Haptic Interface to DI-Guy

Lei Wei, Wael Abdelrahman, James Mullins, Mohammed Hossny, Mick Fielding, Saeid Nahavandi

Centre for Intelligent Systems Research, Deakin University, Australia

{lei.wei | wael.abdelrahman | james.mullins | mohammed.hossny | mick.fielding | saeid.nahavandi} @deakin.edu.au

Abstract. Much of the state-of-the-art commercial simulation software focuses on providing realistic animations and convincing artificial intelligence to avatars within a scenario. Research on enhancing the interactivity and immersion of such simulation is not generally as highly refined. Based on the research of our user-study, we identify that it is desirable for simulation software to have improved interaction between users and the scenario. The more intuitive interactions that the system has, the better the immersive experience will be. Based on this idea, we propose a pipeline to effectively integrate haptic interactions into the DI-Guy simulation environment, with the goal of improving user interactivity with the avatars in the scenario. By implementing such a pipeline, simulation packages will be capable of not only enhancing control over certain actions of avatars, but also providing realistic force feedback to the user. Unlike typical haptic integration approach which is based on high-level libraries and existing haptic collision detection and force rendering algorithms, we implement our own novel algorithm based on the fact we do not have direct access to low-level primitives such as vertices and normals in the virtual environment. Besides, we also implemented our own vendor-independent middleware as a part of our algorithm to provide a generic solution for integration of natural interfaces such as haptics and 3D space mouse with commercial software to enable the building of a complete immersive virtual training system.

1. INTRODUCTION

Computer-based simulation has always played a crucial role in contemporary research and industry projects [13]. State-of-the-art commercial simulation software is widely used across various industry sectors and is continuously expanding. However, the focus of research advances in simulation is currently mostly revolving around providing more visually appealing animations and comprehensive artificial intelligence (AI) of virtual avatars [12]. Although these aspects are indeed very important, there are also some other fundamental aspects to further improve the immersion of simulation software. For example, latest graphical processing units (GPUs) and central processing units (CPUs) could achieve high quality interactive visual rendering of simulation software (e.g. ray tracing [19]) which used to be extremely difficult to achieve before. On the other hand, the ways users control and interact with virtual avatars are still restricted to mostly peripheral devices such as mouse and keyboard.

The development of a virtual training system is important for many domains such as education, manufacturing and military ones. The system needs to be intuitively interactive and based on advanced VR environments to be successful and usable. Therefore, it is important to have low level software and hardware integrated system. By low level, we mean as close as possible to native integration between the system components. Figure 1 shows an example standalone virtual training system developed by CISR with intuitive interface such as multi-point haptic device and 3D mouse.

The unintuitive way of interaction is one major aspect that prevents simulation from further advances [10]. Considering this issue, several approaches have been proposed to provide a more intuitive and interactive way of interacting with simulation software. The most popular approach is to integrate haptic devices into the software and support for complex and immersive interaction [22]. However, integrating haptics into existing simulation packages does impose more challenges. Compared with visual rendering that has been developing for many years and dominant in most simulation software packages, a haptic pipeline is still rather exotic.

It is usually not possible neither advisable to reinvent the wheel and develop the whole software from scratch in order to provide native support of haptics. Therefore, an extension of existing software with accessible programming interfaces and new threads are more commonly adopted as a compromise. On the other hand, force feedback (haptic) rendering requires a very high frame rate of 1000Hz per single point of interaction. This means that a haptic pipeline must run much faster (8-33 times) than a typical visual pipeline, which takes a large amount of CPU time for real time processing. The compromised extension architecture for a haptic pipeline is not as efficient as native code and therefore makes the situation even worse.

From the analysis above, it is very clear that the problem of unintuitive interactions in current simulation software packages need to be solved and further improved to provide better usability. Integrating haptics is feasible but still requires further effort to help achieve the goal of intuitive interactions in simulation. In this paper, our aim is to solve this problem by proposing a general purpose force-enabled interaction framework which efficiently incorporates haptic devices from different vendors into off-the-shelf simulation environments such as DI-Guy. The main contribution of this paper is the integration methodology of natural interfaces such as haptic ones with already established virtual environments, while maintaining the real-time experience.

Figure 1: CISR standalone training system

The rest of the paper is organized as follows. In Section 2 we review related work on simulation software packages and haptic research. In Section 3, we introduce our proposed general haptic interfacing to commercial simulation software packages. Section 4 focuses on implementation and technical details of the proposed framework. In Section 5 we discuss a real life application to show the advantage and capability of the proposed idea. Finally, in Section 6 we summarize the work done and highlight future potential advancements along this research.

