Available online www.jsaer.com

***Journal of Scientific and Engineering Research*, 2018, 5(12):140-146**

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

Designing Haptic Interfaces with Depth Cameras and H3D Depth Mapping

George Kokkonis

Dept. of Applied Informatics, University of Macedonia, Thessaloniki, Greece

Abstract This paper analyses the main haptic and tactile interfaces that could be used for designing haptic applications. It makes a categorization based on their characteristics and possibilities. It presents the most well knowv haptic applications. The software that could be used for haptic application development is described. The role of the Internet and the IoT in expansion of haptic applications is described. The Depth camera and their role in the creation of virtual scene development is analyzed. The transformation of a virtual 3D depth scene in a haptic application is described.

Keywords Haptic Interfaces, depth images, depthmap, Depth Video, Haptic, tactile systems, RGB-D camera

Introduction

Touch is one of the basic senses of the human body for the perception of the surrounding space and its understanding. Tactical experience is critical when the sense of presence or skillful handling of objects is required in a remote or virtual environment. However, the representation of the sensation of touch is difficult, since most of the human body is covered by the skin and each surface of the body responds to different stimuli, resulting in different sensitivity to a variety of different stimuli. Tactical interfaces allow human interaction with virtual or remote systems. They mainly concern feedback devices that are used to carry skin and muscle stimuli to the user. With this integration users can feel, push, pull, and manipulate objects in the virtual environment.

A key point of haptic systems is the way in which the operating mechanism is implemented to transform user movements to into equivalent actions of the tactile interfaces, and the way in which the tactile interface provides feedback to the user's functions and actions. The aim of all these is to increase human experience and understanding through enriched sensory simulation environments.

2. Categories of Haptic Interfaces

The tactile interfaces are divided into two categories regarding a) their portability and b) the type of feedback they will support. Haptic interfaces are divided into wearable and non-wearable devices. Another categorization that often occurs is according to the feedback that they produce and they are divided into haptic interfaces with tactile and kinesthetic feedback.

Tactile Interfaces support skin feedback, and gather and process information that is related to pressure, pain and heat on the user's skin. These interfaces are mainly applied to the fingertips. These devices often help users with vision problems. They can support tasks that requiring hand-eye coordination. They can alert user via vibrations and offer the possibility to explore elasticity and texture in virtual objects.

Interfaces that support kinesthetic feedback are used to gather and process information about user's movement and position. They can be applied to any part of the body, especially in the hand-arm, the ankle and the torso. These interfaces are characterized by two parameters: the Degree Of Freedom (DOF), which is the number of variables required for the complete determination and positioning of a device, and the Workspace, which is the physical space within which the endpoint of the haptic device moves. A device with a higher DOF has a larger Workspace. One of the advantages of the interfaces that support kinesthetic feedback is the spatial awareness, which is the determination of a person's position in the virtual environment. They also offer the determination of the position of the object, the virtualization of distance, and the determination and recognition of object dimensions. They, furthermore, offer accurate movement of objects in virtual environments and improved sense of tele-presence.

Before designing a haptic interface, some decisions have to be made, whether haptic feedback will be tactile or kinesthetic, whether the haptic device will be portable, which will be the perceptual area of the body, the number of the DOF, who will be the users of the application, and finally the size of the workspace.

However, in order to achieve the above goals, a set of hardware such as Joystick, stylus, exoskeleton devices, wearable vests, electrodes, must be developed. This hardware also includes robotic mechanisms, sensors and actuators.

3. The haptic Feedback

One of the first applications of haptic technology that was implemented and applied was to large aircraft, and is known as servo drive systems and involves the use of control surfaces. The first aircraft servo system didn't support feedback. External forces applied aerodynamically to the control surfaces were not perceptible to the user. Previously, in a lightweight aircraft without servo systems, there were vibrations to the pilot's control. That was a useful warning of a dangerous situation on the flight. This "vibration" is not noticeable when simple servo control systems are used. To feel the vibrations in a remote system a feedback system should be developed. The systems that involve a feedback response system and focus on the user's tactile or kinaesthetic feeling are called haptic systems.

