I-TOUCH: A generic multimodal framework for industry virtual prototyping

Aurélien Pocheville
Laboratoire Systèmes Complexes - CNRS - UEVE
40, rue du Pelvoux, 91020 Evry, France
aurelien.pocheville@haptique.com

Abderrahmane Kheddar, Kazuhito Yokoi
AIST-CNRS Joint Robotics Laboratory JRL
IS, Natl Inst of AIST, Tsukuba Central 2, Japan
abderrahmane.kheddar@aist.go.jp

***Abstract*— Simulations based on virtual reality techniques make often special arrangements for haptic rendering. In fact, in most cases, haptic rendering drives the design of the simulation engine. This work proposes alternative software architecture to handle multimodal and human centered interactive rendering with a particular emphasis for the computer haptics problem. Namely, the architecture allows handling both haptic devices requirements, in terms of high refresh rates, and physically-based simulations requirements, in terms of CPU time. The developed I-TOUCH framework is designed to address these issues; in the meantime, it provides an open architecture and powerful tools to benchmark robustness of subsequent algorithms. All undergoing developments are being tested with actual industry virtual prototyping scenarios, the complexity of some of which highlights the extent of the fundamental problems to overcome.**

***Keywords* Haptic feedback, virtual prototyping, computer haptics, I-TOUCH**

I. INTRODUCTION

In order to reduce costs while increasing production efficiency, virtual prototyping (VP) (to be seen as a complementary tool to CADM software techniques) is considered as a promising perspective. It is the front end of a product life management process taking into account constraints related to manufacturing, utilization, and maintenance. To fulfill human-centered designs, the VP architecture should allow "digital mock-up" to be interactively explored, manipulated, and tested in various usage scenarios. This highlights the major importance of the human sensory capabilities other than the visual one. Indeed, in most maintenance and assembling/disassembling instances, feeding back haptic information to the operator is as much essential as vision or sound. Haptic interaction has two main components:

- the kinesthetic part, which reflects motions and forces,
- the tactile part, which reflects touch (surface roughness, shape, thermal exchange, etc.)

Our research focuses on many aspects of haptics foundations: human science related (eg. haptic psychophysics) and technology related (eg. interface design and computer haptics). In order to the meet both research and industry transfer purposes, we conceived a *multimodal* haptic framework, called I-TOUCH, that allows fundamental developments and models to be benchmarked for specific or general use.

II. THE I-TOUCH FRAMEWORK

A software architecture that successfully merges high refresh rates, *high fidelity* multimodal interaction rendering (that is to say visual, haptic and audition) does not exist at this very moment. The integration complexity, and each modality related specific refresh rates, compel to account for one problem at a time and make use of PC based clusters to fulfill the computation time requirements. If we add the option to make such software open and flexible to serve as a research benchmarking tool, it makes it definitely a hard initiative: I-TOUCH is an alternative architecture build on the basis to satisfy both issues.

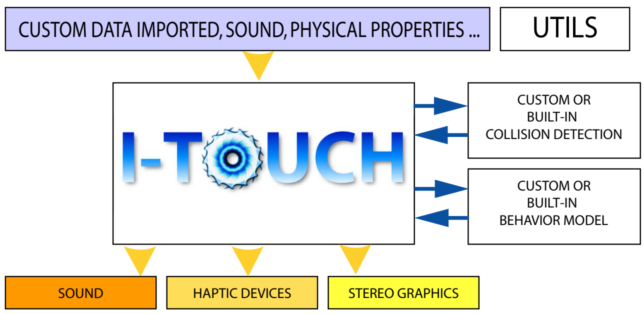

Fig. 1. I-TOUCH architecture: a simplified view.

As shown in the figure 1, the I-TOUCH framework is designed to be modular. Each of its components can be replaced easily. Modularity is achieved through an object oriented design, implemented in C++. For academic purposes, two behavior models have been implemented and can be switched. Note that the "physics engine loop" is considered to be the most important part of the framework, and that the real world is "interfaced".

One of the major features of this framework is in handling haptic devices. Contrary to almost all available haptic libraries (including commercial ones), I-TOUCH treats a virtual object linked to any haptic device just like any other objects of the simulation environment. The virtual object is simply "attached" to the device. Then, thanks to its haptic proxies, I-TOUCH takes care of the proper interaction. This technique not only allows greater freedom but make very simple haptic device interchanging. In order to use any kind of passive or active haptic device in I-TOUCH, a class must be derived. Methods include initialization, communication and shutdown codes. As for transparency and stability, the interface uses a technique minimizing the lag between simulation and physical device (for example, if the device is located at some part of the network, the haptic proxy accounts for this case).