2. LITERATURE REVIEW

A virtual reality (VR) environment is a platform where real and/or imaginary worlds can be simulated [11, 12]. The simulation domains are variable and range from entertainment to military training [13].The rise of VR or computer simulated environments goes back to the middle of the last century [1]. Since then one of the main motivations to build such systems was to capture as much reality as possible. The systems started first as passive platforms and then evolved into interactive ones [2]. The interactions evolved in parallel with personal computers (PC) input devices advanced from being wired and a mechanical unintuitive device to remote device-less interaction [3]. The virtual systems are also multi-modal in a manner that it can support audio, vision, and haptics [4,14]. In [5], Edward analyzed the role of touch in enhancing the sense of presence in virtual scenes and found out that the sense of touch is as important as other visual and auditory senses.

The computer haptics introduction to VR systems is relatively new compared to the long history of the VR systems themselves [6]. This is due to many factors:

1. The evolution of the haptic hardware which is still lacking intuitiveness such as rendering forces at only a single point of feedback [15].
2. The ease of generic integration of different haptic devices into already developed VR systems.
3. The speed requirement of the haptic loop of 1000Hz per haptic point compared to only 30-120Hz for graphics only applications [7, 16].

There are many different models of haptic devices [20]. Each haptic device has its own software driver and specification. The integration of these haptic interfaces to the existing advanced VR systems is an important step towards the utilization of haptics in different applications. Typical haptic interaction concentrates on two main challenges; 1) Haptic collision detection [17]; and 2) Haptic rendering [18]. Typical haptic collision detection and force rendering are based on the fact that the haptic pipeline has prior knowledge of the scenario, such as vertices, normals and axis aligned bounding boxes. However, there are many cases (such as plugins to external software) in which such information is not available to the haptic pipeline. Traditional haptic collision detection and force rendering algorithms will not be able to handle such cases. This challenge is becoming more and more complicated as the prosperity of haptics and demand of integrating haptics into existing visualization and simulation software are increasing monotonically.

DI-Guy, Second Life [8] and others are examples of cutting edge virtual worlds that can provide a high level of realism and immersion. This paper demonstrates the integration of haptic interfaces with DI-Guy as an example of an immersive application in the simulation field. This enables an intuitive control of the human and objects avatars that improve the visualization and training experience.

DI-Guy is a commercial software package for human avatar visualization, simulation as well as corresponding intelligence and decisions. It enables users to create immersive scenarios with various avatars, decisions, interactions as well as external hardware and software through reserved plugin interfaces. A comprehensive library is also integrated, which supports actions like fleeing, following, fearing, crouching, down to the micro level such as, shaking head, waving, eating, gazing and facial expression and many other features. Interactions are defined by the so called "action beads" which are triggered by either predefined time stamps or external events and device signals. Complex direction and collateral reactions for in-scenario events are defined by the so called "decision bead" as well as "script bead". Although DI-Guy is powerful and comprehensive in simulation, modeling and artificial intelligence, it still lacks general support of external device input and output such as haptic device. This results in an inconvenience in using it for research and development.

The integration of haptics with VR environments were approached several times in the literature [26, 27]. However, the integration was usually done with simplified prototypes and experimental environments and not cutting-edge, commercial and closed-source software. In this paper we show how an advanced simulation system such as DI-Guy can be extended with a generic haptic interface.

3. PROPOSED FRAMEWORK

The system framework is composed of the following main components: a haptic device, DI-Guy software package, other input devices, and the two-way plugin that link these components. These three components are synchronized to be utilized by the user in a real-time manner. The system data flow is shown in Figure 2.

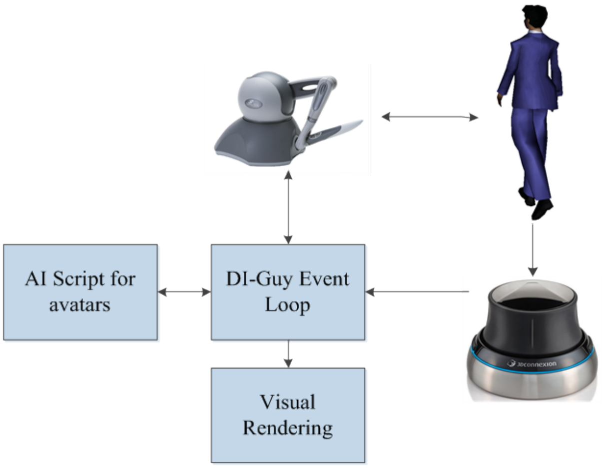


Figure 2: System framework

The end user is responsible of providing inputs (translation and rotation through the haptic stylus) and receiving outputs (force feedback rendered and displayed via the haptic device) from the system. However, the DI-Guy simulation software package has its own loop that process events and messages coming from different parties.