The touch is often replicated by actuators that apply small pressure to the human skin. This is call "haptic" or "tactile" feedback. The motor provides mechanical pressure to the skin in response to electrical stimuli. In order to produce tactile feedback, electromagnetic motor vibrations are often used. Such vibrations are used also in cell phones. These electromagnetic motors provide strong tactile feedback, but their applications are limited only in skin conduct. Besides electromagnetic motors, electroactive polymer, piezoelectric and electrostatically activating surface are often used in tactile systems.

Haptic interfaces often used in teleoperations where they remotely control a robotic arm. The first haptic teleoperators were created in the 1950s at the Argonne National Laboratory by Raymond Goertz to handle radioactive substances remotely.

Simulation systems, such as training devices for pilots, have also integrated haptic technology. The haptic feedback enhances the feeling of tele-presences in virtual environments. Haptic simulators are often used in medical simulators for doctor training and flight simulators for pilot training.

Haptic technology is also used in video games, especially in racing video games and first person shooter games. In 1976, the motorcycle game of the "Sega moto-cross", and the "Fonz", were the first games that used Haptics. They caused a vibration on the steering wheel during a collision with another vehicle. The Tatsumi TX-1 introduced Force Feedback for car driving games in 1983.

3.1 The Haptic interfaces

In 2007, the "Novint Falcon" was released [1]. It was the first consumer 3D touch device to offer high-resolution 3D feedback, Figure 1. It produces the simulation of the haptic feeling applied to virtual 3D objects. The user can feel the texture and the physical properties of virtual objects in virtual reality.

Figure 1: Novint Falcon Haptic Device

In 2008, Apple began to incorporate the "Touchpad " into the MacBook and the MacBook Pro for better functionality and feedback.

The tactile feedback has been used more and more by devices of our everyday life. Many mobile phone manufacturers include different types of tactile technology in their devices. In most cases, it has the form of vibration. The tactile feedback technology is also used in products for navigation and music systems.

Haptics are an essential part of virtual reality systems. Most of them use a “stylus”. The user interconnects to the virtual world via a stylus, giving a form of interaction. Haptic systems are also used for 3D modelling. They give artists a virtual experience of interacting with the virtual models.

4. Software for creation Haptic Applications

4.1 H3D API

H3Dapi is an open source platform for haptic software development. It uses OpenGL and X3D open standards in a single scene graph to handle both graphics and touch. H3dapi enables the integration of haptics and audio on 3D stereo displays. H3Dapi is primarily designed to support fast development of haptic applications. With X3D, C ++ and Python, H3dapi offers three ways to program haptic applications. H3dapi is written in C ++, and is designed to be extensible, ensuring that developers have the freedom and ways to adapt and add the necessary supporting structures or graphics features of H3dapi to their applications. H3dapi has been used to develop a diverse range of supporting structures and multimedia applications in various fields, including medicine and industrial imaging.

4.2 Key benefits of using H3DAPI

The benefits of using the H3dapi software are as follows:

- Supports a wide range of tactile/haptic devices.
- It runs on all popular platforms, including Windows, Linux and Mac OSX.
- H3dapi software allows rapid deployment of applications using haptic X3D and Python.
- Easily expanded using hapto-optical features using the C ++ programming language.
- Allows simultaneous execution of multiple haptic devices in the same scene even with devices from different manufacturers.
- Supports most 3D stereo imaging systems.
- Provides freedom of choice for renderers including H3D based on SensAble OpenHaptics, Chai3D.
- Comes with a series of built-in examples of haptic forces and tactile surfaces.

4.3 H3DAPI Architecture

H3dapi is consisted by layers that integrate graphics and haptics into a platform where it implements and develops X3D and Python applications. The combination of tactile and graphic on a scene graph ensures that the software developer does not need to be troubled by the expansion task nor with the tactile or graphical systems, but could focus on developing the application he is developing.

