Proceedings of International Conference on Advances in Infrastructure for Electronic Business, Science and Education on the Internet, SSGRR-2000, l'Aquila, Italy, July 31 - August 6, 2000.

Designing Virtual Environments for Critical Transactions and Collaborative Interventions: the VERTEX/APIA Framework for Networked, Physics-Compliant Objects

Denis Poussart (1), Denis Laurendeau (1), François Bernier (1)
Martin Simoneau (1), Nathalie Harrison (1)
Denis Ouellet (1), Christian Moisan (2)

(1) Computer Vision and Systems Laboratory
www.gel.ulaval.ca/~vision/ and
(2) Interventional MRI Unit, CHUQ
Université Laval, Sainte-Foy, Québec, G1K 7P4, Canada

contact: poussart@gel.ulaval.ca

Abstract

The expanding availability of high bandwidth networking is heralding a new generation of resources targeting very tightly-coupled distributed interactions. A case in point is the virtualization of critical tasks: advanced training, hazardous industrial operations, or minimally invasive surgery all require that complex entities - mechanical, thermal, physiological, etc. - be blended into cohesive "worlds" and made available to multiple actors requiring different levels of representation. In this paper we discuss issues related to the integration of computer vision and behavioral modeling, hard real-time compliance, the architecture of distributed simulation and an object-oriented framework with an extensible knowledge of physics, leading to the design of Extreme Virtual Environments. We also present current exemples targeting underwater robotics and computer-aided surgery.

Introduction

The expanding availability of high performance computing, networking and human machine interfaces is heralding a new generation of advanced Virtual Environments.

In a not-so-distant future, through the powerful integration of conceptual and technological resources, we may envision systems capable of providing a convincing and highly realistic experience of telepresence. Such systems will allow multiple users to undertake collaborative tasks and to share common or related "worlds". The scope of potential applications is immense, and encompasses all of human activities. One significant area of future development of VE's lies in providing strategic and / or tactical support in the collaborative execution of complex tasks, where failure to properly execute may have grave consequences in terms of human welfare, safety, environmental damage or cost. This is the realm of what we might label as Extreme VEs (EVEs). We are currently exploring this type of resource integration, involving advanced virtualization with modeling and simulation, through VERTEX [1], a project of the *Institute for Robotics and Intelligent Systems* carried out under the Networked Centers of Excellence program of Canada. VERTEX seeks to develop a generic framework for such systems which may best leverage the expanding information technologies, including the synergy of computing and networking and the availability of Commercial Off The Shelf components. As exemples, VERTEX is currently developing two test beds - in industrial robotics and in computer-aided surgery - which are briefly described below.

There have already been examples of such systems in advanced applications such as training for space or through the High Level Architecture approach developed by the military. However, with the rapidly decreasing cost of the required infrastructure, we may now expect EVE systems to gradually become ubiquitous and applied to a broader range of situations.

1 Systems requirements for EVEs

Figure 1 shows the main operational components of VERTEX, in this case as they would materialize while targeting support for assessing and repairing large underwater structures. The harsh and dangerous intervention dictates the use of teleoperation while the high costs involved - stopping a major facility may entail losses of $M's per day - call for upmost care and efficiency in planning and executing the work. The virtualization of key aspects of such complex operations is cost effective