The haptic device is not only an input device but it is an input-output device that requires its own 1000Hz rendering loop to process events. The DI-Guy allows a plugin development to handle external hardware and software through a series of reserved interfaces. The plugin implementation targets the haptic devices handling which includes: feeding new positions, determining collisions with the environment objects, and calculating the force feed for the user. The plugin also supports events for the additional buttons that the haptic device might have. Other forms of input devices are also supported such as 3D mice and head mounted displays (HMDs) which removes the dependency on conventional input hardware such as keyboards.

Beside the manual interactions with human beings, the DI-Guy has a great feature of AI characters. These characters are available as singles or in crowds. The AI entity has a mind which is formed by a set of parameterized automata. It can perform actions and get affected by others actions including human events. These entities are a cornerstone of making interactive sessions within DI-Guy [21].

The main challenges in integrating the haptic interface with the DI-Guy system are the handling of the 1000Hz haptic loop and events, the collision detection and calculation of proper force feedback, and the synchronization with other events and entities.

4. TECHNICAL DETAILS AND APPLICATION

The pipeline was implemented as a plug-in to DI-Guy through some of its interfaces. We created a separate thread and used asynchronous scheduling to handle the input from the haptic device. We also built a wrapper layer on top of the vendor-specific haptic function calls and created our own general haptic function calls, giving a hardware agnostic approach that's capable of interfacing with a wide range of haptic devices from different vendors. This feature is very convenient for real-life application deployment. In addition, the proposed pipeline has the potential to support multiple nodes operating via a network of computers, allowing haptic multi-user collaboration within the virtual scenario. Figure 3 shows the general implementation framework.

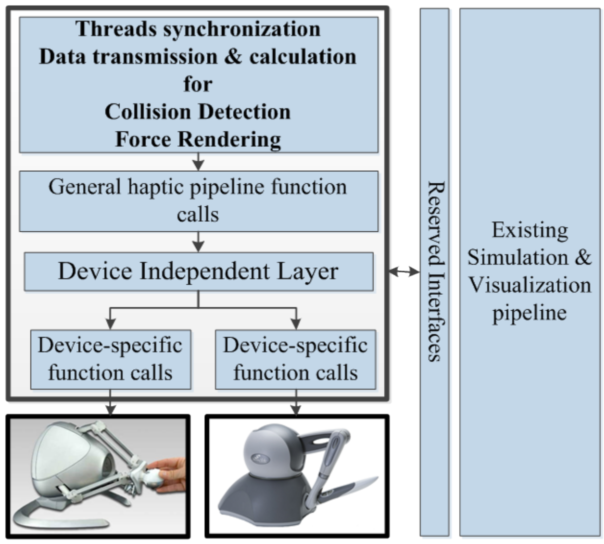


Figure 3: General Implementation framework of the haptic integration into DI-Guy

When integrating haptics into existing simulation and visualization software, there are generally three approaches:

1. Re-write the whole software from scratch and provide native haptic support.
2. Use high-level haptic libraries to enable haptics
3. Use low-level haptic libraries to enable haptics

The first approach is generally not applicable as most of the commercial simulation and visualization software has been developed for many years. Although this approach provides the best integration, it could only be done by the original developers.

The second approach is adopted in most existing haptic-enabled software. It provides easy and straightforward approach to integrate haptics at the cost of direct control of devices and limited functionalities. This drawback is more obvious with large software systems which consume very high processing ability for existing visual rendering and left not enough for the more demanding haptic pipeline.

The third approach is based on top of the most direct function calls close to hardware driver level. This feature enables most possible function calls and best efficiency of running haptic loop. However it is also harder to implement compared with the second approach.

In most cases, integrating haptic pipeline into existing visualization pipeline requires the prior knowledge of the low-level scene information, such as vertices and normals. However, in some cases these information are unavailable due to the nature of commercial software. They may release certain interfacing functions to report simple collision result of one line segment with the whole scenario but they seldom grant access to low-level primitives. In these cases, the traditional haptic collision detection and force rendering algorithms will simply not work.