One of the most important features of H3dapi is that it provides full flexibility to the project user beyond X3D and Python scripting. H3dapi is designed to be easily expandable for both advanced developers and low-level users who can make tactile graphics using the C ++ programming language. H3dapi uses OpenGL for rendering graphics and Hapi for rendering haptic structures. Hapi is the touch engine that provides touch capabilities for creating H3DAPI. H3DAPI is written in C ++ language and is designed to be extremely modular. HAPI consists of four main parts, each of which can be easily extended to meet the individual needs of users:

1) Allows you to handle different haptic HAPI devices.

2) Touch geometry base, consists of three subsections: collision handling, tactile performance algorithm, and interaction surface algorithm

3) Free touchspace, refers to a collection of power effects

4) The handling of issues.

The four tactile algorithms make, to a point where they are directed and based on the 3DOF touch device, where it is applied to HAPI: SensAble OpenHaptics, Chai3D, and two H3D in point and sphere-based situations. This means that H3DAPI users are able to choose from any of the supporting structures including even their own

written algorithm performance. HAPI presents the user the ability to implement the 6DOF touch algorithm in writing.
One of the main advantages of H3DAPI, is the use of the ISO standard, X3D with an open graphical design template. This results in consistency with most graphics software and is able to be saved in X3D format. This allows users to easily add existing supporting graphics structures. Even for inexperienced, developing touch with H3DAPI is not particularly difficult. An experienced developer using X3D will find that H3D programming is not much different than X3D programming. The H3D code is stored in XML files with the X3D extension.
In a H3Dapi code, the Python language is additionally used on X3D to express more complex interactions and movements on the scene.

4.5 OpenHaptics Toolkit - Sensable

OpenHaptics 3.0 software development simplifies and dramatically accelerates the touch of IT applications. It is designed to operate with the Sensable's tactile devices. QuickHaptics is an api, which allows every haptic developers to use C ++ language and add quickly and easily kinaesthetic and tactile feeling to virtual 3D objects.

Figure 2: Sensable's Phantom Haptic Device

The OpenHaptic toolbox includes:
• The Phantom Device Guide (IHF)
• QuickHaptics micro api
• Haptic Device api (HDAPI)
• The Haptic API (HLAPI)
• Utilities
• Examples of source code.

The QuickHaptic micro api creates a simple graphical application with haptic properties. It produces the same outcome with the use of eight lines of touch programming instead of 300 lines. The QuickHaptic allows fast program design and development. It is ideal for adding haptic properties to existing Virtual applications. Figure 3 ilustrates the layers of the QuickHaptics.

Figure 3: Layers of the QuickHaptics

5. The Criteria for Effective Haptic Interfaces

The criteria for designing an effective haptic interface device are three [2]:

1. "Free space must feel free". When a users moves to a virtual free space, he/she should not feel any force of feedback. This means that the haptic interface should be balanced in weight. The inertia and the back-drive friction of the haptic device should be a low as possible.
2. "Solid virtual objects must feel stiff". The maximum obtainable stiffness of virtual objects depends on the sampling rate of the haptic device, on the resolution of its sensors and actuators, and the servo rate. A user in a virtual environment can be convinced that a virtual surface is solid if its stiffness is above 20 Nt/cm [3]. The servo rate of the haptic device should be higher than 1 KHz.
3. "Virtual constraints must not be saturated". Virtual walls should be felt solid. The maximum force for a human finger can apply is 40 Nt [3]. During precise manipulation people often exert forces lower to 10 Nt. Most haptic forces are lower than 1 Nt.

Mean opinion score (MOS) evaluation based on virtual experiments should be made in order to determine the affordable values of the above criteria. The above criteria should be satisfied simultaneously. Improving the specification for one criterion should not degrade the specifications for the other two.

If the haptic interface is interconnected with another haptic interface or virtual objects via a network, the QoS requirements for maximizing the Quality of Experience of the user should meet the constrains of Table I [4].

Table 1: QoS for Interconnected Haptic Interfaces.