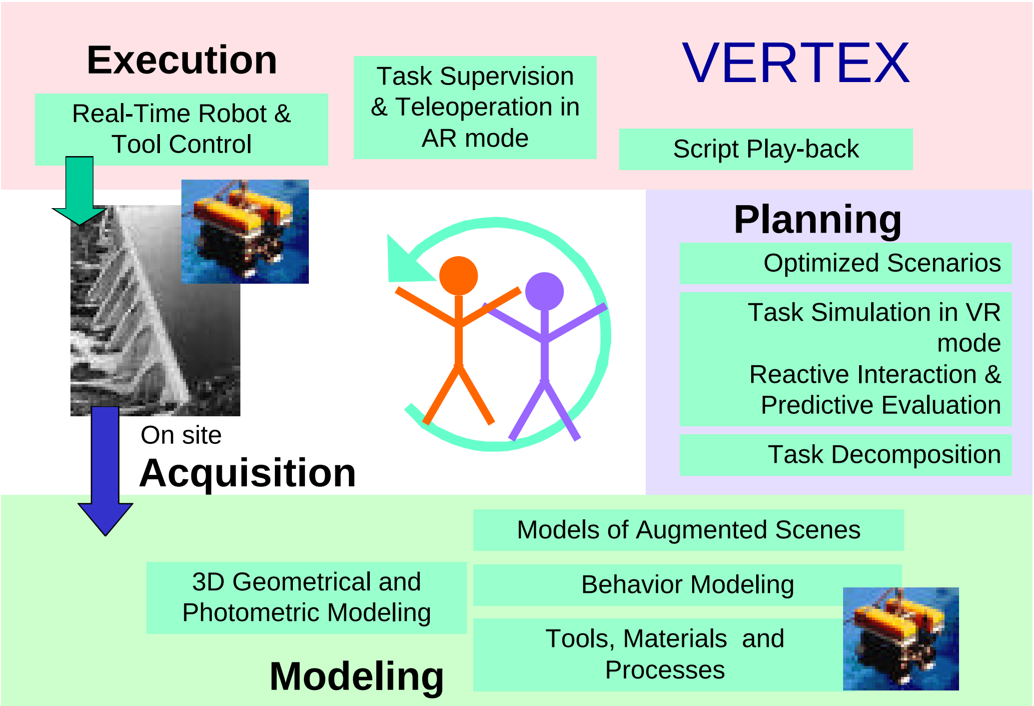

Figure 1: The VERTEX framework is designed to provide comprehensive support for critical operations, as illustrated here for the teleoperated maintenance of underwater structures. Its exploitation starts with on site acquisition leading to extensive 3D modeling. Models are then augmented with physical behaviors. Through advanced visualization, the details of the task at hand may be planned, rehearsed, and eventually executed under supervised, strict real-time conditions.

and is anticipated to be a major contribution. There is a wealth of similar application areas which we may expect to emerge in the future in industry, medicine, training and remotely provided services.

The VERTEX environment is architectured as a suite of resources designed operate in three main modes:

- A preparation phase, typically carried out off-line, involving the acquisition and modeling of an operating site and relevant tools. In the example shown in the Figure, the geometrical modeling could be based on a combination of on-site measurements acquired from range sensors mounted on a Remotely Operated Vehicle and fused with plans drawn for the initial construction of the structure (plans are often incomplete in legacy structures since modifications performed over time have not been integrated into a coherent data base; natural changes have also occurred). Tools to be modeled would include the ROV with its control and response dynamics and the drag of its tether, sensors such as ultrasonic-based position detectors, and actuators such as the dynamics of high pressure jets used to clean the fractures having developed within the structure to be repaired.
- A planning, strategic phase, supporting detailed task decomposition, scenario evaluation, and "what if?" type of reasoning. In the example given this could deal with identifying an optimum path sweeping the structure in minimum time, rehearsing for avoiding collision while approaching the entrance of a penstock, planning for the optimum allocation of resources involved in ground support, etc. This is a virtual reality regime.
- An execution, tactical phase, operating in critical real-time and with a tight coupling occurring between the prediction generated by the system and the actual physical components of the intervention, including updated acquisition. This is an Augmented Reality regime. In this example, there would be need to insure registry between the predicted and actual location of the ROV and to provide peripheral vision and synthetic points of gaze to the ROV operator using 3D scenery regenerated from initial data in such a way as to alleviate visibility degradation arising from turbidity. In large operations of this type, there might be several ROV used simultaneously and cooperatively, and it would be the responsibility of the system to help maximize their synergy. People involved would include ROV operators, engineers and project managers, each requiring a distinct window into the operating world which is well suited to his / her needs.

From an *operational* perspective, there are a number of further considerations associated with such EVE's:

- There should be means for the reliable and accurate visualization of actual, on site objects and scenes, as opposed to simply using synthetic representations of human-generated environments. These representations should be extended by comprehensive behavioral modeling. The visualization could include the display of information which is not intrinsically visual, for instance an on-going evaluation of risk as a delicate task is being simulated. It could also include the visualization of anticipated results or consequences generated by predictive mechanisms. Maintaining a coherent view and avoiding sensory overload to the operators would be an important concern.
- The human operators assume a central supervisory stance and interact through advanced human-machine interfaces. In the context of EVE's, typically there would be relatively few people in the loop, *e.g.* tens rather than thousands. In this sense, EVE's differ from some of the internet-based VE's which are currently under development and which target scalability to a very large number of users, *albeit* with much lesser coupling than is being considered here.
- Complex tasks require a range of expertise: the people involved could (would) therefore have different specialties, hence require personalized, different views on the unified, virtualized world where they need to collaborate. These expertise would be further enhanced through the rehearsing capabilities provided by the EVE system itself.
- Components (sensors, databases, tools, high performance computing, etc.) as well as some of the people involved might be geographically distant, for instance in the context of telemedecine or telerobotics. Therefore the system architecture should be network oriented down to its core, with concern for communication performance and reliability.
- From a designer perspective, such a system needs to be tailored for rapid prototyping: different complex operations may share generic characteristics but typically exhibit one-of-a-kind requirements as well as changing operating conditions which may require considerable adaptability.
- Support for modularity, coherence, expendability and maintainability points toward an Object-Oriented, component approach. There should be provision for dynamic run-time behavior. Furthermore the software structure should provide for interoperability of heterogeneous components and linkage to legacy systems.

Mission-critical, real-life interventions lead to especially stringent requirements in term of representation consistency and robustness. Models necessarily represent simplified abstractions of reality. Here however their (unavoidable) limitations should be explicit, and playing "tricks" to achieve a sense of presence could have grave consequences. For instance exploiting texture dynamics is a well known expedient capable of generating a convincing impression of motion: it may be very appropriate in a computer game or in an architectural walk-through but could lead to severe misjudgments in the case of EVE's.

We note then that important characteristics must further include:

- comprehensive implementation of spatial structure and motion,
- extensive modeling of all relevant physical properties and behaviors,
- adequate handling of time, typically implying a variety of time frames and resolutions, up to and including hard real-time support.

2 Accurate virtualization of the 3D shape of actual objects or scenes

Virtual Environments typically conjure a computer-mediated world which is enhanced by a variety of sensory interactions. While auditory and haptics interfaces are well recognized for providing a powerful contribution to the sense of "presence", typically the visual modality remains first and foremost [2,3].

It is especially noteworthy, then, that in recent years the research community involved in advanced 3D imaging has witnessed a powerful convergence process between Computer Graphics and Computer Vision, two areas which had previously evolved as distinct domains [4]. CG and CV both deal with 3D representations. Their core challenges, however, proceed from opposite ends of the imaging spectrum. In CG, the data thread emphasizes synthesis and targets the rendering of convincing visual percepts from *a priori* postulated models, which typically include volumetric, surfacic, and photometric components. In CV, the data flow is rather one of analysis, as it seeks to transform raw images acquired from actual objects and scenes into models which are optimized with respect to some end-use, ideally with no or minimal human intervention. The central issues involved in both disciplines tend to be quite different. For instance, CG may be concerned with the accuracy and the algorithm efficiency of the rendering of the image of a synthetic human face while CV has to deal with the robustness of model extraction from sensor data which are imprecise, incomplete or mathematically underconstrained and needs to invoke AI and knowledge encapsulation concepts.

While it could be observed that the visual models - start points in CG, target points in CV - were quite similar, little interaction occurred between the respective communities until the recent emergence of Virtual Environments. In an approach which has been referred to as “image-based rendering”, “image-based modeling”, or “virtualized reality” it has become recognized that some of the virtual worlds to be visualized should not be created *ab initio* from models (solely) generated in the mind of artists or architects, but, rather could (should) also be derived from or augmented by models extracted from real objects by means of CV methods [5,6,7,8].

Techniques for acquiring 3D shape data have undergone considerable development over the last decade and a wide range of devices and methods are available, including laser triangulation [9], laser time-of-flight, stereo, enhanced defocusing [10], speckle imaging [11], shape from video sequences, or structured light [12], to name a few. Depending on the application area, other imaging modalities may be involved, such as ultrasound, X-ray, thermal, magnetic resonance, or radar for instance. Some of these are suitable for dynamic scenes while other trade real-time acquisition for metrologic-like, static accuracy. When coupled with segmentation, recognition, pose estimation, mesh reduction and heuristic capabilities of CV, such sensors are capable of rapidly yielding the spatial representation of “things” in formats which are suitable for driving the graphic engine of an advanced VE, while minimizing the need for human intervention [13].