The approach we propose is based on multiple haptic rays casting towards different directions from the current position of the haptic stylus. The rays are cast to try to collide with the scenario. The number and the length of the rays can be adaptively adjusted by the local complexity of the scene, as well as manually by the user.

Multiple haptic rays casting is based on using parametric functions in spherical coordinates u, v and R:

$$x = R * \cos(u)$$
$$y = R * \sin(u) * \cos(v)$$
$$z = R * \sin(u) * \sin(v)$$

A uniform sampling of u and v provides for an even distribution of the rays in the scene around HIP to the distance R (Figure 4).

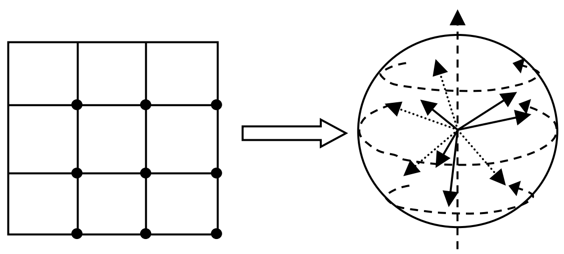


Figure 4: Multi-rays generated from equally divided parametric surface

5. CASE STUDY: OZBOT MOBILE PLATFORM

We built several test cases to validate the proposed idea and implementation work based on Deakin University's designed and developed OzBot system as shown in Figure 5.

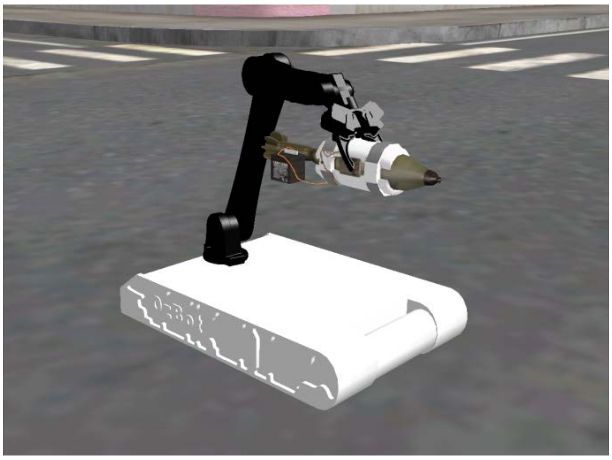
The OzBot robot is built around an inertial measurement unit that continuously measures the robots pose and provides feedback to the operator. Once in danger of tipping, the robot gives the operator the option of proceeding or returning to a known safe position. The software algorithms developed for the platform intelligently monitor all aspects of the robot including current usage, temperature, ambient light, video motion-detection and wireless signal strength to provide the user with as much vital information as possible. Capable of stair climbing, under vehicle inspection and long-range surveillance, OzBot maintains a high level of controllability using advanced sensory systems combined with an intuitive user interface. High-resolution digital video and long-range control, tracked or wheeled formation, modular upgradeability and serviceability, high strength motors and a rugged construction is merged to form a unique and fully capable Improvised Explosive Device (IED) robot [23].

Figure 5: The OzBot setup with the capability of haptics and remote operation [9].

The OzBot is equipped with the Haptic Manipulator, which is a custom designed robotic arm developed specifically to provide force feedback to a remote operator. The Haptic manipulator works as a slave device to a user's input and is fitted with a number of force sensors in the gripper and wrist. Currently the haptic manipulator can lift objects of up to 1kg and move at a speed of 4m/s. The manipulator allows users to feel an object between the remote gripper mandibles, and also the effects of force applied to the end of the arm.

Transparency allows a user to feel realistic forces without adjusting to mechanical issues such as backlash and the weight of the interface itself. More complex devices generally offer less transparency; however provide greater usability for the requirements of rendering and interacting with rich and complex virtual worlds. In Figure 6, the avatar models used are the OzBot mobile platform mode and a Counter Improvised Explosive Device (CIED) manipulator model [24]. The CIED manipulator has 5 revolute degrees of freedom - rotation on arm base, rotation on each of the three arm segments and pitch and roll of the wrist. A typical haptic device, such as Sensable Phantom Omni [25], has 3 active and 3 passive axes of rotation. As the link length and kinematic structure differs between the haptic interface and virtual manipulator coordinate system alignment and scaling are required. Preliminary force rendering routines are implemented within DI-Guy to allow the user feel collisions with the mobile platform, and also to represent the mass of the arm itself as it's in motion. The avatar can grasp and manipulate foreign objects within the virtual environment and reflect interactive forces, such as weight, balance and momentum, back to the user. Corresponding force effects such as the gravity will be felt during the transformation operation. To further extend the interactivity of the proposed pipeline, a 3D Connextion space mouse is used to control platform motion. Both the 3D Connexion device and haptic UI can be operated simultaneously. As the physical OzBot has a differential-steer configuration, the user input is dynamically filtered from the space mouse to ensure motions in the simulation accurately represent those of the real platform.