QOS	Haptics	Video	Audio	Graphics
Jitter (ms)	≤ 2	≤ 30	≤ 30	≤ 30
Delay (ms)	≤ 50	≤ 400	≤ 150	≤ 100-300
Packet Loss (%)	≤ 10	≤ 1	≤ 1	≤ 10
Update Rate (Hz)	≥ 1000	≥ 30	≥ 50	≥ 30
Packet Size (bytes)	64-128	≤ MTU	160-320	192-5000
Throughput (kbps)	512-1024	25000 - 40000	64-128	45-1200

A protocol for transferring medical and haptic data for real time application has already proposed in [5]. Another critical factor that should not be ignored is the system security. In [6] there is a complete description of the system architecture that a haptic application should fulfil in order to safely transmit haptic data.

6. Transforming Depth Video to Haptic applications

A fast and efficient way to transform real world objects to virtual haptic objects is with the help of 3D depth cameras. The 3D-depth cameras use the modulated infrared light in order to measure the distance between the camera and the physical object. The time needed for the light to travel form the 3D –depth camera and return back is called "time of flight". As the speed of light is constant and known, the distance between the camera and the physical object can be exported via the "time of light". The first low-cost, widely used 3D –Depth sensor is the Microsoft Kinect sensor that was used in the Microsoft Xbox 360 [7]. It provides a 640 X 480 pixel depth video at 30 fps with a 11-bit depth resolution at each pixel. Its depth precision begins from 1 cm at a distance of 2 m and ends up to 10 cm at a distance of 6 m between the camera and the physical object. Microsoft Kinect contains three sensors, an RGB color VGA video camera, a depth sensor, and a multi-array microphone. The depth sensor contains a monochrome CMOS sensor and an infrared projector. The infrared projector emits infrared light and the CMOS sensor captures the projection of the reflected infrared light. Figure 3 depicts 3 images produced by Microsoft Kinect 3d-depth sensor.

Figure 3: 3D-depth images from Microsoft Kinect v1 sensor. On the left a simple jpeg image, in the middle a depth image and on the right a 3d-depth image

The above 3d-Depth images can be imported to the H3D software and create a haptic depth-map video. As H3D uses XML files for giving haptic properties to virtual objects, the stiffness, the static friction, the dynamic friction, the mass and the maximum difference in the z-axis can easily be set. An example of an XML H3D file is given below.

```
<DepthMapSurface stiffness="0.6"
maxDepth="0.01"
staticFriction="0.7"
dynamicFriction="0.7"
whiteIsOut="true" >
<ImageTexture containerField="depthMap" url="depthmap_image.png" repeatS="false" repeatT="false"/>
</DepthMapSurface>
```

The available range for the stiffness and friction is *[0,1]*, where the zero 0 value correspond to the minimum value of the haptic interface and 1 the maximum value that the haptic interface can enforce. The *maxDepth* field indicates maximum difference between black and white colour surfaces. The *whiteIsOut* field indicated that the white coloured surfaces are "higher" than the black coloured surfaces. For the haptic rendering the “GodObjectRenderer” of the H3D program should be used. The haptic interfaces that could be used by the user are all the haptic interfaces that are supported by the H3D as it uses the “GodObjectRenderer” for haptic rendering.

The Kinect sensor can be used to create a 3D virtual object, or a virtual scene of a real world object, by using the algorithm of KinectFusion [8]. KinectFusion enables Kinect sensor to instantly create detailed 3D reconstructions of real world. The 3D pose system uses the depth data from the Kinect sensor to track x,y,z position of the user [9]. A representation of a virtual scene of an office is shown in figure 4.