It should be emphasized that the level of scene description needed in EVEs must include a comprehensive implementation of spatial structure. For instance planning and controlling the path of an underwater Remotely Operated Vehicle for rasterizing cracks which have developed within penstocks, or selecting the orientation of an endoscopic probe being inserted into a human body do require accurate, genuinely 3D-oriented models. Traditionally, high level graphic computers have been needed to achieve sufficient frame rates with such scenes involving a large number of 3D polygons. However the performance of desktop graphics is rapidly improving to the point where they are beginning to be adequate for quality interactive VEs. Thankfully, resorting to technical “cheating” to achieve apparently acceptable performance will be less necessary in forthcoming years.

3 But for Extreme VEs, reality is not limited to shape. It must also include a detailed representation of physics and behavior.

There exists numerous software environments designed to support the authoring of VEs [14]. Many include built-in support for representing the physical behavior of objects, but this is typically limited, for instance to collision detection or gravity, as exemplified in VRML semantics. The field of physics modeling and simulation is immense and very well developed, and has been exploited with considerable success in special purpose environments, such as high performance flight simulators. But there are still a large number of issues regarding its integration into open architecture VEs [15]. Achieving satisfactory computing performance is a serious challenge for dealing with EVEs involved in complex situations.

A partial list of questions would include:

- What is relevant to be physically modeled? The modeling space of possibly relevant physical behaviors is boundless. Here we are faced with the same dilemma as a designer’s who must prune possibilities and evaluate the economics of a “proper” implementation, *i.e.* the simplest that fulfills the intended purpose.
- What is the needed Level of Detail? Any given physical entity can be modeled at a multitude of abstraction levels, ranging from the nano to the macroscopic. For efficient use of systems resources, which are always limited - and which shall always remain so as the refinement of applications expand - at any given moment the simulation should only generate that unfolding of the “world” which is necessary and sufficient for the intended purpose. There are only two areas in current VEs where LOD is extensively used: the visualization of texture mapping onto the 3D surface of objects which modulates spatial definition as a function of viewing distance, and collision computation, which uses bounding volumes of different details as a function of the imminence of object trajectories intersecting one another. In Computer Vision, a somewhat similar approach occurs in the well-known strategy of pyramidal, multiresolution processing, where a given raw input image is first processed at a rough resolution in order to identify potential Regions Of Interest, the identification of which then drives a mechanism of adaptable zooming into the tentative ROI’s. We believe that LOD management is an area of significant importance in fostering the introduction of extended physics modeling as needed for EVEs and that considerable research is called for in this area. While LOD implementation and management may be relatively straightforward on a case by case basis, *e.g.* for a specific type of behavior such as collision, developing a generic approach applicable to arbitrary physical phenomenons is an interesting challenge. Some of the issues to be addressed include the following:
    - What would be appropriate forms of models, both from physics and from software implementation aspects?

- Is it appropriate (necessary) to define the multiple LOD's of a given behavior explicitly and segment them fully? Should they be defined in a hierarchical type of structure? How should the discrete granularity of levels be selected or is it possible to envision an approach where the LOD's of a given behavior morph into a continuous simulation space?
- How to manage the transitions from one LOD to the next and what type or class of sensing daemon to use to control the transition throughout the LOD space?
- What is the impact on performance and the corresponding trade-offs on system complexity, robustness and adaptability?

In any case, while these questions are still largely open, the underlying software architecture of EVEs should be such that some of the aspecta above may be supported. This is a feature of VERTEX.

4 A main component of physics modeling relates to the handling of time

Since EVEs deal with the virtualization of situations which are intrinsically dynamic, the proper management of time is a central concern. This aspect has many facets since there are actually several different "flavors" of time in VEs :