Immersion is significantly improved when all modalities are blended within the same environment. We also created new haptic illusions, such as *haptic bump mapping*, which is synchronized with the visual bump mapping. This technique enhances considerably haptic interaction. In the future, we will interface new devices that better mimic tactile interaction. Haptic device abstraction is the key issue of any envisaged extension. Latest computer graphics technologies are also included: pixel and vertex shading, 3D positional audio, etc.

I-TOUCH data are stored in an easy to understand and open standard formats. For example, XML is used for configuration and non-geometry. Another file format can bear for almost any geometrical data format, without affecting the parsing speed and the extensibility. An exporter for 3DSMax has been written to ease 3D model importation. Other applications can be interfaced easily. Moreover, an off-line scene explorer allows scene viewing without running the simulation.

As a proof of extensibility and flexibility, we programmed a sample virtual scribble application. It simulates writing with a pen on a sheet of paper. The implementation scheme for this application is performed as follow:

1) import a pen and a desk model in 3DSMax and export them to I-TOUCH data,
2) in I-TOUCH configuration files, link the pen model to the haptic controller (a PHANToM has been used); the desk is made static,
3) in the I-TOUCH code, add trails as pen interacts with desk. This step, in the future, could be separated from the engine through a scripting language,
4) compile and run the new “Virtual Scribble”.

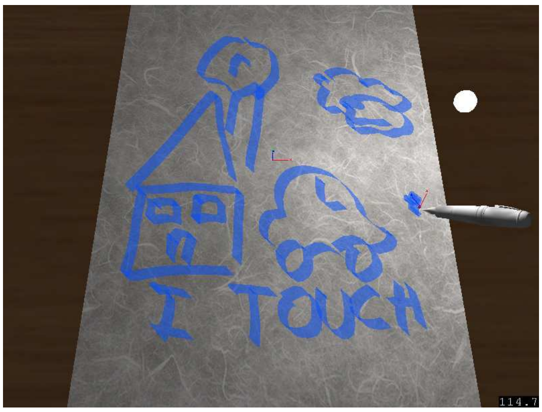

Fig. 2. Virtual Scribble sample application.

As we can see, this is very straightforward, and does not require great amount of I-TOUCH internals.

Another test case is the mounting/dismounting of a window-winder motor in a car door. The operator can test if the window-winder does really fit, and if it is possible to put it in place, accounting for the shape of the door car. Such a scenario is showed in the figure 3.

Fig. 3. An automotive industry VP case. Car door courtesy of Renault©.

III. MARKET SURVEY

There is indubitably a market for virtual prototyping: it equals the CADM one. Design areas, from kitchenware to car assembly, could use virtual prototyping in a wide range of applications. The cost of a prototype is varying from application to application, but is generally priced more than $1 000, and can easily reach $10 000. These costs are to be multiplied by the number of prototypes. While being developed within an academic and research environment, the I-TOUCH framework already unveils its potential to address such needs. Such a software could be licensed for as low as $1 000, and thus interest many industrials for year or lifetime licenses.

ACKNOWLEDGMENT

This work is supported by the European Project TOUCH-HAPSYS `http://www.touch-hapsys.org/` *-Towards a Touching Presence: High-Definition Haptic System-* IST 2001-38040.

REFERENCES

[1] G. G. Wang, “Definition and review of virtual prototyping,” *Journal of Computing and Information Science in Engineering*, vol. 2, no. 3, pp. 232–236, 2002.

[2] A. Gomes de Sá and G. Zachmann, “Virtual reality as a tool for verification of assembly and maintenance processes,” *Computers & Graphics*, vol. 23, no. 3, pp. 389–403, 1999.

[3] A. Gregory, A. Mascarenhas, S. Ehmann, M. Lin, and D. Manocha, *Six Degree-of-Freedom Haptic Display of Polygonal Models*, 2000, pp. 139–146.

[4] G. Pava and K. E. M. Lean, “Real time platform middleware for transparent prototyping of haptic applications,” in *12th International Symposium of Haptic Interface for Virtual Environment and Teleoperator Systems*, 2004, pp. 383–390.

[5] A. Pocheville and A. Kheddar, “I-TOUCH: A framework for computer haptics,” in *IEEE/RSJ IROS, Workshop on Touch and Haptics*, 2004.

[6] J. Sauer and E. Schömer, “A constraint-based approach to rigid body dynamics for virtual reality applications,” *Proc. ACM Symposium on Virtual Reality Software and Technology*, 1998.