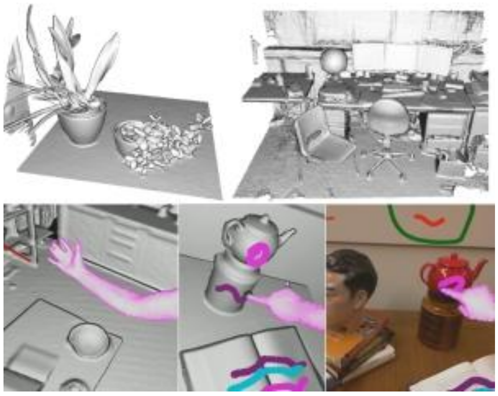

Figure 4: 3D Virtual Representation of an office using Microsoft Kinect sensor and the algorithm of KinectFusion [10]

These frames of the virtual 3D scene are then inserted to H3D and with the help of DepthMap haptic properties can be applied to the virtual objects.

8. Conclusions

This paper introduces the Haptic technology. It is divided into two categories, concerning their portability and the kind of the haptic feedback that they support. Different devices that can be used to experience the haptic applications are analysed. The most known software development kits for haptic applications are presented and analysed. The role of 3D –depth cameras in creation of virtual haptic scenes is presented. The framework for crating haptic scenes with 3D-Depth cameras and the H3D software development kit is illustrated.

References

[1]. N. Karbasizadeh, A. Aflakiyan, M. Zarei, M. T. Masouleh and A. Kalhor. Dynamic identification of the Novint Falcon Haptic device. 4th International Conference on Robotics and Mechatronics (ICROM), Tehran. 2016, pp. 518-523.

[2]. Thomas H. Massie, and J. Kenneth Salisbury. The phantom haptic interface: A device for probing virtual objects. Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems. 1994, Vol. 1, 55.
[3]. P.H. Sutter, J.C. Iatridis and N.V. Thakor. Response to Reflected -Force Feedback to Fingers in Teleoperations. Proc. Of the NASA Conf. On Space Telerobotics. NASA JPL, 1989.
[4]. G. Kokkonis, K. Psannis, M. Roumeliotis, , & S. Kontogiannis. A survey of transport protocols for haptic applications. IEEE 16th Panhellenic Conference on Informatics, PCI2012. 2012, 192-197.
[5]. D. Tomtsis, S. Kontogiannis, G. Kokkonis, I. Kazanidis, & S. Valsamidis. Proposed cloud infrastructure of wearable and ubiquitous medical services. Digital Information Processing and Communications (ICDIPC). 2015.
[6]. C. Stergiou, K. E. Psannis, A. P. Plageras, G. Kokkonis, & Y. Ishibashi. Architecture for security monitoring in IoT environments. IEEE 26th International Symposium in Industrial Electronics. 2017.
[7]. Zhang, Z. Microsoft kinect sensor and its effect. IEEE Multimedia. 2012, Vol. 19(2), 4-10.
[8]. Izadi, Shahram and Kim, David and Hilliges, Otmar and Molyneaux, David and etc. KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology. 2011.
[9]. Ben Glocker, Shahram Izadi, Jamie Shotton, Antonio Criminisi. Real-Time RGB-D Camera Relocalization. International Symposium on Mixed and Augmented Reality (ISMAR) 2013. IEEE, 2013.
[10]. Microsoft Kinect. KinectFusion Project Page. [Online] https://www.microsoft.com/en-us/research/project/kinectfusion-project-page/.

Authors

George Kokkonis is a scientific staff at the Dept. of Business Administration of Western Macedonia University of Applied Sciences, Greece. He received his PhD from Department of Applied Informatics, University of Macedonia, Greece. He received a five-year Eng. Diploma from the Dept. of Electrical and Computer Engineering, Aristotle University of Thessaloniki and his MSc in Information Systems from the University of Macedonia. He has been working as a Lecturer for the last 12 years at the Dept. of Computer Applications in Management and Economics, Western Macedonia University of Applied Sciences, Grevena, Greece. He has been teaching the Laboratory Lessons of Multimedia Systems, Pc Programming and Data Bases. For the last 5 years is responsible of the NOC of the branch of the TEI of West Macedonia in the city of Grevena, Greece. His research interests include haptic protocols and interfaces, teleoperation and human–computer interaction.