- Time might just flow out of the action loop under the control of the users who are generating commands. Since a user's notion of time has limited accuracy, an adequate sense of presence might just require that the system responds in what is perceived as "real-time" to the user input, *i.e*. "fast enough" to support an immediate association between command and response.
- It is well known that in an interactive visualization users are able to adjust, to a point, to modest delays if these are constant. However, usability studies show that latency jitter is more consequential than delay. It jeopardizes the efficiency of the compensating feed-forward mechanism of the user, and may seriously hamper the effectiveness of the VE experience. This is especially significant for a distributed EVE, and suggests the need that a policy of Quality of Service be instantiated in the communication infrastructure [16].
- During strategic planning activities, we have mentioned the opportunity of providing resources capable of supplying projections, in future time, of the consequences of actions being currently taken. Such predictive processes are running in a regime which is faster than real-time.
- Different physical components have a range of dynamics: they may require different time resolutions and run time optimization should support explicit and fine control of timing granularity. For instance an algorithm involved in collision-avoidance may need an update rate of path control in the order of the KHz while another control loop, say involved in thermal control, may be very adequately handled at rates of a few Hz. During the phase of direct, immediate, supervised execution of the task, *i.e*. in the Augmented Reality mode, the EVE system becomes tightly coupled to the events, objects, and processes which are actually occurring. In a ROV, for instance, automated coordinated control is in the immediate loop of action. In order to ensure proper synchronization, it is essential that the section of the EVE system which is directly concerned maintains timing accuracy. This is the realm of hard real time, *i.e* of a computing mode which can effectively guarantee that requests generated by sensors and signals sent to actuators absolutely occur within a prescribed latency and conform to rigorously defined priorities. Hard real-time control is well known in mission-critical industrial or medical applications. Its requirements are far more stringent than for the more common mode of soft real-time, where performance levels are only guaranteed in average. Hard real-time requires that the underlying software Operating System be designed, ground up, with corresponding core features such as suitable thread and process management, interrupt response, or handling of priorities in I/O queues. Common OSs such as WinNT or unix / linux have not been designed with such applications in mind and should not be used in critical segments of an EVEs. They have, however, excellent general purpose qualities and extended support, and are indicated for the non-critical segments. The overall EVE architecture should therefore be able to accommodate such heterogeneity.

5 The VERTEX / APIA architecture

Figure 2 highlights some important features of APIA, the "Actors - Properties - Interaction Architecture" of VERTEX which is currently under development and which addresses the issues mentioned above.

- It is fully net-centric, *i.e.* designed as a set of components which may be geographically distributed as needs arise without significant modifications.
- Unlike conventional VEs, VERTEX's core resource is a physics simulation kernel which is responsible for continuously maintaining a coherent and comprehensive representation of the simulated "world". The kernel is a logical, object-oriented structure. It includes a simulation engine which overviews sequencing and a manager which is responsible for allocating computing and communication resources. It hosts actors, which embody the physical entities, with properties, which

