Participants in Haptic Systems" in Proceedings of IEEE IMTC 2005- Instrumentation and Measurements Technology Conference Ottawa, Ontario, Canada, 17-19 May 2005
[14] A. Alamri, R. Iglesias, M. Eid, A. El Saddik, S. Shirmohammadi and E. Lemaire "Haptic-based Exercises for Post-Stroke Patient Rehabilitation", IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2007.
[15] 3D studio max: www.autodesk.com/3dsma.
[16] W. Baxter, V. Scheib, M. Lin, and D. Manocha. "DAB: Interactive Haptic Painting with 3D Virtual Brushes". In Proceedings of ACM SIGGRAPH 01, August 2001, pp. 461-468.
[17] M. Orozco, I. Shakra, A. El Saddik Haptic: The New Biometrics-embedded Media to Recognizing and Quantifying Human Patterns. ACM Multimedia 2005 Conference, Singapore 6-11 November 2005
[18] G. D. Kessler, L.F. Hodges, Neff Walker, Evaluation of the CyberGlove as a whole-hand input device. ACM Transactions on Computer-Human Interaction (TOCHI), Volume 2 Issue 4
[19] Q. Chen, A.M. Rahman, A. El-Sawah, X. Shen, A. El Saddik, N.D. Georganas, "Access Learning Objects in Virtual Environment by Hand Gestures and Voice", Proc. Of the 3rd annual e-learning conference on Intelligent Interactive Learning Object Repositories (I2LOR 2006), Montreal, Quebec, Canada, November 2006
[20] A. S. Tolba, " GloveSignature: A Virtual-Reality-Based System for Dynamic Signature Verification," Digital Signal Processing Academic Press, vol. 9, pp. 241-266, 1999
[21] MC Çavusoglu, D Feygin, F Tendick "A critical study of the mechanical and electrical properties of the PHANToM TM haptic interface and Improvements for High-Performance Control. Presence, Vol. 11, No. 6 December 2002, 555-558.
[22] J. E. Keogh and M. J. Pazzani, "Derivative Dynamic Time Warping," Department of Information and Computer Science University of California, Irvine California 2000.
[23] B. Mirtich, "V-Clip: Fast and Robust Polyhedral Collision Detection," ACM Transactions on Graphics, Volume 17, No. 3, page(s) 177-208, July 1998
[24] T. Qu, A, El Saddik, A. Adler. *Dynamic Signature Verification System Using Stroke Based Features.* MSc. Thesis, University of Ottawa, Ottawa, ON, 2004.

[25] M. Eid, M. Orozco and A. El Saddik, "A Guided Tour in Haptic Audio Visual Environment and Applications", Int. J. of Advanced Media and Communication, v1 (3), pp: 265 – 297, 2007

Mauricio Orozco received the Dipl.Ing. degree in Electrical Engineering from the Technological Institute of Morelia, Mexico, in 1992 and the MSc. Degree in Software Engineering from the Applied Mathematics and Computing group from the School of Engineering at Cranfield University, UK in 2002. He also received the PhD. in Computer Science from the University of Ottawa in 2007. He worked for (CFE) Federal Commission of Electricity, Mexico as electrical engineer from 1993 to 1999. His current research interests are Multimedia Communications, Computer Graphics, Pattern Recognition, Biometrics, Haptics and Virtual Environments.

Rosa Iglesias is currently a researcher at Ikerlan, a technological research center in Spain. After graduating in Mathematics from the University of the Basque Country, she obtained a PhD in Computer Science from the University of the Basque Country in Spain. Her PhD was developed on the subject of networked haptic virtual environments for assembly tasks at Labein (Spain) and it was partly carried out during a visiting stay at MIT and Queen's University Belfast. She also hold a post-doctoral position at SITE, University of Ottawa. She obtained the second Thesis Award by the Research Basque association in 2007

Her research interests span networked haptic virtual environments, haptic applications on different fields, such as, education, industry or medicine, and ambient intelligence. In the latter area, her current research lines include multimedia "anytime/anywhere", ambient assisted living, agent technologies and smart homes environments. She is currently involved in several European projects on Ambient Intelligence including SmartHealth, SmartTouch and AmIE. She is a member of IEEE and ACM and of the editorial board of the International Journal of Advanced Media and Communication (IJAMC). Moreover, she has been serving on several technical program and organizing committees. She can be contacted via email at: riglesias@ikerlan.es. http://www.ikerlan.es

Abdulmotaleb El Saddik (F'IEEE), University Research Chair and Associate Professor, SITE, University of Ottawa and recipient of the Friedrich Wilhelm-Bessel Research Award from Germany's Alexander von Humboldt Foundation (2007), the Premier's Research Excellence Award (PREA 2004), and the National Capital Institute of Telecommunications (NCIT) New Professorship Incentive Award (2004). He is the director of the Multimedia Communications Research Laboratory (MCRLab) and the ICT Cluster of the Ontario Network for e-Commerce (ORNEC). He is a theme Co-leader in the LORNET NSERC Research Network. He is Associate Editor of the ACM Transactions on Multimedia Computing, Communications and Applications (ACM TOMCCAP) and Guest Editor for several IEEE Transactions and Journals. He is leading researcher in haptics, service-oriented architectures, collaborative environments and ambient interactive media and communications. He has authored and co-authored two books and more than 180 publications. His research has been selected for the BEST Paper Award at the "Virtual Concepts 2006" and "IEEE COPS 2007". Dr. El Saddik is an IEEE Distinguished Lecturer.