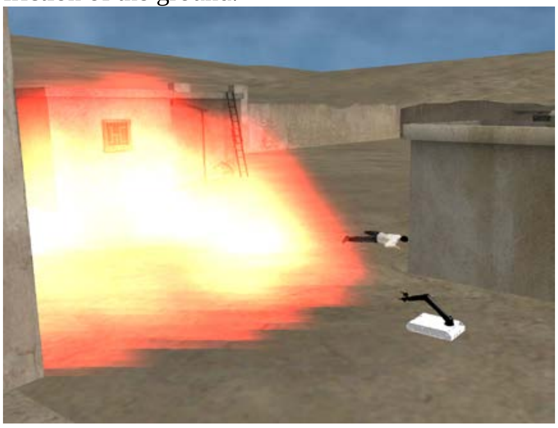
(a)



(b)

Figure 6: (a) Intergration of haptics and 3D mouse into DI-Guy to control the avatar as well as to feel force effects. (b) Interactive control of other avatars in DI-Guy using haptic device, with gravity and other physical properties defined.

Figure 7 shows another scenario that involves more haptic interactions and force rendering. The scenario describes a village attacked by explosives. There is an injured person lying on the ground unconsciously. He is close to an explosion and it is dangerous to send out a team to rescue him. An OzBot is therefore sent out to pull him out of the danger. The trainee will first need to use the 3D Connexion device to move the OzBot and try to find the injured person. As he is unconscious and very close to explosion, the pulling procedure has to be done very carefully. Since the most direct route has been blocked by explosions, the trainee will have to search for another route to reach the target. While rescuing him, the user will also have to constantly check the situation around and ensure that the injured person will not have more injuries during the pulling procedure. Dragging force is calculated and rendered based on the weight of the injured person as well as the friction of the ground.



(a)

(b)

Figure 7: (a) Initial scenario setup. Explosion is dangerous and remote haptic enabled rescue has to be involved (b) OzBot reached target and tries to rescue him by pulling him out of the explosion area. Procedure with gravity and friction properties defined.

6. CONCLUSION AND FUTURE WORK

To summarize, this framework provides a more intuitive and immersive way of interacting with off-the-shelf simulation software using haptic devices to enable direct control and force feedback. The platform is generally applicable to haptic devices and simulation software from various vendors. It can also be deployed on demand, with customizable functionality and immersion level. If the scenario information could be accessed in low level (with the knowledge of vertices and normal of visualized objects), interactive haptic collision detection could also be achieved and thus provides real-time haptic collision detection and object recognition.

Future areas of research include applying the proposed pipeline to other robot avatars in DI-Guy (such as CISR's Universal Motion Simulator) and investigate ways to further improve its stability and interactivity. Further enhancements will also improve the ability to measure the directional components of a collision force, physical property assignment/retrieval and force rendering of various effects. This work could ideally be compatible with many simulation software packages and environments. It is fast, affordable and safe to validate existing scenarios, as well as a training and mission planning tool for future interactive simulation procedures.

ACKNOWLEDGEMENT

This project is supported by the Centre for Intelligent Systems Research at Deakin University.

REFERENCES

[1] Blascovich, J Bailenson, J. Infinite Reality: Avatars, Eternal Life, New Worlds, and the Dawn of the Virtual Revolution, *Harper Collins*, 2011.

[2] Burdea, G. and P. Coffet: Virtual Reality Technology, Second Edition. *Wiley-IEEE Press*, 2003.

[3] Leyvand, T.; Meekhof, C.; Wei, Y.; Sun, J. & Guo, B.: Kinect Identity: Technology and experience Computer, pp. 94-96, 2011.

[4] Chu, C.; Dani, T. & Gadh, R. Multi-sensory User Interface for a Virtual-Reality-Based Computer Aided Design System, *Computer-Aided Design*, 29, pp. 709-725, 1997.