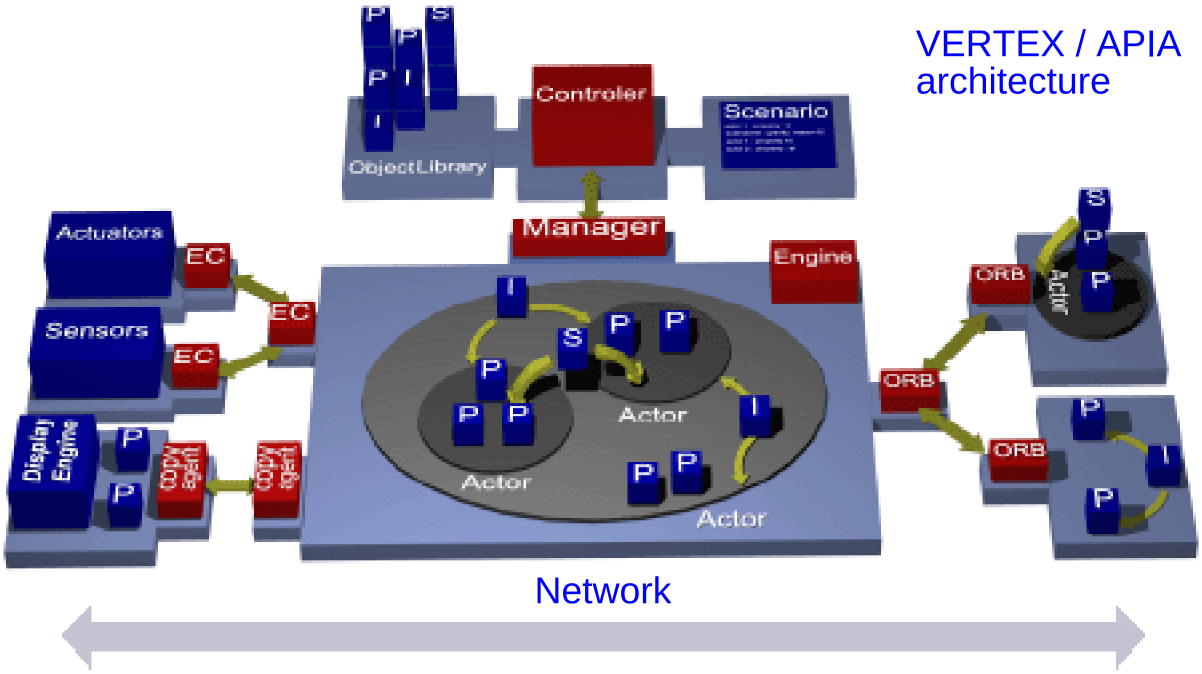

Figure 2: The VERTEX / APIA architecture is net-centric and consists of modules which are interconnected through a CORBA-based software bus implemented on a high-speed network. The simulation kernel, shown in the center and right-hand side, forms the core of the system. It integrates *actors*, which correrspond to the entities involved in the intervention being planned / executed, *properties* which reflect physical laws, and *interactions*, through which the dynamics of interplay between entities take place. User interfacing, shown on the left-hand side, is not part of the critical simulation loop, which executes on a hard real-time OS within the kernel. Other supporting components such as object libraries, overall control ans scenario editing support, are shown in the background.

are shared data and interactions which encapsulate physical laws. The kernel may be physically distributed if needed, as long as the impacts on performance are well understood. It runs under a hard-real time Operating System and is extensible at run-time.

- The simulation of complex behaviors, especially if performed with the added constraint of real-time, lies in the realm of High Performance Computing. In VERTEX, for maximum flexibility and cost effectiveness, we choose to implement the simulation kernel as a computing “farm” of “Commercial, of the Shelf Components”. This is very much in the spirit of the approach typically labelled as Beowulf [19], based on the clustering of commodity parts interconnected through a high speed network. Beowulfs are making serious inroads into the HPC arena. Most of their implementations, however, are based on linux and target batch-like computation on large data sets. In VERTEX, the computation flavor is somewhat different, and the hard real time capabilities, which are normally ignored, are supported by using OS’s such as Lynx [17] and QNX [18].
- Visualization and other modalities such as haptic and sound interfaces are, of course, key components of VEs. However, in VERTEX, they are considered as peripheral to the core of the physics kernel: data being visualized, commands and interactions provided by users are seen as important, but nevertheless side effects occurring in an overall system which is undergoing its course as a consequence of explicit laws of physics.
- This fundamental choice has important benefits:
  - It naturally provides a coherent picture whereby multiple users may enjoy different representations of the overall “world” while being connected to the single - and therefore common - simulation engine.
  - From an implementation point of view, it allows for an effective decoupling between the dynamics of physics simulation and those of visualization. The update rate of a particular component of the model should not depend upon the rate of the frame buffer or its associated priorities be linked to the traversal of the display graph.
  - This decoupling provides great freedom in the choice of implementation platforms. For instance hard real-time support is not really needed for the visualization components, and this allows VERTEX to exploit the modern crop of high performance, low

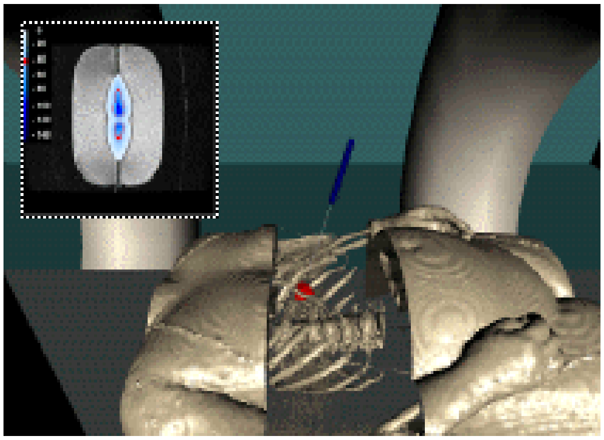

Figure 3. Output visualization of the VERTEX system while providing support for a (simulated) cryosurgery. The trunk of the patient is shown in the workspace of an open-field Nuclear Magnetic Resonance unit. The segmented liver tumor has been extracted from previous NMR scans and is visible in real time but with reduced resolution in the operating room. Also shown is the location and orientation of the cryogenic probe being positionned by the surgeon. The visible insert shows the shape of the corresponding cold front predicted from a finite-element simulation of heat tranfer in the tissue.

cost OpenGL 3D graphics subsystems available on WinNT (and forthcoming on Linux).

- VERTEX uses CORBA as its software bus. Admitedly the CORBA middleware involves a significant amount of overhead. However, in VERTEX, the focus is on robustness, modularity, reusability, maintainability and interoperability, qualities which CORBA supports remarkably well. This selection also acknowledges the fact that, as time evolves, computing resources will further improve and the relative burden of logistical overhead will decrease. Furthermore, as we enter into more and more complex operations, the solid foundation provided by CORBA will be a significant asset. Choosing CORBA does not preclude the use of other alternatives such as DCOM or Jini, but it ensures, at this time, an exceptional capability for interoperability since its implementation is available for all significant hardware - software platforms. In VERTEX, all modules are constructed as CORBA clients. We use ACE/TAO, an implementation developed at Washington University which is a driving force behind the real time, Quality of Service, deterministic aspects that are evolving as the CORBA 3.0 specifications of the Object Modelling Group consortium [21].
  - VERTEX, with its use of CORBA, is designed to take full advantage of the emerging capabilities of very high performance networks which are being put in place. A case in point is Canada's CaNet3, a world-first, Dense Wavelength Division Multiplexed optical network with a bandwidth of 40 gigabit per sec., which was deployed by CANARIE Inc. in late 1998 [22].

6 Current status and applications

The VERTEX Project is currently at mid course of its four-year calendar and has reached the stage of a working prototype. with several of its key components, including the simulation kernel, have been implemented and successfully tested. Over the next months, VERTEX will be further refined, and augmented by a variety of resources which are still at the early design stage, including interactive tools required for scenario editing and the inclusion of a more complete simulation package. This initial version is operational in the VR facilities of our Laboratory.

We cannot overemphasize the importance of validation, especially when dealing with such complex modeling and simulation resources, and this will require significant efforts.

Two application areas have been selected as initial test beds. The first one relates to underwater telerobotics. It is conducted in collaboration with the Robotics Division of the Research Institute of Hydro Québec [23] and has been briefly presented in the context of Figure 1. The second one seeks to provide strategic and tactical support to image guided cryotherapy whose goal is to provoke a complete destruction of tumoral cells *in situ* through a thermal stress at cryogenic temperatures [24]. Nuclear Magnetic Resonance imaging enables minimally-invasive surgery by targeting the tumor site through a per-cutaneous track, usually a working channel only a few millimeters in diameter through the skin. It also allows to directly monitor the treatment as it takes place. Figure 3 illustrates the type of intervention visualization currently available with VERTEX which exploits the facility of an open-field NMR facility which is available to the team, one of a few of such instruments presently available worldwide. The environment provides a segmented view of the tumor, extracted from pre-operational imaging and fused with real-time NMR data. It also displays the results of a simulation of the propagation of the cold front generated at the tip of the cryogenic probe as it is being manually controlled by the surgeon. In this case, the modeled physics are of course quite different from those of underwater robotics, and involve finite-element computations withinin the spatial structure of the patient tissue. When freezing occurs, NMR becomes incapable of generating a detailed map of the region of interest, hence the advantage of using a VE resource to provide an augmented view at this critical moment. The system, when fully developed, should also provide support for high quality training for such a delicate intervention, where it essential to limit the thermal stress to the tumoral tissue while causing minimal damage in its periphery.

Both applications are being developed on exactly the same hardware / software platform. Having to accommodate significantly different applications in terms of physics and dynamics simulation allows us to examine how generic the VERTEX platform is. Both are highly demanding in term of robustness and these initial experiences suggest that the VERTEX framework is indeed widely applicable.

Conclusion

This paper has outlined some of the issues involved in developing virtual environments capable of providing quality support for the execution of critical tasks. Several of these are still open questions for research, especially with respect to the integration of complex simulation in a dynamic interactive environment. Initial observations suggest that the approach is highly feasible, cost effective, and should generalize relatively easily to a wide suite of situations.

Acknowledgments

This research was funded in part by the *Institute for Robotics and Intelligent System* (Networked Centers of Excellence program of Canada), the *Natural Science and Engineering Council of Canada*, the *Canadian Foundation for Innovation*, and *Fonds pour la Formation de Chercheurs et l'Aide à la Recherche du Québec*. We gratefully acknowledge the contributions of the *Research Institute of Hydro Québec* and of Drs. F. Ferrie, R. Eagleson and M. Nahon, respectively from McGill, Western Ontario and Victoria universities, who are participants in the VERTEX Project.