[5] I. Brent Edward: Passive Haptics Significantly Enhances Virtual Environments, *The University of North Carolina at Chapel Hill*, pp. 1-100, 2001.

[6] Salisbury, J. & Srinivasan, M. Phantom-based haptic interaction with virtual objects Computer Graphics and Applications, IEEE, vol. 17, pp. 6-10, 1997.

[7] Salisbury, K.; Conti, F. & Barbagli, F. Haptic Rendering: Introductory Concepts Computer Graphics and Applications, IEEE, vol. 24, pp. 24-32, 2004

[8] Boulos, M.; Hetherington, L. & Wheeler, S. Second Life: An Overview of the Potential of 3-D Virtual Worlds in Medical and Health Education, *Health Information & Libraries Journal*, 24, pp. 233-245, 2007.

[9] Fielding, M.; Mullins, J.; Horan, B.; Nahavandi, S.: OzBot™ - Haptic Augmentation of a Teleoperated Robotic Platform for Search and Rescue Operations, *IEEE International Workshop on Safety, Security and Rescue Robotics*, pp. 1-6, 2007.

[10] Von Hardenberg, C. and Bérard, F.: Bare-Hand Human-Computer Interaction, *Workshop on Perceptive User Interfaces*, pp. 1-8, 2001.

[11] Brooks Jr., F. P.: What's Real About Virtual Reality?, *IEEE Transactions Computer Graphics And Applications*, 19 (6), pp. 16-27, 1999.

[12] Burdea, G. and P. Coffet: Virtual Reality Technology, *Wiley-IEEE Press*, 2003.

[13] Wexelblat, A.: Virtual Reality: Applications and Explorations, *Academic Press Professional Cambridge*, pp. 993.

[14] Stone, R.: Haptic Feedback: A brief History from Telepresence to Virtual Reality, *Haptic Human-Computer Interaction*, pp. 1-16, 2001.

[15] Ang, Q.; Horan, B.; Najdovski, Z. and Nahavandi, S.: A Multi-point Haptic Platform for Grasping and Manipulating Virtual Objects, *IEEE Conference on Virtual Reality*, pp. 1, 2011.

[16] Saddik, E.: The Potential of Haptics Technologies *IEEE Instrumentation and Measurement Magazine*, pp. 10-17, 2007.

[17] Vlasov, R.; Friese, K. and Wolter, F.: Ray casting for Collision Detection in Haptic Rendering of Volume Data, *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pp. 215, 2012.

[18] Ferre, M.; Cerrada, P.; Barrio, J. & Wirz, R.: Haptic Rendering Methods for Multiple Contact Points, *Immersive Multimodal Interactive Presence*, 9, pp. 157-169, 2012.

[19] Piero Foscari: The Realtime Raytracing Realm, ACM Transactions on Graphics.

[20] Hayward, V.; Astley, O.; Cruz-Hernandez, M.; Grant, D. and Robles-De-La-Torre, G.: Haptic Interfaces and Devices, *Sensor Review*, 24, 16-29, 2004.

[21] Luck, M. and Aylett, R.: Applying Artificial Intelligence to Virtual Reality: Intelligent virtual Environments, *Applied Artificial Intelligence*, 14, pp. 3-32, 2000.

[22] Botden, S.; Torab, F.; Buzink, S. and Jakimowicz, J.: The Importance of Haptic Feedback in Laparoscopic Suturing Training and the Additive Value of Virtual Reality Simulation, *Surgical Endoscopy*, 22, pp. 1214-1222, 2008.

[23] Schultz, E.; Curry, R. & Niksch, R.: Improvised Explosive Device Countermeasures, Google Patents, 2009.

[24] Wilson, C.: Improvised explosive devices (IEDs) in Iraq and Afghanistan: Effects and countermeasures, *CRS Report for Congress*, 2005.

[25] Massie, T. and Salisbury, J.: The Phantom Haptic Interface: A Device for Probing Virtual Objects, *Symposium on Haptic Interfaces for Virtual Environment and Teleoperator systems*, 55, pp. 295-300, 1994.

[26] Bourdot, P.; Convard, T.; Picon, F.; Ammi, M.; Touraine, D. & Vézien, J. VR-CAD integration: Multimodal immersive interaction and advanced haptic paradigms for implicit edition of CAD models Computer-Aided Design, Elsevier, 2010, 42, pp. 445-461

[27] Kim, J.; Sylvia, I.; Ko, H. & Sato, M. Integration of Physics Based Simulation with Haptic Interfaces for VR Applications HCI International, 2005