References

[1] VERTEX: Virtual Environments: from 3d Representations to Task planning and EXecution. See `http://gel.ulaval.ca/~vision/`.

[2] J. Gabbard, D. Hix, “Taxonomy of Usability Characteristics in Virtual Environments”, Final Report to the *Office of Naval Research*, (1997), `http://csgrad.cs.vt.edu/~jgabbard/ve/taxonomy/`

[3] D. Hix, E. Swan, J. Gabbard, M. McGee, J. Durbin, T. King, T. “User-Centered Design and Evaluation of a Real-Time Battlefield Visualization Virtual Environment”, Proceedings of *IEEE Virtual Reality '99*, pp. 99-103, 1999.

[4] J. Lengyel, “The Convergence of Graphics and Vision”, *IEEE Computer*, vol.31, no. 7, pp. 46-53, 1998.

[5] L. McMillan and G. Bishop, “Plenoptic Modeling: An Image-Based Rendering System”, Proceedings *Siggraph 95, ACM Press*, New York, pp. 39-46, 1995.

[6] P.E. Debevec, C.J. Taylor, and J. Malik, “Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-based Approach”, Proceedings of *Siggraph 96*, ACM Press, New York, pp. 11-20, 1996.

[7] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, Da. Koller, L. Pereira, M.Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, D. Fulk, “The Digital Michelangelo Project: 3D Scanning of Large Statues”, to appear in Proceedings of *Siggraph '2000*, ACM Press, New York, 2000.

[8] T. Kanade, P. Rander, S. Vedula, and H. Saito, "Virtualized Reality: Digitizing a 3D Time-Varying Event As Is and in Real Time", *Mixed Reality, Merging Real and Virtual Worlds*, Yuichi Ohta, Hideyuki Tamura, ed., Springer-Verlag, 1999, pp. 41-57.

[9] M. Rioux, “Synchronous Optical Scanning Apparatus”. U.S. Patent 4,787,748. Issued November 29, 1988. Europe Patent 0290237B1. Issued February 3, 1993. See `http://www.hymarc.com/`

[10 F. Blais, M. Rioux, D. Poussart, “ A Very Compact Three-Dimensional Camera for Robotic Applications”, *Test and Measurement World*, April 1985. pp. 65-66. See `http://www.vit.iit.nrc.ca/Pages_Html/English/biris.html`

[11] See `http://www.inspeck.com/InSpeck-3D-DF.htm`

[12] For instance see `http://www.minolta.com/dp/3d1500/`

[13] P. Boulanger and S. El-Hakim, “Mobile System For Indoor 3-D Mapping And Creating Virtual Environments”, to appear in Proceedings of Workshop on Reality-based Modeling, *IEEE International Conference on Robotics and Automation*, San Francisco, April 2000.

[14] K. Watsen and M. Zyda, “Bamboo - A Portable System for Dynamically Extensible, Real-time, Networked, Virtual Environments”, Proceedings of *1998 IEEE Virtual Reality Annual International Symposium (VRAIS'98)*, Atlanta, Georgia, 1998.

[15] D. K. Pai, J. Lang, J. E. Lloyd, and R. J. Woodham. “ACME, A Telerobotic Active Measurement Facility.”, in Proceedings of the *Sixth International Symposium on Experimental Robotics*, Sydney, Australia, March 1999.

[16] K. S. Park and R. V. Kenyon. “Effects of Network Characteristics on Human Performance in a Collaborative Virtual Environment”, Proceedings of *IEEE Virtual Reality '99*, pp. 104-111, 1999.

[17] See `http://www.lync.com`

[18] See `http://www.qnx.com`

[19] See `http://www.beowulf.org`

[20] C. O'Ryan, D. C. Schmidt, F. Kuhns, M. Spivak, J. Parsons, I. Pyarali, D. L. Levine, “Evaluating Policies and Mechanisms for Supporting Embedded, Real-Time Applications with CORBA 3.0”, to appear in Proceedings of the *Sixth IEEE Real-Time Technology and Applications Symposium (RTAS'00)*, Washington D.C., USA, May 31-June 2, 2000.

[21] See `http://www.omg.org`

[22] See `http://www.canarie.ca/frames/startnetworks_e.html`

[23] See `http://www.robot.ireq.ca`

[24] A.A. Gage, “Cryosurgery in the Treatment of Cancer”, *Surgery, Gynecology & Obstetrics*, Vol. 174, pp. 73-92, 1992.